Arturo Quirantes Sierra

CUANDO LA CRIPTOGRAFÍA FALLA

Lectulandia

En nuestro mundo digital, la protección de la información es un elemento fundamental. Usted, sin darse siquiera cuenta, está utilizando criptografía a diario: al conectar el móvil, al sacar dinero del cajero, al navegar por Internet, al enviar mensajes por teléfono, al pagar con el abono de transporte. Incluso su documento de identidad y su pasaporte incluyen elementos criptográficos. Pero ¿sabe cómo funcionan? Y sobre todo, ¿sabe por qué fallan tan fácilmente?

He escrito este libro para intentar explicárselo. No se trata del habitual tratado sobre teoría de números, algoritmos de clave pública o firmas digitales, aunque también están explicados. Mi propósito es mostrar cómo la criptografía ha moldeado el comportamiento de industrias y modelos de negocio que usted utiliza asiduamente, combinando detalles técnicos con explicaciones cotidianas. A lo largo de las últimas dos décadas, sectores como la telefonía móvil o la discografía en DVD han sufrido profundas transformaciones, y en más de una ocasión la inseguridad de los sistemas de cifrado ha sido causa fundamental. Podrá usted ver resueltas dudas como:

- ¿Hasta qué punto los algoritmos criptográficos GSM son seguros?
- ¿Por qué en los años noventa pillaron a Txiqui Benegas hablando de “el enano” o “el one” en su flamante Motorola?
- ¿Cómo es posible que haya tanto fraude en tarjetas de crédito, ahora que los chips los hacen invulnerables?
- ¿Qué se han inventado los hackers de Boston o Londres para viajar gratis total con los nuevos bonos de transporte?
- ¿Qué hay de cierto en los rumores de que los mensajes de WhatApp están mal protegidos?
- ¿Realmente tantos usuarios utilizan 12345 como contraseña?
- ¿Puede ser hackeado un marcapasos?
- ¿Es tan bueno Dan Brown cuando escribe libros sobre criptografía?
- ¿Qué tienen en común la criptografía y el asesino que dio origen a la saga de Harry el Sucio?
- ¿Qué debo hacer si me roban la tarjeta de crédito?
- ¿Por qué las autoridades de Washington no pudieron alertar a tiempo a los mandos militares de Pearl Harbor sobre el inminente ataque?
- ¿Por qué tantos hackers odian a Sony y aman su PlayStation3?
- En serio, ¿debería cambiar mi contraseña? Me gusta 12345, es fácil de recordar. Tampoco es tan importante, ¿no?

No es necesario que tenga conocimientos matemáticos de alto nivel. Lo único que necesita es su experiencia de usuario. Si, por el contrario, se queda con ganas de ampliar información, el libro incluye más de quinientas referencias digitales de todo tipo. Un clic en el lugar adecuado, y su lector de libros le llevará directamente a la fuente original.

Y ahora es el momento de que usted juzgue este libro. Descárguese el 10% que le permite Amazon y disfrute sin compromiso. Si después sigue con ganas de más, no tiene más que hacer clic. Un complejo procedimientos de autenticación y cifrado se pondrá en marcha para que, en unos segundos, tenga a su disposición el libro completo.

Y recuerde: 12345 no es una contraseña válida. En serio.

Lectulandia

Arturo Quirantes Sierra

Cuando la criptografía falla

ePub r1.0
jandepora 10.02.14

INTRODUCCIÓN

Usted, lector, vive inmerso en un mundo de criptografía. Constantemente utiliza protocolos de autenticación, control de acceso, identificación y confidencialidad, y lo ha hecho desde que nació. Cuando de bebé reconocía a mamá por la voz y por el rostro, estaba llevando a cabo un procedimiento de identificación biométrica natural. Papi y mami le enseñaron a no abrir la puerta a desconocidos, no dejar las llaves de casa en cualquier sitio, revelar la mínima información posible a esa señorita que tan amablemente llamaba para vendernos algo por teléfono, y acostumbrarse a que había cajones a los que tenía vetado el acceso. Pronto aprendió que las llaves abren mundos ocultos. Ahí está Harry Potter, con sus palabras mágicas y sus contraseñas de acceso, para recordárselo.

Al crecer, la cosa no cambia; al contrario, va a más. Y no es por arrimar el ascua a mi sardina (bueno, sí que lo es), pero probablemente no pase un día sin que utilice usted algún tipo de criptografía: al conectar el móvil, al sacar dinero del cajero, al navegar por Internet, al enviar mensajes por teléfono, al pagar con el abono de transporte. Incluso su documento de identidad y su pasaporte incluyen elementos criptográficos.

Este libro electrónico es una buena prueba de ello. Usted lo habrá conseguido de una de estas dos formas: o bien lo compró legalmente, para lo cual tuvo que activar un complejo pero transparente sistema de claves criptográficas entre usted, la tienda y el banco; o bien lo descargó de una red p2p libre (reduciendo con ello mi fortuna personal, y espero que le remuerda la conciencia), para lo cual tuvo que poner en funcionamiento elementos como sistemas de autenticación, identificación y hash. En cualquier caso, su conexión ADSL, protegida mediante cifrado, fue el medio escogido para la transmisión de los datos; a no ser que se lo enviasen por teléfono, en cuyo caso… bueno, creo que ya va cogiendo la onda.

Los sistemas de cifrado y autenticación forman parte natural de nuestras vidas, y cuando todo va bien prácticamente no reparamos en su presencia. Hasta que, sin motivo aparente, dos mas dos dejan de sumar cuatro. En ese instante, sucede algo extraordinario: sistemas matemáticos nítidos y claros, creados por mentes brillantes, de repente se niegan a operar correctamente. Eso es raro. Que una llave de metal abra la puerta hoy y mañana se atasque es algo que no llama la atención: un muelle se ha roto, el metal se ha desgastado, la cerradura se ha oxidado; pero que un algoritmo, que funciona mediante pasos lógicos y perfectamente bien definidos, se comporte de modo tan caprichoso es algo que no debería suceder.

¿Qué hace que la protección criptográfica deje de funcionar en un momento dado? He intentado responder a esa pregunta en este libro. Aquí va un adelanto: la diferencia entre teoría y práctica suele ser grande. Los desarrollos matemáticos que

tan claramente parecen sobre el papel se convierten en un ente con vida propia, caótico y temperamental. A veces se tienen que insertar en sistemas complejos, o bien hay que adaptarlos para que sean más rápidos, más baratos o más compatibles con otros sistemas anteriores; por no hablar de la complicación del elemento humano, cuya tontería no conoce límite. Todo conspira para hacer que los puros y perfectos métodos criptográficos que en teoría nos protegen acaben fallando.

El resultado es que la civilización tecnológicamente avanzada en la que vivimos depende en aspectos cruciales de sistemas caprichosos. Así, el PIN de su tarjeta de crédito no le protege como usted cree; las comunicaciones de su móvil pueden ser interceptadas con facilidad; las contraseñas que almacena su servicio online no están tan bien protegidas como debieran; y las redes p2p ponen a disposición de todos un conjunto de materiales audiovisuales que, en teoría, no deberían estar allí.

Cada caso es un mundo. Por eso, a lo largo de este libro he recorrido la evolución de diferentes sistemas donde la criptografía juega un papel fundamental. He procurado ser riguroso, y espero haberlo conseguido sin aburrir. Tenía dos opciones: escribir un libro donde se analizasen sistemas de cifrado desde un punto de vista teórico, con extensos desarrollos matemáticos arcanos y complicados; o compilar una lista de ejemplos, anécdotas y casos particulares sin rigor alguno. He leído libros de los dos tipos, y por lo general resultan aburridos los unos y frívolos los otros.

Mi elección ha sido un híbrido. A lo largo de este libro podrá usted seguir la evolución de diversas aplicaciones del cifrado, su desarrollo, sus fallos, sus modificaciones; pero al mismo tiempo, mi vena de profesor ha intervenido para aportar algo de rigor. No tiene sentido, en mi opinión, explicarle a usted cómo evolucionó la tecnología de protección del DVD sin enseñarle los rudimentos matemáticos básicos. No necesitará usted saber de ciencias y de informática; su experiencia como usuario será suficiente.

Por supuesto, un buen libro que se precie debería, en mi opinión, acompañarse de un conjunto de referencias bibliográficas. Eso permite al lector dos fines: ampliar información si así lo desea, y comprobar que el escritor no le está tomando el pelo. Al final de este libro encontrará más de quinientas referencias escogidas con cuidado, que espero le resulten de utilidad. Una gran ventaja de los libros electrónicos (como el que tiene usted ahora en la mano) es que permiten conexión a Internet. De ese modo, basta con hacer clic en el punto apropiado del libro y enseguida tendrá usted acceso a la referencia original. Esto es una gran ventaja respecto a los libros de papel, donde no había más remedio que buscar una biblioteca especializada para encontrar la referencia adecuada.

Espero que esta experiencia de uso le resulte cómoda, lector. Ahora le toca juzgar si mis esfuerzos han dado sus frutos. Ha llegado el momento en que este escritor deje de interrumpirle. Le deseo buena lectura.

EL ASESINO DEL ZODÍACO

Si yo llamo su atención hacia Clint Eastwood y le pregunto por un personaje de cine que haya interpretado, probablemente su primera reacción sea responder “Harry el Sucio”. Aunque Eastwood ha gozado de una larga y fructífera vida como actor y director de cine, uno de los papeles que le han dado mayor fama (con permiso de Sergio Leone y sus *spaghetti westerns*, por supuesto) fue el del inspector Harry Callahan, del Departamento de Policía de San Francisco. Quizá sin pretenderlo, su personaje de policía cínico y expeditivo creó escuela: Magnum 44, implacable con los asesinos, sin paciencia con los burócratas, azote del criminal. Alégrame el día.

La influencia de la serie de películas *Harry el Sucio* (cinco en total) es profunda. Le voy a mencionar solamente una, que me atrevería a decir que usted ha visto a menudo. Una de las opciones del navegador Google es un botón rotulado *Voy a tener suerte* (*I’m feeling lucky*, en inglés). El nombre del botón proviene de una escena de la primera película de la serie. Harry apunta con su arma a un ladrón, y le conmina a permanecer quieto o preguntarse si “se siente afortunado”. El ladrón decide no probar su suerte y aleja la mano de la escopeta. Sabia decisión.

Lo que quizá no sepa usted es que, mientras el despreciable Scorpio sembraba el terror en el San Francisco cinematográfico del inspector Callahan, un asesino auténtico apodado Zodiac hacía lo propio en la vida real. Zodiac, apodado el “asesino del Zodíaco,” cometió una serie de asesinatos en el norte de California a lo largo de 1968 y 1969. El número de víctimas confirmadas es de siete, de las que cinco murieron, aunque se tienen dudas sobre otras cuatro; el propio Zodiac se atribuyó en una misiva la muerte de 37 personas. Puede que jamás sepamos el número total, así como la identidad del asesino, quien nunca fue identificado de modo inequívoco.

Este caso tiene una vertiente cripto, y es el motivo por el que lo mencionamos aquí. El primero de agosto de 1969, los diarios californianos *Vallejo Times-Herald*, *San Francisco Examiner* y *San Francisco Chronicle* recibieron otras tantas cartas en la que el autor reconocía la autoría de varios asesinatos. Cada una de las cartas contenía un tercio de un mensaje cifrado en un sistema al que llamaremos “cifra 408” por la longitud de las cartas (408 símbolos en total). El autor amenazó con más asesinatos si no era publicada en la prensa, de modo que el público tuvo acceso íntegro a su contenido.

El 6 de agosto se reveló el descifrado del mensaje. El criptoanalista no fue un profesional, sino un profesor de instituto y su esposa. Donald y Bettye Harden lograron el éxito donde los expertos de la policía, el FBI y la Armada fracasaron. Podemos hacer un esbozo del método que utilizaron, lo que nos proporciona una interesante lección de criptoanálisis aplicado.

El primer paso consiste en averiguar, o cuando menos conjeturar, qué tipo de

cifrado se ha utilizado. Puesto que la cantidad de símbolos cifrados utilizados superaba el número de letras del alfabeto, los Harden pensaron que el tipo de cifrado podía ser una sustitución monoalfabética homofónica.

En una sustitución monoalfabética simple, cada letra es sustituida por otra, siempre la misma. Se trata de un sistema de cifra sencillo, fácil de cifrar y descifrar, que se remonta al siglo XIV. La siguiente cifra es un ejemplo real: se trata de la llamada Clave Violeta, utilizada por la Brigada Mixta 104 del Ejército Republicano durante la Guerra Civil Española:

```
abcdefghijklmnopqrstuvxyzñ
XÑUTVZHKEADORYMCILJBOFQGNS
```

Para cifrar, basta con tomar la letra del texto llano (fila superior) y sustituirla por la letra correspondiente de texto cifrado (fila inferior). De ese modo, la palabra *zodiac* se convierte en *NMTEXU*.

El problema del sistema de sustitución monoalfabética es su extrema debilidad criptoanalítica. Si aparecen bloques de dos letras iguales consecutivas, es muy posible que representen dúplices habituales en castellano, como *ll* o *rr*. La frecuencia de las letras no queda enmascarada, lo que también es un problema. Si encontramos muchas veces un símbolo concreto en un mensaje cifrado, digamos la letra X, es muy posible que represente una letra muy frecuente en el idioma (por ejemplo, la *e* en castellano).

La debilidad de la sustitución monoalfabética simple es bien conocida. Una mejora, introducida a comienzos del siglo XV, pasa por introducir símbolos adicionales para cifrar las letras más frecuentes[1]. A tales símbolos se les denomina homófonos. La cifra de sustitución monoalfabética con homófonos es un buen barrunto inicial.

Una vez supuesto el tipo de cifra, Donald Harden y esposa probaron el método de la palabra probable, que consiste sencillamente en conjeturar que el cifrador ha utilizado una palabra o expresión en concreto dentro del mensaje. Esto resulta especialmente útil para atacar el tráfico diplomático y militar, donde abundan los títulos y expresiones formales. En concreto, intentaron suerte con la palabra *kill* (matar), un término que un asesino (*killer*) tendría tendencia a utilizar para hablar de sus asesinatos (*killings*).

Kill es una palabra interesante para nuestros propósitos de descifrado: consta de una consonante poco frecuente, una vocal y una consonante que se repite. Se trataría entonces de buscar una cadena de símbolos del tipo ABCC, donde A sería un signo poco repetido a lo largo del texto cifrado. Por supuesto, es posible que la letra *l* tenga más de un símbolo, pero incluso en ese caso cabría esperar que el asesino la utilizase de modo doble en alguna parte del texto. Había que apostar, claro, a que sus habilidades criptográficas no estuviesen a la altura.

Resulta que el mensaje original incluye la palabra *kill* en cuatro ocasiones, mas otras dos donde aparece *killing* (matar, matando). Si lo unimos a términos como *will* (complemento verbal de futuro, y también significa “voluntad”) o *all* (todo, todos), comenzaremos a tener una pauta. A cada éxito, tendremos otros símbolos descifrados; si nos equivocamos, probaremos otras posibilidades.

Puesto que la “cifra 408” de Zodiac está compuesta en su mayor parte por símbolos, voy a representarlos aquí mediante letras y números arbitrarios. Veamos lo que podemos deducir. En dos partes distintas del texto cifrado se puede leer *3GSS* y *3TSS*, lo que según nuestra hipótesis implicaría que *S* podría representar la letra *l* (ele minúscula; es habitual representar las letras de texto llano en minúsculas, y las de texto cifrado en mayúsculas). Aún no podemos afirmar que la palabra que esconden sea *kill*, así que vayamos despacio. En otro lugar del mensaje, tenemos los grupos de símbolos cifrados *CB* que se repiten en varios lugares. Es posible que se trate de una sílaba cualquiera, pero justo dos lugares antes tenemos símbolos como 3T. ¿Podría ser también una representación de la palabra *kill*?

Permítanme incluir varias secciones del mensaje ordenadas de esta forma:

… 3UBSKOR …
… 3TCBPOR …
… 3PCB …
… LMRTCBPDR …
… ATCB …
… GSS …
… 3TSS …
… APBS …
… ATBS …
… ATSS …
… EDCCJEXPOR …
… QSCB …

¿Comienzan a notar un patrón? En algunos casos, aparece *3* o bien *A*, luego un signo, y después una pareja de signos tomados de entre el conjunto (*C, B, S*). Vamos a suponer que, por ejemplo, *3* es el resultado de cifrar la letra *k*, y que (*C, B, S*) son tres formas distintas de escribir la letra *l*. Por supuesto, podemos equivocarnos. Podría ser el caso, por ejemplo, que *3* represente la letra W y en realidad formen la palabra *will*. En ese caso, nuestra hipótesis sería inválida y habría que razonar otra, pero en cualquiera de los dos casos, tendríamos también una representación de la letra *i*. Y ya puestos a barruntar, fíjense en que algunos grupos cifrados terminan de forma similar. ¿Podrían representar el sufijo *ing* que indica un gerundio en inglés?

Dicho así parece fácil, pero yo cuento con la ventaja de que conozco el resultado. El proceso es más largo y tortuoso, a base de probar una hipótesis, luego otra y otra,

hasta que las piezas comienzan a encajar. Se trata de un trabajo laborioso pero básicamente sencillo, que cualquier criptoanalista profesional podría realizar sin gran dificultad. Realmente, resulta difícil imaginar cómo es posible que las agencias policiales y militares norteamericanas fuesen incapaces de descifrar el mensaje. Los Harden lo hicieron.

Nosotros, con su permiso, seguiremos con nuestra hipótesis de trabajo. Con un poco de ensayo y error, el descifrado de los fragmentos anteriores quedaría así:

```
3UBSKOR         = killing
3TCBPOR         = killing
3PCB            = kill
LMRTCBPDR       = thrilling
ATCB            = will
GSS             = all
3TSSUV          = killed
APBS            = will
ATBS            = will
ATSS            = will
EDCCJEXPOR      = collecting
QSCB            = fall
```

Si desea usted probar suerte, y ver hasta dónde es capaz de descifrar, puede intentarlo. El mensaje original (las tres partes) está disponibles en[2], y la cifra está en[3]. No diré nada de la cifra para no fastidiarle el intento, pero podrá comprobar que, efectivamente, la letra *k* dispone solamente de un símbolo para cifrar.

El mensaje propiamente dicho, en texto llano, está descifrado en[4]. Podría incluirlo aquí, pero sinceramente, es bastante desagradable y no viene al caso. Al final, resulta que el texto llano resultante contiene errores. El asesino se equivocó al cifrar algunas palabras: *forrest* por *forest* (*bosque*), *anamal* por *animal*, *thae* por *that* (*que*). Incluso con fallos, el descifrado efectuado por los Harden tuvo éxito.

Quedó, eso sí, un enigma sin resolver. Las últimas letras del mensaje se traducían como un batiburrillo sin sentido: *EBEORIETEMETHHPITI*. Tal vez sea un mensaje cifrado en un sistema diferente. Algunos lo vieron como un anagrama, e intentaron sacar de ahí la identidad de Zodiac, por ejemplo como: "*Robert Emmet the Hippie*". Tal vez se refiriese al patriota irlandés del mismo nombre, que intentó una rebelión contra los ingleses y fue ejecutado por traición en 1803; tiene varias estatuas dedicadas en los Estados Unidos, incluida una en el Parque Golden Gate de San Francisco. La referencia hippie puede parecer fuera de lugar, pero recordemos que estamos hablando del San Francisco de los años setenta visto desde la mente de un asesino. O puede que tan sólo se trate de una tomadura de pelo. Quién sabe.

Zodiac incluyó otros dos criptogramas en sendas cartas enviadas el 2 de abril y el

26 de junio de 1970 al *San Francisco Examiner*. Aunque los símbolos eran similares, el sistema de cifra había cambiado y la corta longitud de los criptogramas (13 y 32 signos, respectivamente) ha impedido hasta la fecha un descifrado claro. El primero de ellos resulta potencialmente revelador, porque viene precedido por las palabras en texto llano "*mi nombre es*". Suponiendo que se trata de anagramas, existen centenares de posibles soluciones.

Sin embargo, el gran enigma que permanece sin resolver es el de la "cifra 340". En el mensaje de 20 de abril, Zodiac preguntaba sarcásticamente "*Por cierto, ¿habéis descifrado el último criptograma que os envié?*" Se refería a una carta recibida el 8 de noviembre de 1969 en el *San Francisco Chronicle,* que incluía un mensaje cifrado de 340 signos de longitud (de ahí su apodo: cifra 340 o Z340).

En este caso, no parece que la cifra sea de tipo similar a la anterior, esto es, una sustitución monoalfabética homófona. Entre 2008 y 2011, la Universidad Estatal de San José publicó tres trabajos de Master relacionados con la cifra 340:

—*Efficient attacks on homophonic substitution ciphers* (Amrapali Dhavade, 2011). Este autor muestra una técnica generales para resolver cifras de sustitución monoalfabética con homófonos, de manera similar a como hicieron los Harden pero aplicando procedimientos informáticos modernos. Como comprobación, utilizó las dos cifras largas de Zodiac. Su resultado, aplicado a la cifra 408, mostró dos resultados posibles; ninguno era correcto, pero uno de ellos tenía suficiente información para poder descifrarse. La cifra 340 no pudo ser descifrada[5].

—*Heuristic search cryptanalysis of the Zodiac 340 cipher* (Pallavi Kanagalakatte Basavaraju, 2009). Este trabajo se centra específicamente en la cifra 340, y también supone que se trata de un sistema de sustitución monoalfabética con homófonos. El procedimiento usado es muy útil en este tipo de cifras, pero ni la cifra 340 ni la 408 pudieron ser descifradas de modo satisfactorio[6].

—*Analysis of the Zodiac 340-cipher* (Thang Dao, 2008). Al igual que en los dos casos anteriores, la hipótesis de la cifra de sustitución homofónica resultó ser infructuosa. No se consiguió una solución[7].

Como puede verse, la cifra 340 es de tipo distinto a la 408. Surge entonces la pregunta, ¿de qué clase puede tratarse? A finales de los sesenta, todavía no existía criptografía informatizada, y el estándar DES ni siquiera se había inventado. Es muy dudoso que Zodiac tuviese acceso a una máquina cifradora como Enigma, TypeX, Hagelin u otro tipo; eso dejando al margen el hecho de que esas máquinas producen un texto cifrado de aspecto aleatorio y utilizando solamente las letras del alfabeto. La repetición de los símbolos hace asimismo improbable el uso de la llamada libreta de uso único (*One Time Pad*, OTP). Hay algunos sistemas de tipo "lápiz y papel" que podía haber usado, pero su confección hubiese requerido una cantidad de tiempo y talento inusualmente grande.

Queda entonces un conjunto de sistemas de cifra. Podemos, por ejemplo, usar un sistema de sustitución para cifrar las letras impares (primera, tercera, quinta…) y otro para las letras pares (segunda, cuarta, sexta…). En este caso, cuando tenemos más de una sustitución posible, tenemos lo que se denomina un sistema de cifrado polialfabético. Podríamos suponer que Zodiac, indignado ante la facilidad con que su cifra 408 fue descifrada, echó el resto y complicó su nueva cifra hasta el límite de lo posible: polialfabetos, símbolos, quizá trasposiciones. Seguro que no utilizó una cifra tan boba como la de César.

¿O acaso sí? En julio de 2011, Corey Starliper, un aficionado a los códigos de Tewksbury (Massachusetts), afirmó haber descifrado el mensaje. Su solución es tan endeble que borda lo absurdo. En primer lugar, supuso que los símbolos se convertían en letras en virtud a su similitud geométrica. Cualquier símbolo circular, fuese hueco, macizo, con un trazo vertical, una cruz en su interior o similar, representaría la letra *o*; una cruz o una T invertida, la letra *t*; y así sucesivamente. En cuanto a los símbolos en forma de cuadrado, decidió que representaban espacios entre palabras.

A continuación, pensó que el cifrado consistía en un conjunto de sustituciones de César. La cifra original de César sustituye cada letra por la que se encuentra tres posiciones a su derecha: la *a* pasa a ser la *D*, la *b* se convierte en la *E*, y así sucesivamente. En general, podemos definir una cifra de César *n* como la que convierte una letra en la que está *n* posiciones a su derecha. Por ejemplo, esta sería la cifra de César correspondiente a *n*=5 (a=F); como anteriormente, las filas superior e inferior indican los alfabetos en texto llano y cifrado, respectivamente:

```
abcdefghijklmnopqrstuvxyzñ
FGHIJKLMNOPQRSTUVXYZÑABCDE
```

Por supuesto, usar una sola cifra de César hubiera sido evidente hasta para el más torpe de los criptoanalistas, así que Starliper probó con una sucesión de cifras de César: la primera letra con la cifra César 3, la segunda con César 4, la tercera con César 6, y vuelta a empezar (el motivo de utilizar esos números es que, de ese modo, el mensaje cifrado comenzaba por *kill*; es decir, estaba utilizando el método de la palabra probable). A este sistema se le denomina *cifra de Vigenère*, y se conoce desde mediados del siglo XVI.

La noticia saltó pronto del diario local[8] a la prensa de alcance nacional[9] y diversos foros especializados en criptografía[10]. Sin embargo, pronto se demostró que su solución no servía. La cifra de Vigenère usada por Starliper dejaba de funcionar en seguida, y tuvo que sustituirla por una con una clave mucho más larga. Básicamente, iba inventándose la solución para que encajara con las suposiciones iniciales[11]. La solución es falsa, lo que los norteamericanos llaman un *hoax*.

No fue la primera vez que se anunció una solución, ni será la última. En todos los

casos, el descifrado arroja frases típicas de un asesino desequilibrado; pero los métodos de descifrado son, en todos los casos cuestionables, las “claves” resultan inconsistentes, y las “soluciones” parecen encajar más en las ideas preconcebidas del descifrador que en patrones lógicos.

Hasta el momento, la cifra 340 permanece sin resolver. El propio caso sigue abierto. Se han barajado diversas identidades para el asesino, pero nunca se ha podido demostrar nada. El caso permanece abierto en varios condados de California.

Recientemente, una película volvió a atraer atención sobre el caso. *Zodiac*, dirigida en 2007 por David Fincher, está basada en dos libros de Robert Graysmith. La película incluye algunas referencias sobre el uso de criptografía. En una escena, el periodista que sigue el caso, en un intento por penetrar en la mente del asesino, aparece con dos libros sobre criptografía. Uno de ellos es *Codebreakers*, de David Kahn, un libro clásico sobre la historia de la criptografía; el otro es una versión en inglés del libro *Códigos y cifras*, de John Laffin.

Dudo que el protagonista pudiera sacar algo en claro de ambos libros. *Codebreakers* es un excelente texto sobre historia de la criptografía, pero tiene muy pocos ejemplos prácticos y se refieren fundamentalmente a la ruptura de códigos (criptoanálisis) y no a su creación (criptografía). En cuanto al libro de Laffin, tengo una copia en español y lamento decir que es uno de los peores de mi colección (advertencia: el libro que se ve en la película es bastante más gordo que el mío, así que a lo mejor yo tengo una versión reducida y me equivoco en mi juicio). Me resulta poco probable la afirmación de Graysmith acerca de que Zodiac podía haberlos utilizado para crear sus criptogramas, pero al menos hizo un esfuerzo por dotar al personaje de credibilidad.

En ocasiones pienso que la afición a la criptografía es como un resfriado contagioso, que engancha y no tiene cura. El propio Robert Graysmith puede dar testimonio de ello. Trabajaba en el *San Francisco Chronicle* como dibujante satírico cuando llegó la primera carta de Zodiac. Se fue involucrando cada vez más, y acabó obsesionado con el caso. Pasó trece años de su vida volcado en su propia investigación del caso, lo que le costó un divorcio y su carrera como dibujante (que le había valido una nominación a los premios Pulitzer).

Si usted es aficionado a los códigos y quiere intentar descifrar la cifra 340, permítame antes un consejo de amigo: no se deje la salud en ello. Estamos hablando de un misterio sin resolver desde hace cuarenta años. Hay cosas en esta vida por las que merece obsesionarse. Esta no es una de ellas.

12345

Contraseñas, códigos de acceso, claves. Cualquier usuario de Internet las necesita para proteger sus secretos. Son el análogo digital de las llaves. De hecho, las palabras clave y llave provienen de la misma raíz romana (*clavis*). Es lógico, puesto que en ambos casos representan la misma idea: un conjunto de átomos o bits cuya disposición permite el acceso a lugares protegidos.

Los orígenes del sistema de contraseñas se remontan a tiempos de la antigua Roma. La maquinaria de guerra romana estaba altamente organizada y reglamentada, y uno de los "protocolos" que establecieron se refería a la identificación de sus tropas. En ocasiones llega un grupo de hombres armados al cuartel, o bien una patrulla se encuentra en mitad de la noche con otra. Resulta imprescindible, por supuesto, saber si se trata de "los nuestros" o del enemigo. La identificación mediante el uniforme no vale mucho: los uniformes se pueden capturar, y en un tiempo de guerras civiles no es un procedimiento muy fiable. Incluso en la actualidad, la vieja técnica de vestir el uniforme del enemigo sigue siendo muy efectiva: en septiembre de 2012, insurgentes talibanes vestidos con uniformes del ejército norteamericano entraron en la mayor base británica de Afganistán y destruyeron ocho aviones Harrier del Cuerpo de Marines de EEUU en lo que un periodista calificó como "*la mayor pérdida de poder aéreo norteamericano desde Vietnam*" [1]

La solución escogida por los romanos fue la de proporcionar a sus soldados información que sólo los del propio bando conocen. De ese modo, un soldado que vuelve al acuartelamiento se identificará mediante una palabra clave, llamada **seña**. En el ejército español era costumbre acompañarla con el nombre de un santo, lo que dio origen a la expresión **santo y seña**. A su vez, este soldado puede tener sus dudas sobre quién le está pidiendo identificación, de forma que él puede solicitar una palabra clave para identificar al identificador: tenemos así la **contraseña**.

Incluso en la actualidad, el uso de contraseñas es muy habitual para fines de identificación. Si queremos entrar a una habitación, obtener privilegios de algún tipo o acceder a archivos informáticos, lo habitual es utilizar una de estas tres herramientas:

—Algo que poseemos (una llave, tarjeta o documento identificador)

—Algo que somos (huella dactilar, ADN u otra identificación biométrica)

—Algo que sabemos (una contraseña)

De estos procedimientos, el tercero es el más utilizado en Internet por su sencillez y comodidad. Un usuario escoge un secreto, o bien se le asigna uno, y dicho secreto se considerará prueba de identidad válida. Es habitual que exista más de una contraseña, en función de los objetivos a lograr. Mi banco online, por ejemplo, me proporcionó una contraseña de varios dígitos para acceder a mi cuenta corriente; una

tarjeta de coordenadas para cuando tengo que autorizar pagos o transferencias; un PIN para la tarjeta del cajero automático.

En la actualidad, la sociedad digital se basa de modo habitual y casi rutinario en el uso de contraseñas. Son necesarias para garantizarnos el acceso a nuestras redes sociales, acceder a nuestras comunicaciones y, en general, demostrar al otro lado de la línea que somos nosotros. Por desgracia, no es oro todo lo que reluce. En ausencia de un peligro claro, lo habitual es escoger contraseñas fáciles de recordar, y por tanto fáciles de adivinar. En tal caso prácticamente ningún procedimiento de seguridad podrá evitar un robo de información. Parafraseando a Schiller, contra la estupidez los propios dioses luchan en vano.

1) MISIÓN: PROTEGER LAS CONTRASEÑAS

Para que un sistema informático pueda dar acceso a un usuario, debemos comprobar que su contraseña es válida. Necesitamos, pues, construir y asegurar una base de datos con todas las contraseñas. Pero si un hacker consigue acceso a esa base de datos, ya no importa lo compleja que sea la contraseña de un usuario. Ese es el motivo por el que una base de datos de seguridad NUNCA debería guardar las contraseñas en "texto llano," es decir, sin cifrar. Por desgracia, no siempre se cumple esta elemental recomendación de seguridad, y los usuarios pagan las consecuencias.

En los últimos años se han sucedido los casos de robos de grandes cantidades de contraseñas, no ya unas cuantas, sino todas las que permiten el acceso a un sistema. Uno de los casos más sonados fue el de la web RockYou. En diciembre de 2009, sufrió un ataque cuya consecuencia fue el robo de la información (usuario y contraseña en texto llano) correspondiente a más de 32 millones de usuarios[2]. Los responsables de *RockYou* fueron duramente criticados por las circunstancias que se combinaron en este caso:

—A pesar de que la empresa de seguridad Imperva había avisado del ataque el día 4, la web no advirtió a sus usuarios hasta diez días después.

—La cantidad de contraseñas custodiadas en texto llano, sin ningún tipo de protección, era enorme, lo que lo convierte en uno de los mayores robos de contraseñas en texto llano de la historia.

—Las normas de seguridad de RockYou permitían crear contraseñas de tan sólo cinco caracteres, y recomendaba no utilizar "caracteres especiales" (es decir, no alfanuméricos)

—La web de RockYou permitía a sus clientes acceder a sus perfiles de otras redes sociales como Myspace, Hi5, Orkut o Facebook. Para ello, el cliente introducía las contraseñas de su perfil en dichas redes, y RockYou guardaba una copia. Sí, ha leído bien: *RockYou* guardaba contraseñas de otras redes sociales… y lo hacía en texto llano [3]

Imperva aprovechó la oportunidad para realizar un análisis de las contraseñas robadas. Las más utilizadas fueron términos sencillos como *123456*, *12345*, *123456789*, *Password* e *iloveyou* [4][5].

Me gustaría poder decir que el caso RockYou marcó un antes y un después en el mundo de la seguridad online. En cierto modo así fue, pero no precisamente para bien. La lista de candidatos a esta especie de Festival del *Epic Fail* no hace más que aumentar desde entonces. En febrero de 2012, Microsoft abrió una investigación con relación a un ataque contra su tienda online en India. Un grupo llamado Evil Shadow Team consiguió, entre otras cosas, acceder y robar una lista de nombres de clientes y contraseñas, de nuevo en texto llano[6].

De forma casi simultánea (aunque, que yo sepa, no relacionada), alguien consiguió acceso a datos sobre una gran cantidad de direcciones de correo electrónico y contraseñas de la web YouPorn, entre varios miles y más de un millón, según las fuentes. El daño potencial que puede hacer la revelación de datos de una de las cien webs más populares de Internet se une a la naturaleza íntima de sus contenidos. Ser usuario de YouPorn es algo totalmente legal y legítimo, pero a nadie le gustaría que sus hijos se enterasen.

En esa ocasión, la vía de entrada de los ladrones fue vergonzosamente fácil. Un programador de YouPorn creó un servicio de chat llamado YP Chat. Los archivos de registro (los llamados *logs*), usados por los programadores para poder comprobar fallos de seguridad, estaban ocultos a la vista, pero eran de acceso público y estaban sin cifrar[7]. Alguien buscó, encontró y se llevó los logs, que contenían datos personales de los usuarios, incluidas direcciones email y contraseñas desde al menos 2007[8]. Aunque un comunicado de YouPorn afirmó que no se habían accedido a datos de YouPorn.com[9], el hecho es que el chat estaba enlazado por ellos. Eso significa que muchos usuarios de YouPorn habrán usado las mismas contraseñas y email en YP Chat; y puede que en otros servicios totalmente diferentes, ya que la precaución "una web, una contraseña" no siempre es seguida por el usuario medio.

No les sorprenderá el hecho de que más de la mitad de las direcciones de email correspondiesen a dominios de hotmail.com, yahoo.com y gmail.com, discretas y fáciles de crear. Lo que resulta increíble es que los usuarios utilizasen contraseñas débiles para el acceso a un servicio tan íntimo y privado. La contraseña más frecuente, utilizada por un 9% del total de usuarios de YouPorn, es *123456*. La segunda más frecuente es *123456789* (por lo visto, los usuarios interpretan a su manera la sugerencia de hacer contraseñas largas). Por supuesto, no podían faltar otros sospechosos habituales como *12345*, *1234*, *password* y *qwerty*[10].

Cuatro meses después de la intrusión de YouPorn, el 12 de julio de 2012, el famoso hacker Kevin Mitnick anunció que un servicio perteneciente a la empresa Yahoo! había sido comprometido[11]. La empresa se apresuró a arreglar el fallo, que atribuyó a un antiguo fichero propiedad de la Associated Content, ahora integrada en Yahoo[12]. La autoría fue reivindicada por el grupo D33Ds, quienes publicaron en una web los nombres de 450 000 usuarios y contraseñas en texto llano[13]. Poco tardó el experto en seguridad sueco Anders Nilsson en analizar el botín. Las contraseñas más habituales resultaron ser (sorpresa, sorpresa) *123456*, *password*, *welcome*, *ninja* y *abc123*[14]. Una contraseña que llamó especialmente la atención, cuando fue mencionada en una web española sobre seguridad, era *mercadona* [15].

Y no fueron los únicos casos. Julio de 2012 fue un mes récord en las contraseñas robadas en masa:

—Androidforums.com (10/7): 1 000 000 contraseñas[16]

—Billabong.com (12/7): 35 000 contraseñas[17].

—Foros nvidia (12/7): 390 000 usuarios, número de contraseñas desconocido [18] [19].

—Pinterest (5 a 16/7): Número desconocido de usuarios afectados por un hacking contra sus cuentas que, según algunos indicios, pudo comenzar en marzo[20].

—Gamingo (6/7): 8 200 000 contraseñas, robadas en febrero y hechas públicas en julio [21][22][23]

Y la lista sigue y sigue: Philips[24], AMD[25], Blizzard[26], ITWallStreet.com[27], RevTT[28], el registro de dominios de Perú[29], el gobierno ruso[30], Adobe[31]… mejor paramos aquí. Lo que sorprende es que entre las víctimas de hacks masivos se encuentran algunas de las empresas más tecnificadas del mundo. No son unos pardillos precisamente.

¿Cómo es posible que los hackers tengan acceso a tantas contraseñas? En contra de lo que pueda parecer, los atacantes no se dedican a buscar el valioso archivo de contraseñas, copiarlo y salir corriendo a estilo Indiana Jones. No tienen que devanarse los sesos buscando privilegios de administrador, adivinando la contraseña maestra o ejecutando trucos de película. En la mayoría de los casos, se limitan a engañar al sistema mediante una técnica denominada *Inyección SQL* que pasa por "inyectar" información que el sistema interpreta erróneamente como órdenes legítimas.

Un ejemplo real, tomado de la Wikipedia, nos ayudará a entenderlo[32]. Supongamos que una página web nos pide el nombre de usuario. El código SQL usado por el programador que usó el programador es:

```
consulta: = "SELECT * FROM usuarios WHERE nombre = '" + nombreUsuario + "';"
```

En esta consulta, el sistema seleccionará los registros para los cuales `nombre` coincida con la entrada del usuario (`nombreUsuario`). Si éste teclea la palabra *Alicia*, la orden que recibe el sistema es:

```
SELECT * FROM usuarios WHERE nombre = 'Alicia';
```

Esa orden hará que la base de datos busque todos los registros con nombre Alicia. Hasta aquí todo normal. Pero supongamos que el atacante teclea esto:

```
'Alicia'; DROP TABLE usuarios; SELECT * FROM datos WHERE nombre LIKE '%
```

En ese caso, esto es lo que entenderá el sistema:

```
SELECT * FROM usuarios WHERE nombre = ''Alicia';
```

```
DROP TABLE usuarios;
SELECT * FROM datos WHERE nombre LIKE '%';
```

Ahora, en lugar de una línea de instrucciones, tenemos tres. Las dos últimas han sido “inyectadas” en el sistema, que hará esto:

—Seleccionar los registros con nombre Alicia.

—Borrar la tabla de usuarios.

—Seleccionar toda la tabla de datos (que en condiciones normales no debería estar accesible al usuario)

Es algo así como si un guardia de seguridad me interceptase en la puerta. El guardia me pregunta por mi nombre, y yo respondo diciendo “mi nombre es Arturo Dametucartera”. El guardia cree que el apellido es una orden y me entrega su cartera. Estúpido, ¿verdad? Pues así se roban contraseñas a millones. En el caso del guardia de seguridad, bastaría con contratar a alguien menos tonto, y por lo que corresponde a la inyección SQL existen formas de mitigar sus efectos. Ahora bien, si la web no ha tomado en serio este tipo de ataques (sea por desconocimiento, ignorancia, negligencia, o sencillamente para ahorrar), tarde o temprano un hacker efectuará una inyección SQL con éxito y logrará acceso a información confidencial, como por ejemplo la lista de usuarios y contraseñas.

Una solución al problema de tener un archivo de contraseñas consiste en no tener un archivo de contraseñas, es decir, no guardar las contraseñas en su forma original. En su lugar, se guardan datos que están relacionados con las contraseñas pero que no revelan ninguna información sobre ellas. Voy a ponerles un ejemplo sencillo. Supongamos que yo deseo acceder a un sistema mediante el uso de mi número de DNI. Llego a la entrada, y de repente tengo miedo de que, al sacar mi documento de identidad de la cartera, alguien pueda copiar o escuchar el número. Una opción para evitarlo podría ser comprobar tan sólo la letra final, que depende del número completo. Así, yo le digo al vigilante de la entrada “mi letra del DNI es la V,” él lo comprueba y me deja pasar. De ese modo, he demostrado que conozco un secreto sin revelarlo.

En la práctica, la letra del DNI no resulta útil para este fin. El motivo es que el número de letras es escaso y predecible. En realidad la letra del DNI se ideó con otros fines (fundamentalmente, para evitar que alguien falsifique un documento con un DNI inventado sobre la marcha). Pero la idea es buena, y podríamos utilizarla para comparar contraseñas sin revelarlas. Para ello, nada mejor que las llamadas funciones hash (una palabra inglesa que significa condensado, o destilado). Se toma un texto o mensaje cualquiera M y se le aplica una transformación representada mediante la función H, de forma que el resultado $h=H(M)$ es el destilado o hash del mensaje.

En ocasiones, el hash h se utiliza para representar un mensaje M de mucho mayor tamaño, lo que resulta muy útil, por ejemplo, en los casos de firma digital. Pero

también tienen aplicación para ocultar contraseñas, ya que las propiedades anteriormente mencionadas me permitirán asociar un único valor hash a cada contraseña.

Una empresa con conciencia de seguridad tomaría todas las contraseñas de los usuarios, sometería cada una a la función hash, y guardaría el resultado. Cuando el usuario introduzca una contraseña *C*, el sistema usará una función hash *H* y determinará su valor: *h=H(C)*. A continuación, el sistema consultará la base de datos de hashes. Si el hash que corresponde a la contraseña *C* coincide con *h*, eso significa que el usuario ha tecleado una contraseña válida. Ya no hace falta almacenar las contraseñas. En teoría, un ladrón que se llevase la base de datos de hashes no podría obtener ninguna información sobre las contraseñas.

El uso de funciones hash para protección de contraseñas debería ser política habitual en todo sistema de acceso con gran número de usuarios, pero aún hay sistemas donde no se utiliza. En el caso de Billabong.com, de julio de 2012, se reveló que las 35 000 contraseñas capturadas estaban en texto llano [17b].

Incluso los grandes cometen este tipo de errores de vez en cuando. No tienen más que preguntarle a Sony. El 20 de abril de 2011, PlayStation Network, la red de juegos de Sony, tuvo que ser desconectada durante casi un mes tras el descubrimiento que los datos personales de sus usuarios habían sido robados. El número de clientes afectados, más de 77 millones, lo convirtió en uno de los mayores casos sobre robo de información de usuarios de la historia. Por fortuna, la información sobre tarjetas de crédito estaba cifrada de modo independiente. En cuanto a la información sobre contraseñas, Sony afirmó que estaban protegidas con una función hash, aunque no dio más detalles[33]. Se ignora qué uso se dio a la información robada, pero en caso de que se hubiese recuperado información sobre las contraseñas o los números de tarjetas, se hubiera filtrado a Internet como en otros casos.

Aunque Sony incrementó la seguridad de su red de juegos, no pareció haber aprendido la lección. Acosado por grupos hackers que le habían declarado la guerra, el gigante japonés sufrió una oleada de ataques. En junio de 2011, la web de Sony Pictures sufrió el robo de la información privada correspondiente a más de un millón de personas, incluyendo direcciones de email y contraseñas en texto llano, es decir, sin protección hash alguna. El grupo LulzSec, que se atribuyó la autoría del ataque, colgó parte de esa información en el portal The Pirate Bay, fácilmente accesible para cualquier usuario de Torrent [34]. Un tercer ataque, en octubre, permitió el acceso a 93 000 cuentas de PlayStation Network y Sony Online Entertainment. En este caso, los atacantes probaron técnicas de fuerza bruta y consiguieron acceder a cuentas de usuarios con contraseñas débiles [35].

Aunque el uso de valores de hash en lugar de contraseñas aumenta la seguridad de un sistema, no es la panacea universal y existen diversos fallos explotables por un

atacante con recursos. El más eficaz sigue siendo el hecho de que algunos usuarios utilicen contraseñas sencillas de forma habitual.

Digamos que la contraseña *password* tiene como hash la cadena alfanumérica *1c24A*, esto es, H(*password*)=*1c24A*. Eso significa que, si en la base de datos de funciones hash que acabamos de robar, aparece un hash igual a *1c24A*, ya sabemos que la contraseña correspondiente es *password*.

Un atacante podría tomar un "diccionario" compuesto por las contraseñas y palabras más comunes, y aplicar la función hash H a cada una de sus términos; luego comprobaría esos valores con los que aparecen en la base de datos robada. Hay herramientas informáticas capaces de comprobar millones de hashes por segundo, como el programa John the Ripper [36], usado habitualmente por los administradores de sistemas para comprobar la fortaleza de las contraseñas de usuario.

Pueden ver ustedes un ejemplo del uso de esta herramienta, y en general del proceso de ruptura de contraseñas, en las referencias[37] y [38]. Recientemente fue modificado para aprovechar la potencia de los procesadores GPU (que controlan las tarjetas gráficas), haciéndolo especialmente útil contra las funciones hash que están diseñadas para ser lentas, y por tanto son difíciles de asaltar en un ataque de fuerza bruta [39].

Ha habido casos sonados de ataques contra bases de datos de contraseñas protegidas con hash. El caso Gamingo, en el que más de ocho millones de contraseñas fueron robadas en julio de 2012, fue un ejemplo típico. En diciembre de 2011 hubo otra intrusión. En esa ocasión, la víctima fue nada menos que Stratfor Global Intelligence, una empresa dedicada a la recogida y procesado de datos de inteligencia. Algunos la consideran una CIA privada, aunque oficialmente se dedica a realizar análisis geopolíticos: *nuestro fin es simple, hacer que la complejidad del mundo sea comprensible para un lector inteligente, al margen de ideología, agenda y prejuicios nacionales*[40]. Según parece, pretenden ser al mundo de la inteligencia privada lo que Blackwater es al de los mercenarios. No todos comparten esa opinión, y hay quien afirma que Stratfor "*es tan sólo [el diario] The Economist con una semana de retraso y mucho más caro*". [41]

Sea cual sea nuestra opinión sobre su naturaleza o fines, se supone que una agencia como Stratfor debería mantener una férrea seguridad en sus sistemas. Por dicho motivo, sorprendió que en Navidad de 2011 el propio fundador reconociese que su empresa había sufrido una intrusión informática. Los atacantes consiguieron una lista de miembros y clientes, incluyendo números de tarjetas de crédito y, como se reconoció posteriormente, casi un millón de direcciones de email y contraseñas de acceso. Las contraseñas se guardaban en forma de hash, para lo cual utilizaron la función de hash MD5. Ese robo de contraseñas fue tan sólo una parte dentro de la masiva filtración de correos electrónicos de Stratfor apodada *The GI Files* y

desvelada a finales de febrero de 2012 por Wikileaks[42].

Un ataque similar fue llevado a cabo sobre la web de juego Battlefield Heroes en junio de 2011. El grupo de hackers conocido como LulzSec afirmó haberlo hecho como despedida y cierre. La información filtrada incluía datos sobre 500 000 direcciones de e-mail, contraseña, números de teléfono y fechas de nacimiento[43]. El sistema usado para proteger las contraseñas era nuevamente la ya obsoleta función de hash MD5. El "paquete de despedida" de LulzSec incluía saqueos procedentes de otras fuentes como hackforum.net (200 000 nombres de usuario y contraseñas) e incluso la librería online de la OTAN (12 000 usuarios)[44].

La red social LinkedIn fue otro caso de libro. Sus contraseñas estaban ocultas gracias a la función de hash SHA-1 considerada más segura que MD5 no sólo por su fortaleza criptoanalítica sino también por la mayor longitud de sus valores hash (160 frente a los 128 de MD5). El 6 de junio de 2012 sucedió lo inevitable: aproximadamente 6 500 000 contraseñas de sus usuarios fueron colgadas en una página web[45]. Para ser exacto, fue el valor hash de esas funciones lo que se hizo pública, y en este caso no se incluyó información sobre cuentas de email. Se especula con la posibilidad de que los ladrones hayan publicado la lista de hashes para que otros hackers les ayudasen en su descifrado[46]. Eso significa que los 6,5 millones de hashes son los valores únicos (no repetidos). Es posible que algunos usuarios usen la misma contraseña, con lo que el número total de cuentas comprometidas puede ser mucho mayor. En un comunicado en el que reconocían el robo de información, la red social pedía a sus miembros que cambiasen su contraseña a otra que fuese difícil y rara (al menos 10 caracteres complejos)[47].

En adición a los 6,5 millones de hashes de LinkedIn, la base de datos expuesta incluía aproximadamente un millón y medio de valores hash (y lo que es peor, nombres de usuario) correspondientes a los clientes de eHarmony, un portal online de citas. La seguridad de eHarmony era evidentemente menor, ya que además de permitir contraseñas cortas usaba como función de hash MD5, considerada como inferior a SHA-1. Al igual que LinkedIn, eHarmony reconoció casi de inmediato la gravedad del ataque[48] y recomendaron a sus usuarios el cambio de contraseña a una más robusta. Para rematar la faena, también había información sobre unos 8000 usuarios de last.fm[49].

2) UNA PIZCA DE SAL

El hecho es que ni siquiera el uso de funciones hash proporciona protección suficiente para un sistema de contraseñas. Los atacantes cuentan a su favor con una gran potencia de computación, proveniente no sólo de la enorme capacidad de cálculo de los ordenadores actuales sino también del hecho de que cada vez se coordinan mejor. Una base de datos puede ser analizada por centenares de personas trabajando pacientemente durante meses, logrando resultados que una década antes eran tarea de un superordenador. Incluso puede usarse Google para buscar el hash correspondiente[50].

Afortunadamente, hay otras técnicas de defensa. El administrador del sistema puede aumentar el nivel de seguridad añadiendo "sal" al sistema. Como en las aplicaciones culinarias, la sal da sabor al guiso. En este caso, lo que se denomina "sal" es una pequeña cadena de bits que se añaden a la contraseña antes de ejecutar la función hash. De este modo, lo que guarda el sistema no es H(contraseña) sino H(contraseña+sal). Las características de las funciones hash hacen que incluso valores muy próximos de sal den como resultado valores hash muy distintos; por ejemplo:

```
H(contraseña+1)=1c24A
H(contraseña+2)=L1g3B
H(contraseña+3)=1nKA9
```

Ahora el contrincante lo tiene más difícil, ya que a cada elemento de su diccionario tendrá que añadirle muchos valores de sal hasta encontrar el correcto. Una ventaja del uso de sal es que incluso usuarios que, por azar, utilicen el mismo valor de la contraseña, tendrán valores hash muy diferentes. La base de datos de valores hash ya no tendrá valores repetidos, algo que constituye un indicativo del uso de contraseñas idénticas y frecuentes.

En el caso del robo de contraseñas en LinkedIn, los responsables tranquilizaron a sus clientes diciéndoles que "hemos incrementado nuestras medidas de seguridad mediante una capa adicional de protección técnica conocida como 'sal' para asegurar mejor su información"[51]. Un aviso posterior parece sugerir que el cambio de hash a hash+sal se llevó a cabo antes del robo de datos del 6 de junio, pero no queda claro[52]. Tampoco eHarmony usaba sal[53].

Bien utilizado, el uso de sal es una herramienta muy útil para la defensa, pero tampoco es el remedio definitivo. Una estrategia que puede utilizar el atacante es probar conjuntos de contraseñas probables, cada una de ellas con todos los posibles valores de sal. El ataque será mucho más lento que en el caso de hash sin sal, pero en la práctica es factible gracias a dos elementos. El primero es la potencia de cálculo

accesible a los atacantes: un ordenador personal con una buena tarjeta gráfica puede comprobar contraseñas al ritmo de miles de millones por segundo, y programas como el mencionado John the Ripper han sido modificados para probar grandes cantidades de valores de sal[54].

En segundo lugar, hay aplicaciones informáticas que usan poca sal. En las versiones más antiguas de Unix, la sal disponible tenía 12 bits de longitud, lo que significa que solamente puede adoptar 4096 valores distintos. Eso complica el trabajo a un atacante, pero no lo hace inviable. Los sistemas más robustos utilizan funciones hash con granos de sal más gordos, entre 48 y 128 bits; sal gruesa, si me permiten la expresión.

Todo ello hace que el ataque a un sistema que use hash y "sal fina", si no inmediato, sea al menos factible. Las contraseñas fáciles de atacar (sea por su sencillez o por su corta longitud) resistirán algo más gracias a la sal, pero incluso si se utilizan funciones hash y sal, un usuario que utilice una contraseña débil sigue siendo vulnerable. Armados con una lista de contraseñas de uso común, los atacantes no tienen más que probarlas combinándolas con todos los valores posibles de sal para obtener acceso a al menos algunas de las cuentas.

Tenemos ejemplos de ello. Los datos publicados hasta la fecha no proporcionan más información, pero una consulta al Tech Herald me ha permitido confirmar que el sistema de hash MD5 usado por Stratfor no contaba con ningún tipo de sal. Un año antes, en diciembre de 2010, la red Gawker Media sufrió un ataque en el que alguien robó la base de datos con más de un millón de contraseñas y nombres de usuario. La base de datos estaba protegida mediante hash y sal, pero los hackers la descifraron e hicieron pública[55]. Eso permitió descubrir cuáles son las contraseñas más usadas por sus usuarios: *123456, password, 12345678, qwerty, abc123*.

Dos meses después, un ataque similar a *rootkit.com* desveló las contraseñas más habituales de sus usuarios: *123456, password, rootkit, 111111, 12345678*[56]. La empresa Duo Security efectuó un análisis de la base de datos de contraseñas robada en el caso Gawker Media. Utilizando el programa John the Ripper durante unas horas en un ordenador de ocho núcleos, consiguieron extraer más de 400 000 contraseñas, de un total de 1 300 000, y eso a pesar de que el sistema de protección utilizaba sal.

El 10 de julio de 2012, la red social Formspring sufrió el robo de 420 000 hashes correspondientes a contraseñas de sus usuarios. La función hash utilizada era la SHA-256, y también empleaban sal, lo que significa que Formspring hizo bien sus deberes (salvo por el detalle de dejarse robar información)[57]. En estos casos, el sistema es parcialmente seguro porque los usuarios de contraseñas fuertes estarán protegidos.

La única defensa eficaz contra una intrusión de este tipo consiste en deshabilitar todas las contraseñas, una medida que requiere valor porque significa molestar a millones de usuarios y enviarles el mensaje de que el sistema es inseguro

(Formspring lo hizo); y, en segundo lugar, implementar en el sistema un comprobador que rechace las contraseñas cortas o débiles.

Además de ello, Formspring anunció que cambiaría la función hash de SHA-256 a bcrypt, una función mucho más útil en estos casos porque es más lenta. Sí, han leído bien. Por lo general, los criptólogos buscan funciones que no solamente sean eficaces y robustas, sino también rápidas, y las funciones hash habitualmente utilizadas se diseñan de forma eficiente. Una web con millares de accesos por segundo necesita que las funciones hash necesarias para la comprobación de las contraseñas sean rápidas y no consuman demasiados recursos.

Paradójicamente, ahora lento significa bueno. El atacante va a probar gran número de valores hash, miles de millones, y cuanto más rápida sea la función hash más eficiente será su trabajo. El objetivo ahora es que a un usuario legítimo, que introduce su contraseña para acceder al sistema, no le reporte ninguna diferencia apreciable, pero al mismo tiempo ralentice considerablemente el esfuerzo necesario para un ataque a gran escala. Por eso tiene sentido la decisión de Formspring de decantarse por bcrypt. No solamente es mucho más lenta que las del grupo SHA, sino que su grado de lentitud de ajustable: si dentro de unos años la potencia de los microprocesadores se ha hecho diez veces mayor, podemos ajustar bcrypt para que sea diez veces más lento[58].

3) LAS TABLAS DEL ARCOÍRIS

El atacante cuenta siempre con que el usuario medio presta poco interés en cuestiones de seguridad. A despecho de los avisos que emiten los administradores de sistema, muchas personas seguirán escogiendo contraseñas como *password* o *12345*. Pero supongamos por un momento un mundo ideal en el que los usuarios utilizan contraseñas absolutamente aleatorias. Eso obligaría al atacante a olvidarse de los diccionarios de contraseñas y montar con un ataque de fuerza bruta, es decir, probar todas las posibilidades. Podría crear una tabla en la que, para cada valor posible del hash, se obtenga el texto llano correspondiente. Eso nos daría directamente la contraseña correspondiente a cualquier valor de hash.

Hasta hace relativamente poco, la mera idea de construir una tabla así era algo impensable en términos de espacio y de capacidad de cálculo. Tomemos, por ejemplo, la función de hash MD5. Convierte cualquier archivo en una cadena de 128 bits de longitud. Eso nos da un total de 2^128 posibles valores de hash, lo que excede a la capacidad de cálculo de todos los ordenadores del mundo combinados. Una tabla que contuviese cada valor del hash y de la contraseña correspondiente sería imposible de almacenar.

Afortunadamente, no es necesario llegar a tanto, porque se conocen procedimientos más sencillos. En 1980, el criptoanalista Martin Hellman propuso un compromiso (*trade-off*) entre tiempo y tamaño. En lugar de mirar una sola vez una tabla gigantesca, es posible utilizar tablas precalculadas más pequeñas, a expensas de un mayor tiempo de computación. Como ejemplo, postuló un sistema capaz de obtener la contraseña correspondiente al hash generado por el algoritmo DES (que, aunque es un algoritmo de cifra, también sirve como función hash). Según sus cálculos, una tabla de 10^13 bits conseguiría hacer el trabajo en aproximadamente un día[59]. El coste de construir una máquina así sería se estimó en varios millones de dólares. Pero eso fue en 1980. Desde entonces, la capacidad de cómputo de los ordenadores ha aumentado en un factor de un millón, y delante de mi ordenador tengo un disco duro capaz de albergar 10^13 bits (poco más de un TB) sin problemas.

Esas tablas precalculadas se conocen con el nombre de *tablas rainbow* o *tablas arcoíris*, y permiten acelerar enormemente el trabajo de encontrar una cadena alfanumérica conocida su función hash. Más correctamente, reduce el tamaño de las tablas que necesitamos para conseguirlo. La técnica de las tablas arcoíris no se limita a las funciones hash, ya que tiene aplicaciones criptográficas contra sistemas WiFi e incluso de telefonía móvil.

Por ahora, vamos a suponer que no hay sal en el guiso. Digamos que tenemos el hash *301B3C*, correspondiente a la palabra en texto llano *abcxyz* (en lo que sigue, escribiré las contraseñas en minúscula y los hashes en mayúscula). Si tuviésemos una

tabla con todos los posibles valores de hash, no habría más que buscar *301B3C* y ver a qué contraseña corresponde. Pero no es posible fabricar una tabla tan grande, así que echaremos mano de un atajo.

Lo que vamos a hacer es definir una función matemática que nos relacione un valor de hash con una palabra determinada. La vamos a llamar **función de reducción**, y la representaremos con la letra *R*. Hasta cierto punto la función de reducción hace lo contrario que la función hash, pero mucho cuidado: *H(a)=b* no implica que *R(b)=a*. No son funciones inversas.

Las funciones *R* y *H* se utilizan para calcular lo que se denominan cadenas de hash. Vamos a crear los eslabones de la cadena mediante la aplicación de esas funciones, y luego guardarnos tan sólo los eslabones extremos. Por ejemplo, supongamos la siguiente cadena de cinco eslabones:

- Eslabón 1: *aaaaaa*
- Eslabón 2: H(*aaaaaa*)=*KH28AQ*
- Eslabón 3: R(*KH28AQ*)=*uygwso*
- Eslabón 4: H(*uygwso*)=*301B3C*
- Eslabón 5: R(*301B3C*)=*abcxyz*

De este modo creamos la cadena. A continuación, nos guardaremos los eslabones inicial y final (*aaaaaa* y *abcxyz*) y tiraremos el resto de los eslabones. Podemos ir repitiendo el mismo proceso para todos los eslabones iniciales que se nos ocurran. Los eslabones inicial y final de cada cadena formarán la tabla arcoíris. La columna de la izquierda representa las palabras de partida, que en nuestro caso representarán las posibles contraseñas; la de la derecha son el resultado de aplicar las funciones H y R un cierto número de veces:

```
aaaaaa abcxyz
aaaaab kzyqna
aaaaac fouqba
…
```

Fíjese el lector que la tabla arcoíris NO relaciona palabras con sus respectivos valores de hash. Esto es, H(*aaaaaa*) no es igual a *abcxyz*. Pero no importa, porque la tabla nos permitirá construir cadenas que nos revelen la información que buscamos.

Volvamos a nuestro problema inicial, a saber, ¿cuál es la palabra cuyo hash es *301B3C*? Para descubrirlo, lo primero que hacemos es aplicarle la función de reducción, y obtenemos R(*301B3C*)=*plskas*. A continuación, miramos la tabla arcoíris y buscamos *plskas* en la segunda columna. No aparece. En ese caso damos de nuevo un pasito adelante y un pasito atrás: aplicamos la función hash H y de nuevo la función de reducción R:

H(*plskas*) = *N986B1*
R(*N986B1*) = *series*

De nuevo, miramos la tabla arcoíris y comprobamos si *series* está incluida en la columna de la derecha. Si no es así, damos otra vuelta a la manivela: H y después R. Llegará un momento en que el resultado sí estará en la tabla arcoíris. Digamos que en nuestro caso *series* sí está en la lista. Lo que he hemos hecho es construir la siguiente cadena:

301B3C --> plskas --> N986B1 --> series

Buscamos la entrada *series* en la columna de la derecha, y vemos que se corresponde con la entrada *ooqksa* en la columna de la izquierda de la tabla arcoíris. Eso NO significa que H(*ooqksa*)=*series*, ya que la tabla arcoíris no relaciona palabras con sus respectivos hashes. Lo que nos indica es que, partiendo de *ooqksa*, podemos construir una cadena que acabe en el hash que estamos considerando (*301B3C*). Bien, construyamos esa nueva cadena. No hay más que ir aplicando la función hash H y la función de reducción R hasta que aparezca el hash *301B3C*. Puede ser algo de este estilo:

H(*ooqksa*) = *KI32LA*
R(*KI32LA*) = *unidos*
H(*unidos*) = *301B3C*

De esa forma, hemos creado una segunda cadena. Y ya lo hemos conseguido. Fíjense en el último eslabón de la cadena. Nos dice que, aplicando la función hash a la palabra *unidos*, obtenemos el hash que estábamos buscando: H(*unidos*)=*301B3C*. Eso significa que la contraseña que buscábamos era *unidos*.

Es posible que usted, lector, considere que esta forma de obtener contraseñas partiendo del hash es algo compleja. Hasta cierto punto lo es, y eso que he simplificado el proceso (por ejemplo, en la práctica no se utiliza una función de reducción R, sino varias). Sin embargo, esto nos permite reducir mucho el tamaño de las tablas de búsqueda. Como contrapartida, cada proceso de búsqueda deberá calcular varias veces las funciones *H* y *R*. Es un compromiso entre espacio y tiempo [60].

En cierto modo, es como si usted quisiera encontrar un documento en un gran edificio administrativo. Usted no sabe dónde está su documento, así que entra en un despacho y pregunta. No saben responderle con precisión, pero le dirigen con un “vaya a la cuarta planta, despacho 42, y pregunte allí”. En dicho despacho tampoco saben nada, así que le indican otra dirección. De ese modo, subiendo y bajando pisos,

abriendo y cerrando despachos, finalmente consigue usted encontrar lo que desea. Por supuesto, hubiera sido más fácil para usted si le hubieran indicado correctamente al principio, pero el tamaño del cartel necesario para indicar la posición de todos los documentos habría sido demasiado grande.

Una de las limitaciones de la tabla arcoíris es que está calculada para un conjunto grande pero limitado de contraseñas. Eso significa que no hay garantía absoluta de que funcione en un caso general, pero proporciona unas probabilidades de éxito muy altas. Por ejemplo, el programa para reventar contraseñas ophcrack dispone de diversas tablas arcoíris. La más grande de ellas (135 GB) puede revelar una contraseña de Windows Vista de hasta ocho caracteres alfanuméricos (letras minúsculas, mayúsculas y números) con una probabilidad de éxito del 99%[61]. Las tablas del RainbowCrack Project, son todavía mayores, de hasta 864 GB, y son válidas para las contraseñas procesadas mediante funciones de hash como MD5, así como para las de los sistemas operativos Windows 2000, XP, Vista y 7[62].

Otros proyectos, como los de Free Rainbow Tables, se basan en la participación de miles de colaboradores que donan su tiempo de cálculo[63]. Los chicos de Password Crackers no solamente le proporcionan gratuitamente los enlaces torrent a las tablas que usted desee, sino que le venden un disco duro con las tablas ya copiadas; la mayor de ellas ocupa 460 GB, de modo que no es mala idea de negocio[64].

Hay incluso servicios online para auditores de seguridad, como el ofrecido por Cryptohaze, que hace innecesario descargar las enormes tablas arcoíris. El software necesario ha sido liberado y se encuentra disponible en Internet[65]. En julio de 2012, su creador montó un sistema de computación compuesto por 31 chips GPU, utilizados normalmente en la tarjeta gráfica que controla el monitor de un ordenador, con el que logró comprobar valores hash a la increíble velocidad de 154 000 000 000 hashes por segundo[66]. A esa velocidad, podría descifrar un mensaje cifrado con el algoritmo DES en un día.

Afortunadamente, hay defensa contra los ataques de tablas arcoíris. El más sencillo es, nuevamente, utilizar sal. Si una función hash toma la contraseña del usuario y un valor único de sal para éste, el uso de tablas precomputadas es inútil, a no ser que el atacante se tome la molestia de calcular una tabla arcoíris distinta para cada valor de sal. Una de las ventajas de la sal en este tipo de aplicaciones es que no necesita guardarse en secreto, así que se suele almacenar junto al valor hash.

Atacar un sistema con hash y sal ciertamente no es un trabajo de media tarde; pero su implementación en un algoritmo de cifrado ampliamente usado hace que el número de objetivos potenciales sea enorme, y por tanto puede valer la pena el esfuerzo extra. Las tablas arcoíris disponibles en Internet, hasta donde yo sé, no incluyen valores de sal alguno; y dudo que lo hagan en el futuro, ya que las

implementaciones actuales de muchos sistemas de seguridad utilizan sal de 48 a 128 bits de tamaño, lo que reduce a cero la utilidad de las tablas arcoíris. Eso sí, mientras haya sistemas que se nieguen a utilizar el sencillo truco de la sal, seguiremos viendo casos de intrusiones informáticas masivas como los que hemos visto en LinkedIn, Stratfor o PlayStation Network.

Pero incluso el uso de “sal de grano grueso” es inútil contra un ataque de diccionario. Si los usuarios emplean contraseñas fáciles de adivinar, o si éstas son demasiado cortas, un atacante puede tomarse el esfuerzo de aplicarles todos los valores posibles de sal, y tarde o temprano conseguir el premio. Eso sí, un usuario más comprometido con la seguridad que escoja una contraseña difícil de averiguar, estará protegido. Por desgracia, todavía proliferan los usuarios que piensan que *12345* es una contraseña segura. Veamos algunos ejemplos.

4) EL USUARIO ES IDIOTA

El doctor House, célebre personaje de televisión, se hizo famoso por su carácter áspero y sus malos modales. Paradójicamente, es un modelo perfecto para muchos BOFHs y administradores informáticos, para los que el comportamiento de algunos usuarios solamente se puede entender con una de las máximas de House: el paciente es idiota. Veamos el "principio de House" en algunos de los casos que hemos ya mencionado.

El ataque contra la web de Sony Pictures en junio de 2011 permitió analizar las contraseñas que se habían filtrado[67]. ¿Adivinan cuáles eran las más habituales? Ahí van: *seinfeld*, *winner, purple*. Palabras fáciles de obtener, así como el inevitable *password* (contraseña, en inglés) y nuestros eternos amigos los seis dígitos 123456.

Podríamos pensar que los usuarios de Stratfor Global Intelligence lo hicieron mejor. Los datos nos cuentan una historia muy distinta. Tras desvelarse el robo de datos, diversos grupos se pusieron a trabajar, y volvieron a desvelar lo que a estas alturas ya se puede usted imaginar: incluso usuarios de sistemas de inteligencia, supuestamente concienciados en el campo de la seguridad, usaban contraseñas débiles. Por su parte, Stratfor no solamente no comprobaba la fortaleza de las contraseñas, sino que ni siquiera ponía un límite inferior a su longitud: en teoría, un usuario podía introducir una clave de una sola letra. Que se sepa, ningún usuario llegó a hacerlo, aunque hay muchos usuarios que utilizaron contraseñas de cuatro caracteres.

Un análisis del diario online The Tech Herald utilizó un conjunto de diccionarios y una lista de contraseñas reveladas en anteriores ataques informáticos, y los combinó con el programa *hashcat*, similar al ya mencionado Jack The Ripper. El resultado es demoledor: de los 860 160 hashes que obtuvieron para estudio, consiguieron recuperar casi 82 000 contraseñas en algo menos de cinco horas, usando un sencillo ordenador de trescientos euros. Algunas contraseñas eran tan absurdas como ****** (seis asteriscos). Hubo usuarios que utilizaron contraseñas largas, pero luego lo estropearon escogiendo términos como *111222333444*, *qwerty123456*, *lawenforcement*, *surveillance4u* o *intelligence*.

También se vio cómo algunos usuarios, en un intento por aumentar su seguridad, utilizaban sustituciones de caracteres fácilmente adivinables, como cambiar *O* por *0*, o poner @ en lugar de *a*. Eso es algo desaconsejado porque proporciona solamente una sensación de seguridad, no seguridad en sí misma. Una de las contraseñas de seis caracteres más utilizadas en el caso Stratfor era *¡@#$%^*… que no es más el resultado de pulsar *123456* con el bloqueo de mayúsculas activado. Otras elecciones poco inteligentes incluían *$intel*, *@gn0st!c* [agnóstico], *@irF0rce* [airforce] y *@tt0rn3y* [attorney]. Por supuesto, la cadena de cinco dígitos más utilizada fue

nuestra conocida *12345*, con *55555* en segundo lugar[68].

¿Qué hay de LinkedIn? ¿Podemos esperar que una de las redes sociales más extendidas del mundo haya hecho bien los deberes? Según un estudio de Ars Technica, los usuarios de LinkedIn no son muy distintos de los demás a la hora de escoger malas contraseñas. La campeona, 12345, no aparecía en la lista, pero solamente porque LinkedIn exigía contraseñas de al menos seis caracteres. Por supuesto, no les sorprenderá encontrar cadenas como 123456, 1234567 o 12345678, junto con otras creaciones más originales como *ihatemyjob* (“odio mi trabajo”) o LinkedIn. Hay quien, ante la sugerencia de escoger una contraseña fuerte (“strong password”) hizo caso literal y optó por *strongpassword*[69].

Si ha salido algo bueno en esta larga lista de robos de información, es que al menos los investigadores han conseguido información sobre el tipo de contraseñas que utiliza el usuario medio, lo que ha propiciado diversos estudios. Por ejemplo, en 2006, más de cien mil nombres de usuarios y contraseñas correspondientes a MySpace fueron robados mediante el procedimiento de *phishing* (crear una web falsa para hacer creer a los usuarios que están haciendo *login* en la página auténtica).

Bruce Schneier consiguió datos sobre 36 000 de ellos y procedió a hacer un análisis. Más de la mitad de las contraseñas tenían ocho caracteres o menos (casi el 1% no pasaba de cuatro). Más del 80% de las contraseñas estaban formadas por caracteres alfanuméricos (letras y números). De ellas, el 28% eran una combinación de letras minúsculas seguidas por un número, la mayoría de las veces un uno. Por ejemplo, *password1* (que fue la contraseña más utilizada), *monkey1*, *myspace1*, *football1, baseball1 o qwerty1*.

Parece como si los usuarios pensasen que, con añadir un número a la contraseña, eso la convierte automáticamente en segura. Otros intentaron subir la apuesta usando más números, no por ello menos predecibles. Había ejemplos como *blink182* (alusión a una banda de música llamada Blink 182) o *jordan23* (referencia al jugador de baloncesto Michael Jordan, que llevaba el número 23 en los Chicago Bulls), así como contraseñas sencillas del tipo *abc123* o *123abc*[70].

En mayo de 2012, Joseph Bonneau, de la Universidad de Cambridge, publicó el que probablemente sea el mayor estudio sobre contraseñas hasta la fecha[71]. Este investigador consiguió la colaboración de Yahoo!, y durante los días 23 y 25 de mayo de 2011 observó y guardó más de 69 millones de contraseñas de usuarios de Yahoo! Para preservar la confidencialidad de las contraseñas, éstas fueron sometidas a un hash utilizando una clave especial; de ese modo, incluso sin conocer las contraseñas directamente, se pueden hacer estudios estadísticos sobre los hábitos de los usuarios. Algunas de las conclusiones son bastante interesantes. Entre ellas podemos destacar que:

—En general, las personas de edad tienden a escoger mejores contraseñas que los

jóvenes.

—Si un usuario con una cuenta comprometida (hackeada o robada) ha completado un cambio de contraseña basado en email, tenderá a escoger una contraseña más débil; si el cambio lo hace vía página web, no suele escoger una contraseña mejor que la anterior.

—Los usuarios que cambian con frecuencia de contraseñas, así como los que se conectan desde ubicaciones distintas, suelen escoger mejores contraseñas.

—No parece haber grandes diferencias entre idiomas en lo que respecta a la elección de contraseñas fuertes o débiles.

—En general, el usuario medio escoge contraseñas demasiado débiles para proporcionar seguridad.

Una cosa que las masivas filtraciones de datos nos han dejado es una enorme lista de contraseñas. Cualquier persona interesada, sea un investigador legítimo o un hacker, puede compilar un gran diccionario de contraseñas probables. Además de las filtraciones habituales, hay muchas otras formas de conseguir diccionarios de palabras probables. Un hacker nos dio recientemente un curioso ejemplo de la utilización de Twitter para obtener términos utilizables en un ataque de diccionario[72]. El consultor de seguridad Mark Burnett utiliza Google y otros mecanismos de búsqueda para recuperar grandes cantidades de contraseñas[73]. En combinación con los diccionarios de términos comunes (con sus variantes habituales, como sustituir la O por un cero), un experto en seguridad puede utilizarlas para bloquear contraseñas débiles. Por supuesto, también en este caso el hacker podrá aprovechar para montar un ataque con mayores probabilidades de éxito.

Como puede verse, el robo de la base de datos con las contraseñas es algo cada vez más común, no diría yo inevitable pero sí lo bastante frecuente como para que tengamos que acostumbrarnos a ello. Desde el punto de vista del administrador del sistema, el uso de trucos como el uso de funciones hash y sal ya no son opcionales, sino imprescindibles.

Otras herramientas al alcance del defensor incluyen la aplicación de técnicas para evitar ataques de inyección SQL, la protección del archivo de hashes en archivos sin permisos de lectura, la aplicación de la función hash repetidas veces para ralentizar un ataque (o bien el uso de funciones hash deliberadamente lentas como bcrypt), el uso de caracteres raros… y por supuesto la exigencia de que el usuario no utilice contraseñas fáciles; entendiendo como “fáciles” aquellas que sean cortas, fácilmente adivinables, conocidas o contengan solamente letras o números. Nada mejor que un buen diccionario de contraseñas para comprobarlo[74].

No hay que olvidar un elemento muy importante en la cadena de seguridad: las preguntas de autenticación. Cuando una persona pierde u olvida su contraseña para un servicio online (por ejemplo, correo webmail), se le suele ofrecer la oportunidad

de recuperarla o bien crear una nueva. Para autenticarse, el usuario debe responder a una pregunta de seguridad que solamente él sabría responder. Por ejemplo, Google Mail permite introducir cualquier tipo de pregunta, pero también proporciona preguntas de seguridad estándar como el nombre del mejor amigo de la infancia, el primer jefe, el número de la matrícula del coche o el primer profesor.

En teoría, nadie debería saber el nombre de mi primer jefe; en la práctica, el número de posibles respuestas es grande pero limitado, y basta con que yo haya comentado su nombre siquiera una vez en Internet para que la información esté a dos clics de distancia. En el caso de personas famosas, la cantidad de información pública disponible es mucho mayor. Eso es lo que facilitó el robo de fotografías de Scarlett Johansson en 2011. Según la nota de prensa del FBI[75]:

"*Los investigadores creen que [el detenido] usó fuentes públicas para obtener datos sobre sus víctimas… una vez hubo accedido, obtuvo información privada como correos electrónicos y archivos adjuntos*".

El culpable fue finalmente detenido y se enfrenta a más de un siglo de prisión por sus delitos, pero las fotos han dado la vuelta al mundo y no hay forma de detener el daño ya hecho. Un desliz puede dar al traste con su vida personal.

Un sub-apartado interesante del mundo de las contraseñas se refiere a los números de cuatro dígitos PIN, utilizados habitualmente en teléfonos móviles, cajeros automáticos y sistemas de acceso. No hay estadísticas al respecto, pero una empresa de "minería de datos" llamada DataGenetics tuvo una brillante idea: ¿y si recurrimos a las bases de datos ya filtradas y escogemos las contraseñas con cuatro dígitos? Eso hicieron, y los resultados son interesantes.

Antes, sin embargo, un punto de precaución. Los datos utilizados provienen de bases de datos con contraseñas usadas en servicios web, y no hay garantía de que los usuarios las utilicen como números PIN con la misma frecuencia. Sin embargo, nos proporciona una muestra significativa. Se da la circunstancia de que muchos usuarios (demasiados, me temo) utilizan la misma contraseña en diversos servicios, así que resulta verosímil que la misma contraseña de cuatro dígitos sea usada en sistemas con PIN. Por otro lado, los bancos y compañías telefónicas proporcionan números PIN aleatorios. El usuario, por supuesto, puede cambiarlo, pero habrá muchos usuarios tipo "1234" que utilicen una contraseña más aleatoria por simple dejadez o ignorancia. Por otro lado, si un usuario quiere escoger una contraseña boba, lo hará de un modo u otro. En cualquier caso, veamos lo que DataGenetics tiene que decirnos[76].

La muestra fue de 3,4 millones de números de cuatro dígitos, y como cabía esperar a estas alturas, los números escogidos son de todo menos aleatorios. Seguro que ya se lo espera, pero se lo diré de todos modos: el PIN más frecuente es 1234. Lo que resulta sorprendente es la cantidad de usuarios que escogieron este número: uno

de cada nueve. Me parece sencillamente asombroso, aunque visto lo visto, quizá no tanto. Las siguientes elecciones populares son 1111 (6% de los usuarios), 0000 (2%), 1212 (1,2%) y 7777 (0,75%). Esto quiere decir que un ladrón que solamente dispusiese de tres intentos tendría una probabilidad de éxito de casi el 18%.

La vagancia se impone, ya que los diez números PIN con los cuatro dígitos repetidos aparecen entre los veinte más usados. También hay otras combinaciones sencillas de recordar, como 4321, 1122, 6969 y similares. Los pares de números repetidos, del tipo *abab*, representan casi el 18% del total de elecciones. En el fondo de la lista, el PIN menos escogido fue el 8068, seguido por 8093 y 9629. Aparentemente sucede como con los números de la lotería, los hay bonitos y feos, y por algún motivo al pobre 8068 le ha tocado hacer de feo.

Hay números que no parecen tener un origen sencillo. Es el caso del 2580, que aparece el 22º en la lista. No parece ser de los facilones, pero busque un teclado de cajero o de teléfono y se dará cuenta del motivo: los dígitos 2, 5, 8 y 0 forman una línea vertical en el teclado. También es un misterio el 1004, que aparece en sexta posición. Hay quien afirma que el motivo es cultural: en coreano, 1004 suena similar a la palabra *cheonsa* (ángel).

Es interesante comprobar que los números cuyas dos primeras cifras sean 19 son bastante más probables que otros. Eso se debe a que la gente que los usa los asocia a años. Incluso se pueden intentar extraer datos demográficos: los “años” más utilizados como PIN son 1984, 1985 y 1986. Parece que la gente en los veintitantos tiene mucho que aprender en cuanto a seguridad informática. También son bastante frecuentes los números que representan una fecha en formato mes-día (habitual en los países anglosajones). Otra observación: por lo general, los usuarios prefieren números que comiencen por cero o uno.

Si quiere un consuelo, piense que ni siquiera los usuarios expertos se libran de meter la pata. En septiembre de 2012, se filtraron los nombres y contraseñas de más de 100 000 miembros del IEEE, una de las mayores organizaciones de ingenieros electrónicos del mundo: empleados de Apple, Google, la NASA, Oracle, Samsung, los mejores de los mejores[77]. Los datos estaban en un servidor FTP accesible a todo el mundo, y las contraseñas estaban en texto llano, lo que ya es malo de por sí.

Lo peor vino cuando se efectuó un análisis de las contraseñas usadas. La ganadora fue *123456*, usada 271 veces, seguida por *ieee2012*, *12345678*, *123456789* y *password*. La octava más popular era *123*, ¡una contraseña de tres dígitos! La situación era tan embarazosa que la IEEE, en el mismo comunicado en que admitía el error y pedía disculpas, se vio obligada a recordar a sus afiliados la necesidad de crear contraseñas fuertes y no reutilizarlas en otros lugares[78].

Decididamente, no podemos confiar en el usuario. Veamos ahora el otro lado. Porque seguro que las empresas lo hacen mejor, ¿verdad?

5) LA EMPRESA TAMBIÉN ES IDIOTA

Si el usuario medio no sabe crear o gestionar contraseñas de forma segura, al menos le queda el consuelo de saber que también algunas entidades con muchos más recursos y conocimientos fallan estrepitosamente en ocasiones. Recuerdo hace años haber visto un anuncio televisado de una conocida empresa de sistemas de seguridad (no les diré cuál). Entre imágenes de casas idílicas con niños jugando en el suelo, nos intentan convencer de lo seguro que dormiremos con un buen sistema de alarma. El padre, sonriente y confiado, teclea la contraseña del sistema de alarma. ¿Y cuál es el número que escoge? ¡La fecha de cumpleaños de su hijo! Evidentemente el usuario no tiene el monopolio de la tontería. Ni mucho menos.

Comencemos con un guión típico de película de acción. ¿Qué tal si envenenamos el suministro de agua de una gran ciudad? En septiembre de 2009 se descubrió que cualquier persona con un móvil y conexión Bluetooth podía haber obtenido acceso al sistema de control y suministro de agua potable de Oslo. La contraseña no era problema, ya que había sido fijada como 0000[79].

Sigamos con los argumentos de película. ¿Qué le parece si nos atrevemos a controlar un sistema de comunicaciones militares? Podría pensarse que los hombres y mujeres de uniforme conocen mejor que nadie el valor que se debe otorgar a la seguridad en las comunicaciones, pero no siempre se aplican su propia medicina. Los motivos pueden ser muchos, como por ejemplo las limitaciones presupuestarias, pero en más de una ocasión se trata de un viejo y obsceno pecado que los militares cometen una y otra vez: minusvalorar al enemigo. Los norteamericanos, empeñados en una guerra continua que se extiende por varios países musulmanes, parecen creer que sus adversarios apenas son capaces de usar el mando a distancia del televisor.

Sólo eso puede explicar que, por ejemplo, los famosos aviones sin piloto (los “drones”) envíen a la base información no cifrada sobre lo que captan y graban. Esta vulnerabilidad se conoce desde los años noventa, cuando se hizo patente durante la guerra de Bosnia. Un enemigo vigilado por estos pájaros metálicos podría utilizar la propia grabación de video que están emitiendo para saber cuáles serán sus próximos objetivos y evitar el ataque; una transmisión de un dron podría incluso revelar información sobre las unidades propias. No es una posibilidad remota: en julio de 2009, las fuerzas norteamericanas en Irak detuvieron a un militante chií que almacenaba videograbaciones captadas por vehículos no tripulados[80].

¿Nos vamos al espacio? Apuesto a que piensa que hackear satélites militares es algo que un civil solamente puede ver en las películas de James Bond. Pues agárrense, porque vienen curvas. Durante los años ochenta, la armada de EEUU utilizó un sistema llamado FLTSATCOM (*Fleet Satellite Communications System*) para comunicarse con buques y submarinos por todo el mundo en frecuencias de

UHF. Seis satélites desplegados en órbita geoestacionaria lo hacían posible (otros dos quedaron dañados durante el lanzamiento). Durante los años noventa, fueron reemplazados por otra generación de satélites, pero a pesar de que se diseñaron con una vida útil de cinco años algunos de ellos siguen funcionando.

El problema es que los transpondedores de los FLTSATCOMs carecían de protocolos de autenticación o cifrado. Como consecuencia, cualquiera con conocimientos y equipo técnico adecuado puede acceder a ellos y utilizar sus canales como si de un satélite de comunicaciones se tratase. Un lugar donde se aprovecharon particularmente de ello es Brasil, un vasto país en expansión donde la cobertura de las redes telefónicas móviles deja mucho que desear. A comienzos de la década del dos mil, se comenzó a ofertar estos servicios piratas a usuarios de todo tipo, desde camioneros a leñadores ilegales, por no hablar de traficantes de droga. Una operación conjunta de las autoridades brasileña y norteamericana atajó el problema temporalmente en 2009, pero para entonces miles de personas sabían cómo construir sus propios transpondedores[81].

Podría darles más ejemplos de ese tipo, pero si quiero asustarles con historias sobre mensajes transmitidos sin cifrado, creo que lo haré mejor tocándoles el bolsillo, en la zona donde guarda el móvil. Un famoso FAIL tuvo como protagonista el popular servicio de mensajes WhatsApp. El protocolo usado para enviar información entre usuarios se llama XMPP, que incluye el protocolo de seguridad TLS para transacciones electrónicas seguras. El problema radica en que estaba configurado por defecto para NO utilizar cifrado, con lo que todos los datos intercambiados estaban en texto llano, accesibles a cualquiera con programas para capturar paquetes de datos[82].

Tras una serie de críticas acerca de su seguridad, la red de mensajes WhatsApp anunció que en lo sucesivo los mensajes de sus usuarios estarían cifrados[83]. Ahora bien, surgió el problema de siempre: ¿qué claves utilizar, y sobre todo, cómo repartirlas entre todos los clientes? La solución de WhatsApp para los móviles con sistema operativo Android fue tomar el IMEI (un código de identificación único para cada teléfono), invertirlo y aplicarle la función hash MD5; en el caso de otros dispositivos, utiliza la dirección MAC (identificativa del router). El resultado es la clave de cifrado. En cuanto al nombre de usuario, es sencillamente el número de teléfono[84].

Con un sistema tan sencillo, WhatsApp permite generar tráfico cifrado y evitar los problemas derivados de diseminar millones de contraseñas. Por supuesto, al lector no se le escapará que el sistema de generación de claves es altamente inseguro. Cualquier persona con acceso a un teléfono ajeno puede obtener su IMEI sin más que pulsar la secuencia *#06#. Un atacante más sigiloso puede crear una aplicación que filtre ese dato, o bien atacar directamente a la empresa telefónica y robarle su base de

datos de usuarios.

Las desgracias criptográficas de WhatsApp no acaban aquí. El programa almacena en el dispositivo del usuario gran cantidad de información de forma insegura. Datos como los "logs" de todos los mensajes enviados y recibidos, la información de los contactos y hasta las coordenadas de geolocalización (si el GPS está activado) se guardan en dos archivos sin cifrado en absoluto[85].

Para rematar la faena, la versión para Android guarda un fichero adicional que funciona como copia de seguridad (backup). Se supone que ese fichero está cifrado con el algoritmo AES mediante una clave de 192 bits. Sólo hay dos problemillas. El primero es que la clave está insertada en el propio paquete de software de WhatsApp. Y el segundo es que todos los móviles Android del mundo utilizan la misma clave. Ni siquiera añaden algún factor dependiente del móvil[86]. Esta es la clave:

```
346a23652a46392b4d73257c67317e352e3372482177652c
```

Evidentemente, no es una contraseña fácil de adivinar, pero ahí la tiene. A partir de ahora, la seguridad que proporciona es la misma que la de toda contraseña una vez descubierta y revelada: cero. Que conste que yo soy un usuario —hasta ahora satisfecho— de WhatsApp, pero a la vista de los garrafales fallos de seguridad que han perpetrado últimamente soy muy cauto a la hora de utilizarlo. Si usted transmite información mínimamente confidencial por WhatsApp, cruce los dedos.

Tampoco el gigante informático Apple ha permanecido inmune a los problemas de contraseñas y códigos de identificación. Justo unos días después de que WhatsApp anunciase su nuevo servicio de cifrado, Apple anunció que le habían robado más de un millón de números UIUD, que son unos códigos que identifican de forma única cada iPhone e iPod[87]. El grupo hacker que los robó afirmó que los había extraído del ordenador de un agente del FBI[88], algo que fue prontamente desmentido por la agencia federal. La fuente real de la filtración fue la editora digital Blue Toad, quien reconoció el robo de la información[89].

¿Por qué Blue Toad tenía esos datos? Yo también me lo pregunté. Por lo visto, uno de los servicios que proporciona es "soluciones de contenido digital y aplicaciones a editores y creadores de contenido". Cuando una aplicación se instala en el móvil o la tablet, suele requerir ciertos permisos. Algunos de ellos incluyen la conexión a Internet y la recopilación de ciertos datos, que para los vendedores representan una verdadera mina de oro. En cierto modo, esas bonitas aplicaciones "gratuitas" que nos descargamos ya las estamos pagando con nuestros datos.

Por añadidura, en mayo de 2012 se anunció la existencia de un fallo en el sistema Filevault de Apple, un sistema de cifrado para proteger la información del ordenador. Al parecer, cuando la compañía de la manzana lanzó la versión del sistema operativo MacOX Lion 10.7.3, cometió un error de programación en virtud del cual se

guardaba una copia de la contraseña en el disco duro, en un archivo de registro (log) sin proteger[90]. Apple reconoció el error y lo corrigió en la siguiente versión de Lion, la 10.7.4[91].

Peor incluso lo tuvo Yahoo! En mayo de 2012, presentó al mundo su programa Axis, un navegador para dispositivos móviles que también incluye extensiones para poder ser utilizado como buscador en los navegadores tradicionales. Cuando un investigador instaló la extensión Axis correspondientes al navegador Chrome, descubrió que Yahoo! había cometido un error de novato. Se supone que en su interior habría una clave pública que permitiría verificar la autenticidad del paquete de software; pero en lugar de ello, lo que aparecía era la clave privada, ¡la que se utiliza para crear la firma! Con esa clave privada, cualquiera podría firmar una aplicación maliciosa[92]. Es un perfecto ejemplo de EPIC FAIL, que podemos traducir como “cagada cum laude”. Yahoo! tardó muy poco en reconocer el error y cambiar la extensión Axis. El servicio de alertas de INTECO-CERT clasificó este fallo de seguridad como de “importancia 2-baja”[93], piadosamente baja, opino yo; el ridículo que hizo Yahoo! será algo que van a tardar en olvidar.

Incluso si una empresa tiene experiencia y determinación en el campo de la seguridad, algunas veces las prácticas que imponen a sus clientes rozan lo absurdo. Cualquier sistema de seguridad que se precie debería rechazar las contraseñas débiles, sea porque son conocidas o porque tienen pocos caracteres. Bien, digamos que usted está concienciado con la seguridad. Escoge una contraseña fuerte y larga, se dispone a utilizarla, y resulta que el sistema la rechaza ¡por ser demasiado larga!

Parece increíble, pero así es. Gmail impone un límite de 200 caracteres, lo que puede parecer razonable dado que se trata de introducir una contraseña, no de escribir un libro. Yahoo! tiene unos límites de entre 6 y 32 caracteres. Vale, aunque pienso que 6 caracteres es demasiado corto. Outlook prohíbe contraseñas de más de 16 caracteres[94]. Los usuarios del antiguo servicio de Microsoft Hotmail se enteraron entonces de que sus contraseñas, fuesen de la longitud que fuesen, eran secretamente “podadas” si superaban los 16 caracteres de longitud[95].

No fue la única metedura de pata de Microsoft. Las versiones de Windows anteriores a Windows NT utilizaban un sistema llamado Lan Manager (LM) para almacenar las contraseñas de los usuarios en forma de hash. El problema es que la contraseña, cualquiera que fuese su longitud, se truncaba y solamente se tomaban sus primeros 14 bytes. Es decir, la longitud máxima de la contraseña era de 14 caracteres; si era menor, se rellenaba el resto con caracteres nulos. Para calcular los hashes, las letras minúsculas de la contraseña se convertían en mayúsculas. Luego se dividía la contraseña en dos segmentos de 7 caracteres de longitud, y se aplicaba la función hash a cada segmento por separado.

¿Qué quiere todo esto decir? Pues que una contraseña como

"*EStoYdisfrUTANndoEsteLIbro*" se convertía en "*ESTOYDI*" y "*SFRUTAN*". La seguridad de una ristra de 25 letras mayúsculas y minúsculas pasa a ser la de dos grupos de 7 letras minúsculas. Peor aún, si escogemos "*quieroleer*" el sistema lo Internet como "*QUIEROL*" y "*EER* "lo que hace que la segunda contraseña (tres letras y cuatro espacios nulos) sea extremadamente fácil de extraer por fuerza bruta. Personalmente, me resulta increíble que durante varios años las mejores y más actualizadas versiones de Windows hayan incluido un sistema tan ridículamente debilitado. Afortunadamente, Microsoft deshabilitó esta opción y posteriormente sustituyó Lan Manager por el sistema de protocolos NTLM (NT Lan Manager), pero a la vista de la cantidad y tamaño de las tablas arcoíris para NTLM que circular por la Red, yo tendería a mostrarme cauto.

Microsoft volvió a meter la pata en el asunto de las contraseñas a finales de 2012, en un entorno diferente. La empresa de Redmond decidió regalar a sus clientes de Windows 8 el paquete informático multimedia Microsoft Media Center. El usuario descargaba el paquete y pedía a Microsoft una clave, que recibía a vuelta de correo electrónico. Algunos usuarios, que solamente tenían una copia temporal de evaluación del sistema operativo, descubrieron que esa clave no solamente activaba el Media Center, ¡sino todo Windows 8! De ese modo, quienes se descargasen una versión de evaluación de Windows 8 podía registrarlo con la clave de Media Center.

Se da la circunstancia de que para aprovechar este fallo, el usuario debe tener ya instalado Windows 8; y puesto que Microsoft no concede el período de gracia habitual con fines de evaluación, el truco de Media Center no es en teoría posible. Sin embargo, el usuario puede activar Windows 8 temporalmente gracias a un sistema denominado Servicio de Administración de Claves (KMS), creado para entornos empresariales[96].

Volviendo a España, me surge una duda: ¿por qué El Corte Inglés solamente acepta contraseñas de entre 6 y 8 caracteres? Sí, ha leído bien, 6 a 8[97]; eso a pesar de que la web de El Corte Inglés afirma estar preocupada por la seguridad:

"*Conforme a nuestra garantía de seguridad y confidencialidad, en El Corte Inglés estamos especialmente interesados en ofrecer a nuestros clientes el más alto nivel de seguridad y proteger la confidencialidad de los datos que nos aportan. Por ello, las transacciones comercialesson realizadas en un entorno de servidor seguro bajo protocolo SSL (Secure Socket Layer) y todas las comunicaciones se transmiten encriptadas bajo un cifrado de 128 bits,que asegura el mayor nivel de protección a las comunicaciones …*"[98].

Curioso como soy en estos casos, pregunté a la empresa por los motivos. Después de dos semanas, recibí respuesta. No puedo incluírsela, ya que venía acompañada de la típica advertencia de confidencialidad que prohíbe su divulgación no autorizada. Aun así, creo que no les sorprenderé si les digo que no me aclaró mis dudas. Volví a

preguntar. No entendía por qué poner un límite superior tan bajo a la longitud de las contraseñas, e impedir el uso de símbolos no alfanuméricos, era considerado un buen compromiso de seguridad. Mi petición fue enviada "al departamento correspondiente". A fecha de hoy, sigo esperando respuesta.

El Corte Inglés puede al menos consolarse: la operadora norteamericana de telefonía móvil Virgin Mobile USA les ha superado en tontería. Estos genios de la seguridad no solamente han fijado la longitud de las contraseñas a seis, sino que los caracteres han de ser necesariamente dígitos[99]. La única concesión a la seguridad por su parte era que no se admitían más de tres dígitos iguales o consecutivos, es decir, nada de 222 o 678.

Aun así, probar casi un millón de contraseñas impunemente ha de ser difícil, ¿no? No solamente será lento, sino que seguro que nos detectarán en seguida. Puede que tras un número de intentos, el sistema se cierre, como es el caso de los cajeros automáticos. No serán tan tonos, ¿verdad?

Preguntemos a Kevin Burke, suscriptor de Virgin Mobile USA y experto en comunicaciones. Para demostrar lo fácil que es entrar en el sistema, Burke efectuó un ataque de fuerza bruta. Nada de cracks, tablas de contraseñas o técnicas sofisticadas: se limitó a crear un pequeño programa (*script*) que probaba una contraseña por segundo. El propio Burke fijó ese límite tan lento para no sobrecargar los servidores de Virgin USA. Cuando encontró la contraseña correcta, se limitó a introducirla y entrar en su propia cuenta sin problemas.

Burke nos cuenta en su web lo que sucedió después. Tras lograr el éxito el 16 de agosto, intentó contactar con representantes de Virgin Mobile USA, ¡para lo cual le exigían su número de identificación secreto! Tras un mes de intentos infructuosos, se hartó y publicó todo el asunto en Internet[100]. Finalmente, la empresa adoptó la decisión de limitar el número de intentos antes de bloquearse. Buena idea… que no sirve de nada, porque se basa en el uso de cookies, fáciles de borrar. Ahora el sistema se cierra tras 20 intentos fallidos. Eso sí, son veinte intentos por dirección IP. Un atacante con una IP dinámica podría irla cambiando cada 19 intentos, lo que ralentizaría el ataque pero no lo detendría.

Podemos terminar esta sección con un apartado que podríamos denominar "contraseñas que se dejan encima del piano". En 1983, una película llamada *Juegos de Guerra* acercó el mundo de la informática y el hacking al gran público. Una de las actividades favoritas del protagonista era cambiar sus notas. ¿Cómo lo conseguía? Fácil: cada vez que lo llamaban al despacho del director, se pasaba por la mesa de la secretaria… que guardaba una copia de la contraseña de acceso en un post-it bajo la mesa. Les sorprendería saber cuántas veces esta resulta ser la mejor vía de acceso a un sistema protegido por contraseñas.

El Reino Unido fue testigo de un ejemplo así, de la mano de nada menos que el

Príncipe Guillermo, Duque de Cambridge y segundo en la línea de sucesión al trono de Inglaterra. En noviembre de 2012, su página web incluyó fotografías de su trabajo como teniente de vuelo en el Servicio Aéreo de Rescate de Gales, en la base de la RAF de Anglesey. Una de ellas muestra a Guillermo y un sonriente grupo de pilotos descansando; en la pared del fondo, una hoja de papel muestra los datos de acceso de un sistema militar, incluidos nombre de usuario y contraseña. Las imágenes fueron prontamente "saneadas," pero no antes de que un reportero del diario *The Guardian* publicase la noticia[101].

Durante varios días, las imágenes oficiales resultaron inaccesibles ("*debido a la demanda popular*"), y cuando volvieron a ser visibles la hoja con la contraseña había sido modificada para que no revelase información confidencial[102]. Los diarios, por su parte, tuvieron la sensatez de borrar los datos confidenciales de la fotografía publicada. Según el periodista del *Guardian*, "*me decepcionó descubrir que la contraseña era extremadamente obvia, fácil de adivinar y —francamente— un tanto diabólica*"[103]. Por cierto, que no es la primera vez que se desvelan de esta forma contraseñas fotografiadas accidentalmente[104], ni tampoco la segunda[105]. Moraleja: si quiere mantener un secreto, comience por no dejar que lo fotografíen.

6) LOS CÓDIGOS DEL ARMAGEDÓN

El uso de contraseñas tontas forma ya parte del folklore de Hollywood. Sheldon Cooper, el protagonista de la popular serie de TV *The Big Bang Theory*, no encuentra gran dificultad en acceder a la cuenta de Facebook de su amigo Leonard: "*usas la misma contraseña para todo, Kal-el*". Su vecina puede acceder a la wifi sin más que pedirle la contraseña, que suele ser del tipo *pennyesunagorrona*. Y cuando Sheldon decide hacer doblete en una tienda de informática, descubre que el ordenador de ventas es fácilmente accesible: "*puede entrar hasta un niño, 1234 no es una contraseña muy segura*".

En la película-parodia *La Loca Historia de las Galaxias*, el planeta Druidia protege su atmósfera mediante un escudo que tiene el código 12345. Cuando el malvado planeta Spaceballs consigue el código, su presidente no sale de su asombro: "*¿12345? ¡Es asombroso, yo tengo la misma combinación en mis maletas!"*

El presidente de Spaceballs nos parece un tonto redomado, y de hecho fracasó en sus malévolos planes (a pesar de tomar la precaución de cambiar la combinación de sus maletas), pero en el planeta Tierra no lo hacemos mucho mejor. En febrero de 2012, el grupo Anonymous hackeó la cuenta de correo electrónico del presidente sirio Bashar al-Assad. Más concretamente, entraron en el servidor de correo del ministerio sirio de asuntos presidenciales y accedieron a los mensajes de 78 cuentas distintas de correo electrónico[106]. Lo crean o no, la contraseña correspondiente a un tercio de esas cuentas era 12345[107]. Otros usuarios creyeron estar protegidos con contraseñas como *mopamopa, iloveyou, 25, mopa2012, 123vivasyria, 123456*… y mi favorito de todos, *sonyroot*, en aparente alusión al "rootgate" de Sony (que describo en el capítulo "Descifrando a Nemo" de este mismo libro).

Parece difícil de superar, pero el gobierno griego aceptó el reto. En noviembre del mismo año, hackers de Anonymous accedieron al Ministerio de Finanzas de Grecia, obteniendo cuentas de email y contraseñas[108]. Si lo que encontraron es sintomático de la situación económica allí, mucho me temo que los griegos lo tienen crudo. ¿Adivinan qué contraseña se repitió el 37% de las veces? No, no fue 12345… sino 123456. Llega uno a pensar que el Ministerio proporcionó a sus empleados 123456 como contraseña por defecto. En la escala Spaceballs de contraseñas tontas, los funcionarios griegos se sitúan marginalmente mejor que sus colegas sirios, pero hay de todo, desde el que escogió 123 a los que utilizaron como contraseña su mismo nombre de usuario.

Otro de los clichés de Hollywood trata de la contraseña por excelencia: la que protege al mundo del Apocalipsis nuclear. Los códigos de lanzamiento de misiles nucleares siempre acompañan al presidente norteamericano dondequiera que vaya, algo necesario durante la Guerra Fría, cuando una aniquilación nuclear podía ser tan

sólo cuestión de minutos.

En 1983, dos películas llevaron al público norteamericano visiones muy diferentes de la disuasión nuclear. Una de ellas, *El Día Después*, mostraba los efectos de un ataque nuclear contra los Estados Unidos, con un realismo y verosimilitud nunca vistos hasta entonces. Cinco meses antes se estrenó *Juegos de Guerra*, en la que un adolescente casi provoca una guerra nuclear accidental con su ordenador personal. Ambas películas tuvieron una profunda influencia en su época: la primera inició un intenso debate sobre la disuasión nuclear, y la segunda supuso el punto de arranque de la generación hacker.

En *Juegos de Guerra*, un ordenador de defensa intenta lanzar un ataque preventivo, lo que le exige utilizar su potencia de cálculo para determinar cuál es la clave de lanzamiento de misiles. Los humanos consiguieron detener el ataque por la mínima. Menos suerte tuvieron los humanos en *El Día Después*, que narra las consecuencias de un intercambio nuclear masivo debidamente autorizado.

La necesidad de un código especial para lanzar un ataque nuclear o para detenerlo es una constante en las películas sobre la guerra fría, desde *¿Teléfono Rojo? Volamos hacia Moscú* (1964) hasta *Marea Roja* (1995) y *Pánico Nuclear* (2002). Durante la Guerra Fría, tanto Estados Unidos como la Unión Soviética tenían el mismo problema de control, a saber, asegurarse de que un arma nuclear sea utilizada cuando lo determine la autoridad correspondiente (control positivo), y evitar su uso en casos no autorizados (control negativo).

Este último caso es especialmente preocupante. Un país armado con bombas nucleares puede entrar en una fase de inestabilidad, como sucedió en la China de Tiananmen, la Unión Soviética tras su caída o el Pakistán de nuestros días. Durante la revuelta de los generales contra De Gaulle en 1960, el gobierno francés tuvo que ordenar la detonación de un arma nuclear en Argelia para evitar que cayese en manos de los militares sublevados.

Más allá de las tramas sobre generales rebeldes y vengativos, lo cierto es que Estados Unidos tenía desplegadas centenares de armas nucleares en otros países, algunos de los cuales no eran muy fiables. Por ese y otros motivos, en junio de 1962 el presidente Kennedy firmó la orden NSAM-160, que ordenaba la instalación de los llamados Enlaces de Acción Permisivos (PAL, *Permissive Action Links*) en todas las armas nucleares norteamericanas.

Incluso en nuestros días, ignoramos cómo son los PAL en detalle. Se sabe que incluían dispositivos electromecánicos, combinados con sistemas criptográficos, y estaban diseñados de tal forma que la bomba no podría efectuar una explosión nuclear deliberada sin los códigos correspondientes. Otro conjunto de mecanismos adicionales evitaría una detonación accidental.

Los PAL tardaron en instalarse en todas las armas nucleares. Primero hubo que

vencer la resistencia de los militares, y luego hubo que instalarlas en millares de bombas de todo tipo, desde misiles intercontinentales a piezas de artillería táctica. En 1974, durante el enfrentamiento entre Grecia y Turquía por el control de Chipre, EEUU comprobó que las armas nucleares tácticas estacionadas en aquellos países no tenían salvaguardias ni códigos de bloqueo.

Al menos, los misiles ICBM, pieza clave del sistema de disuasión, estaban a salvo. Mientras *Juegos de Guerra* entretenía a millones de norteamericanos y *El Día Después* les aterraba, Estados Unidos tenía 2100 armas nucleares a bordo de un millar de misiles balísticos intercontinentales, con una potencia explosiva combinada setenta mil veces superior a la bomba de Hiroshima. Todos y cada uno de esos mil misiles tenían exactamente el mismo código de ocho dígitos. Un código que no fue cambiado en veinte años. El escritor Bruce Blair, que estuvo destinado en un silo de misiles Minuteman durante los años setenta, reveló en 2004 el código del armagedón nuclear[109].

El código era 00000000.

Se lo voy a repetir, por si no lo ha leído bien. Durante los momentos álgidos de la Guerra Fría, el código que hubiera lanzado setenta mil Hiroshimas contra la Unión Soviética era 00000000. En 2004, Blair le comunicó este hecho a Robert McNamara, quien fuera Secretario de Defensa durante los años sesenta. Su respuesta textual fue "*Estoy asombrado, absolutamente asombrado y furioso. ¿Quién diablos autorizó eso?*" Nadie sabe quién ordenó el uso de ese código.

Pero incluso tan estúpida elección de contraseñas palidece frente a la respuesta soviética. Uno de los problemas que las fuerzas de misiles estratégicos tenían que resolver era el siguiente: ¿qué hacer si llega una orden de lanzamiento pero por algún motivo la caja fuerte que contiene los códigos no se puede abrir? La solución adoptada fue de lo más rusa: usar un mazo para reventar la caja. En 1980, un grupo de oficiales en visita de inspección se mostraron muy críticos con el método, pero sorprendentemente el Estado Mayor soviético respaldó tan curiosa táctica[110]. Queda la duda de hasta qué punto la "técnica del mazo" hubiera podido proteger a la URSS frente a un general revanchista o aburrido que un día decidiese acabar de raíz con el capitalismo. Afortunadamente seguimos vivos, así que en cierto modo la estrategia tuvo éxito.

7) PUERTAS TRASERAS

Uno de los usos más espectaculares, y potencialmente peligrosos, de las contraseñas es la protección de puertas traseras (*backdoors*). Una puerta trasera permite el acceso subrepticio a un sistema protegido, algo así como la entrada por el conducto de ventilación que el héroe de película invariablemente utiliza cuando la entrada principal está demasiado protegida. Esta entrada especial es insertada en el sistema por los propios diseñadores para revisar o cambiar software.

El protagonista de *Juegos de Guerra* utilizó una de esas puertas traseras. Para ello, por supuesto, necesitó la contraseña adecuada. Su táctica consistió en averiguar todo lo posible sobre el diseñador y probar cualquier palabra relacionada con él: nombres, direcciones, temas de interés. Al final sus esfuerzos fueron recompensados: la contraseña utilizada era la del hijo del programador. Ahora tenía vía libre a su objetivo, el catálogo de un fabricante de videojuegos… tras el que se escondía un poderoso ordenador de defensa.

Su ordenador, querido lector, posiblemente oculte una puerta trasera sin que usted lo sepa. Y no, no le estoy introduciendo a una oscura conspiración de los Hombres de Negro. Hará veinte años que conozco este pequeño truco, que ahora compartiré con usted. En los ordenadores hay un sistema llamado BIOS, que funciona como "arranque" del sistema: localiza los elementos hardware y los prepara para enlazarlos con el sistema operativo (Windows, Linux, Mac). La BIOS, entre otras cosas, establece el orden de prioridad en el arranque del ordenador (permitiendo escoger, por ejemplo, si el sistema operativo se buscará en el disco duro, una unidad DVD o un USB), y como opción adicional, permite escoger una contraseña de acceso. Sin esa contraseña, nadie podrá acceder al sistema operativo.

En principio, la BIOS no tiene asignada ninguna contraseña por defecto, y puesto que usted probablemente no sepa siquiera que exista, no habrá asignado ninguna. Pero incluso si lo hace, no piense que está protegido, porque la BIOS incorpora una puerta trasera que se abre con una contraseña determinada, distinta para cada fabricante. Durante años yo he usado la contraseña **589589**, que servía en gran cantidad de ordenadores. Puede usted entretenerse con la lista de contraseñas disponibles, por ejemplo, en[111], [112] y [113].

¿Por qué han hecho tal cosa? No tengo ni idea, pero el hecho es que los diversos fabricantes tienen puertas traseras a sus respectivas BIOS; algunos utilizan más de cuarenta contraseñas. No parece que sirva para otorgar acceso a los fabricantes; e incluso en ese caso, la prudencia impone que ese acceso se cierre antes de comercializar masivamente el producto.

La explicación más habitual que he visto es la de poder recuperar el control del propio ordenador en caso de olvido o pérdida de la contraseña, pero la puerta trasera

permite saltarse la protección de contraseña en cualquier ordenador, propio o ajeno. Es como si el portero de una comunidad colgara en la puerta de entrada un cartel que dijera: “señores propietarios, si no recuerdan dónde han dejado su llave de entrada, aquí tienen la copia”. Como sistema de protección, es sencillamente horrible. Por otro lado, su papel como protección frente a pérdidas de contraseña solamente sería útil si se proporcionase esa información a los clientes, y les aseguro que en ninguno de los ordenadores que yo he manejado en mi vida se acompaña la más mínima indicación sobre la existencia o propiedades de la puerta trasera. Lo único que se me ocurre es que los destinatarios finales de este sistema fuesen los servicios técnicos oficiales, para que pudiesen cobrarle al usuario lo que él mismo podría hacer gratis en su casa.

Con respecto a usted, lector, puede usar la protección BIOS si desea algo de seguridad contra un atacante casual que pase por allí (¡y que no haya leído este libro!), pero de ningún modo confíe en que protegerá el contenido de su ordenador de forma eficaz.

8) LA RESPUESTA: CÓMO PROTEGERNOS

Generalmente no sabemos cómo nuestro servicio web favorito guarda las contraseñas. ¿Las almacena en texto llano? ¿Las esconde con funciones hash? ¿Usa sal? Solamente podemos confiar en que estén haciendo bien las cosas, pero como espero haberles mostrado en este capítulo, la mejor defensa es inútil si el usuario utiliza una contraseña débil.

Hay multitud de consejos sobre cómo escoger una buena contraseña. Permítame aquí incluirle algunos de ellos. Comencemos con los PIN, esos números de cuatro dígitos que utilizamos para proteger el móvil o acceder al cajero automático. En estos casos, lo habitual es que el banco o la operadora telefónica nos proporcione uno, bien en el mismo establecimiento, bien mediante un envío por correo. En principio, el PIN será aleatorio, de modo que puede usted quedárselo. Si no se fía y quiere cambiarlo, hágalo, pero siguiendo la siguiente regla: nunca utilice un número que signifique algo para usted. Ni secuencias numéricas fáciles, ni años, ni fechas de nacimiento, ni las cuatro primeras cifras del número pi, o las cuatro siguientes.

Recuerde que, si usted se cree listo, los ladrones también. Lo digo por si al final le da por escoger 1234 en la creencia de que, puesto que es tan evidente y fácil de adivinar, nadie pensará que lo está usted usando. No se complique usted la vida. La seguridad aquí se basa en pasar desapercibido. Escoja un número al azar y listo. Ah, y nada de escribirlo o anotarlo en ningún lugar, por ingenioso que se crea usted.

En cuanto a las contraseñas de uso en servicios web, correo electrónico, etc, el problema estriba en escoger una que sea al mismo tiempo fácil de recordar y difícil de adivinar. Lo primero que debe hacer es huir del reciclado: jamás reutilice una contraseña (o PIN, ya puestos). Cada servicio web, cada acceso a webmail, cada tienda online debe tener su propia contraseña. Soy consciente de que es difícil recordar un conjunto de contraseñas complicadas, pero hay formas de guardar esa información de forma segura. Puede usted, por ejemplo, escribir las contraseñas en un archivo y protegerlo mediante un programa de cifrado (PGP, por ejemplo). Hay en el mercado, y también en el dominio libre, diversos programas para guardar contraseñas, como por ejemplo Password Safe[114]. Y los navegadores de Internet tienen opciones para guardar los datos de *login* (usuario y contraseña) para diversos servicios web.

Siguiente consejo: el tamaño sí importa. La longitud mínima ha de ser de ocho caracteres, y si puede usted memorizar una más larga, mejor. Si la web en la que quiere registrarse no le permite introducir contraseñas de esa longitud o mayores, escuche mi consejo y líbrese de ellos. No vale la pena.

Mucho ojo con la calidad de una contraseña. Con el avance en las técnicas modernas para romper contraseñas en grandes cantidades, nuestra mejor defensa pasa

por escoger una contraseña robusta. Y ha de ser escogida de forma que no pueda ser adivinada de ninguna forma. He aquí una lista (no exhaustiva) de todo lo que una contraseña no debe NUNCA contener:

—Una palabra reconocible en ningún idioma (ej: "*Libro*")

—Palabras repetidas o encadenadas ("*Librolibro*," "*Holacaracola*")

—Palabras al revés o con las vocales eliminadas ("*Orbil*," "*Lbr*")

—Sustituciones como "1" por la letra l o "0" por la letra o ("*L1br0*")

—Cualquier información con formato reconocible: fechas, nombres, motes, animales, ciudades, números de teléfono, matrículas de coche ("*AnaLaExploradora*")

—Complicaciones sencillas y fácilmente atacables ("*Password1*" "*abcd1234*")

—Caracteres de un solo tipo, sólo letras o sólo números ("*hibsoq*," "*228193*")

Lo ideal, visto lo visto, es una sucesión de caracteres de varios tipos (números, letras, símbolos) que no signifiquen nada en apariencia. En lo posible, mezcle estos tipos de caracteres: *rque58osw* es mejor que *rqueosw58*.

Para crear una contraseña fácil de recordar, un procedimiento habitual que recomiendan muchos expertos es basarse en una frase. Tome usted una frase, la que más le gusta, y escoja la primera letra de cada palabra; o la segunda, o la última. El Quijote nos servirá como ejemplo. Abrimos y leemos "*en un lugar de la mancha, de cuyo nombre no quiero acordarme*".

Primer paso: extraer datos a partir de la frase. Vayamos a lo sencillo, y escojamos las primeras letras del comienzo. Tendríamos así *euldlm*. Segundo paso: aderezar al gusto. Introduzca números, signos, convierta minúsculas en mayúsculas lo que prefiera; pero no lo haga solamente al principio, o al final. Procure repartir los signos en la contraseña, evitando combinaciones como *Euldlm01* o *$%euldlm*. Algo como *EUl6=dlm* o *@eludl$M* serán buenas elecciones. Si quiere usted utilizar contraseñas más largas, mejor que mejor: *e77ul&$dlM*, *euLd1&13Lm*… el límite es la imaginación.

Por supuesto, yo no le recomendaría utilizar la primera frase de uno de los libros más famosos en la historia de la literatura mundial, pero cualquier frase fácil de recordar vale. Quizá esté acostumbrado a oír en casa algo como "¿Te has acordado de bajar la basura?" Genial, ya tenemos *thadblb*, o *esoeraa* (usando las últimas letras de cada palabra). Complíquela un poquito con letras, mayúsculas y algún símbolo, y ya está. Puede incluso utilizar la frase completa como contraseña, aunque no lo recomiendo.

Por último, es buena idea cambiar de contraseña con cierta periodicidad. Cuanto más tiempo la use, más aumentan las probabilidades de perderla por descuido, desidia o fallo informático. Si, a pesar de mis recomendaciones, la apunta en cualquier sitio y tiene la menor sospecha de que alguien pueda haber accedido a ella, cámbiela de inmediato. Incluso el presidente del planeta Spaceballs tuvo el buen juicio de

reaccionar cuando se enteró que la combinación de sus maletas era la misma que la del escudo de aire del planeta Druidia: *"Preparen el Spaceballs 1 para despegue inmediato, ¡y que cambien la combinación de mis maletas!"* Y eso que él no ha leído este libro. O puede que sí.

LOS CÓDIGOS DE DAN BROWN

El nombre de Dan Brown no es precisamente extraño para los amantes de la lectura. Nacido en 1964, este hijo de un matemático y una compositora de música sacra ha combinado historias sobre conspiraciones, secretos, religión y ciencia en algunos de los libros más vendidos de la historia moderna. Al principio, nada hacía presagiar el éxito. Su primera novela, *La Fortaleza Digital* (1998), pasó sin pena ni gloria. No tuvieron más éxito las dos siguientes, *Ángeles y Demonios* (2000) y *La Conspiración* (2001). Sin embargo, *El Código da Vinci* (2003) fue un auténtico best seller. Su polémico argumento lo hizo muy famoso y convirtió al autor en un personaje de éxito. En 2006, una adaptación cinematográfica protagonizada por Tom Hanks se convirtió en una de las películas más taquilleras del año.

La fama alcanzada por el autor de El Código da Vinci alcanzó a su producción anterior. Los libros que antes vegetaban en los estantes de las librerías eran ahora éxitos literarios que se vendían como rosquillas. Y eso nos interesa aquí, porque los temas tratados por Brown incluyen los códigos secretos, los símbolos y la criptografía. De hecho, su primera novela trata directamente sobre criptografía, con una pugna entre hackers y la Agencia de Seguridad Nacional (NSA). Ni hecho a nuestra medida.

Antes de comenzar, me sinceraré con ustedes. He leído los libros de Dan Brown, y creo que Brown ha escrito libros de narrativa fascinante que te dejan pegado a las páginas, pero también me parece un escritor chapucero. Comete errores y fallos garrafales, mete la pata por doquier, y en mi opinión se ha adocenado demasiado. Ni siquiera me molestaré en incluir aquí una reseña de su último éxito, *El Símbolo Perdido* (2009), pues me pareció demasiado predecible y falto de interés.

Quede hecha esta advertencia para que el lector de este libro no se lleve a engaño. Por supuesto, es mi opinión personal, y usted tendrá la suya propia. Pero no vamos aquí a criticar el estilo de Brown sino a aprovechar la temática criptográfica de su bibliografía. Nos centraremos en dos de sus libros, *La Fortaleza Digital* y *El Código da Vinci*. Pero antes de eso, vamos a irnos de tribunales. Nos vemos en el juicio.

1) EL CÓDIGO DEL CÓDIGO

En 2004 Dan Brown fue acusado de plagio por los autores Michael Baigent y Richard Leigh, quienes afirmaban que su libro *El Enigma Sagrado* había servido de inspiración para *El Código da Vinci*. Baigent y Leigh llevaron ante los tribunales británicos a la editorial Random House, editora de los libros de Brown en el Reino Unido (el tercer autor del libro, Henry Lincoln, no tomó parte en el proceso). El 4 de abril de 2006, el juez Peter Smith, del Tribunal Superior de Justicia de Inglaterra y Gales, determinó que no había violación de copyright ni por tanto plagio.

La victoria de Brown y su editorial no hubiera tenido la menor importancia para nosotros de no ser porque el documento del fallo judicial contenía algunos caracteres extraños. Se trata de un documento de 71 páginas donde los pasajes relevantes se marcaron con fuentes en negrilla, cursiva e incluso hipervínculos; pero de cuando en cuando, y sin venir a cuento, aparecía una letra en negrilla y cursiva. Por ejemplo, en la página cinco, la palabra "demandantes" aparece como "claimant***s***". Un párrafo más abajo, vuelve a aparecer, pero esta vez como "clai***m***ant". En el siguiente párrafo podemos leer "***i***s ***t***hat… ***h***is… reali***y***…" Las letras extrañas se pueden unir para formar ***smithy***, diminutivo de Smith[1]. Cuatro letras más nos dan la palabra ***code***. El código de Smithy.

Parece que el juez estaba poseído por el espíritu del código da Vinci y se animó a incluir su nombre esteganográficamente en su propia sentencia judicial. Y no sólo su nombre, ya que si seguimos leyendo la sentencia resulta que hay más mensaje oculto. En efecto, tras "smithy code" aparece un texto cifrado en letras cursivas. El mensaje completo es:

smithycodeJaeiextostgpsacgreamqwfkadpmqzv

El abogado Dan Tench fue el primero en romper el código[2]. El propio juez Smith le animó a ello, e incluso le proporcionó una pista: la clave del código estaba basada en la secuencia de Fibonacci. Como sabrán los lectores de Brown (y muchos otros que no lo sean), se trata de una secuencia en la que cada número se construye como la suma de los dos anteriores: 1, 1, 2, 3, 5, 8, 13, 21… y es parte argumental de *El Código da Vinci*. La secuencia de Fibonacci proporciona la clave o contraseña. Veamos ahora qué sistema de cifrado utiliza.

Probablemente el lector conozca la llamada *Cifra de César*, en la que cada letra se sustituía por la que tenga tres posiciones a su derecha. La *a* se convierte en la *D*, la *b* pasa a ser la *E*, y así sucesivamente. Esta cifra tan sencilla es inadecuada para proteger mensajes, de forma que en el siglo XVI el abad Trithemius propuso una modificación: usar varias cifras de César en orden consecutivo. Digamos que queremos cifrar la palabra *hola*. La primera letra se cifrará con la cifra de César que

transforma cada letra en sí misma. El resultado es *H*. La segunda letra (*o*) se convertirá en *P* tras aplicar la cifra de César que convierte una letra en la siguiente. La tercera letra (*l*) se convierte en la que tiene dos lugares más allá en el alfabeto, es decir, la *N*. Finalmente, la cuarta letra (*a*) se convierte en *D*. *Hola* queda cifrado como *HPND*.

Por desgracia, este sistema es demasiado rígido. Siempre se cifran las letras siguiendo el mismo orden. Por ello, años más tarde el diplomático y criptólogo francés Blaise de Vigenère propuso un conjunto de modificaciones al esquema de Trithemius. Curiosamente, lo que hoy conocemos como cifra de Vigenère es uno de los sistemas *menos* seguros propuestos por el francés. Ironías de la historia.

La cifra de Vigenère tal y como la conocemos pasa por la idea de que podemos escoger qué cifras de César vamos a utilizar. Digamos que queremos utilizar las cifras de César que cambian una letra por la que se encuentra 2, 4, 5 y 1 lugares a su derecha. El texto llano *hola mundo* se transformaría de la siguiente forma:

```
H se convierte en H+2 = J
O se convierte en O+4 = S
L se convierte en L+5 = P
A se convierte en A+1 = B
M se convierte en M+2 = Ñ
U se convierte en U+4 = Y
N se convierte en N+5 = R
D se convierte en D+1 = E
O se convierte en O+2 = Q
```

De esa forma, *hola mundo* se cifra como *JSQB ÑYREQ*. La gran ventaja de la cifra de Vigenère sobre la de Trithemius es su gran flexibilidad: podemos escoger las cifras de César que vamos a utilizar para cifrar, y hacerlo en el orden que queramos. Lo único que tenemos que hacer es enviar al remitente la “clave” que indique nuestra elección. En nuestro caso, podemos hacerlo mediante los números 2,4,5,1. Por convenio se suele asignar el valor uno a la cifra de César trivial, ya que en las tablas que se construían para cifrar mediante este sistema la cifra de César trivial se escribía la primera de todas, y se le asignaba el número 1 o la letra A. De esa forma, la clave vendría dada por 3-5-6-2, o por CEFB.

La llamada “variante Beaufort” de la cifra Vigenère es esencialmente idéntica, pero las cifras de César se usan justo al revés. Es decir, la C se cifra para obtener una A, y por tanto la A se descifra para dar una C. El proceso de cifrado de la tabla Vigenère es igual al proceso de descifrado de la Beaufort.

Volvamos al código del juez Smith. Según nos dice, la palabra de contraseña viene dada por los ocho primeros números de la secuencia de Fibonacci: 1, 1, 2, 3, 5, 8, 13, 21. Esto es, una cifra de Vigenère variante Beaufort con clave AABCEHMU.

Puesto que la variante Beaufort es inversa a la Vigenère normal, tenemos que descifrarla como si estuviésemos cifrando una Vigenére, es decir, sustituyendo letras por las que se encuentran a su derecha.

Las primeras dos letras corresponden a la cifra de César que comienza con una A, que es lo mismo que dejar las cosas como están: *JA* se queda como *ja*. A continuación, hemos de someter la letra cifrada *E* a una cifra de César que comience con B, esto es, que convirtiera cada letra en la siguiente.

Según eso, la *E* se convierte en la *f*. Para la cuarta letra, la clave es C, así que hay que sustituir la cuarta letra cifrada *I* por la que hay dos posiciones a su derecha, lo que nos da una *k*. Obtenemos así el siguiente texto llano:

```
jafkiefisthrwhoareboudreadqough
```

El mensaje sugiere algunas palabras en inglés (*fist* = puño, *who are* = quién eres, *dread* = temido), pero el resto es confuso. El problema aquí es que el juez Smith hizo trampa. En lugar de descifrar las letras 3ª, 11ª, 19ª y 27ª moviendo la letra una posición a la derecha, hay que moverlas dos posiciones a la izquierda: la *E* cifrada no se convierte en *f* sino en *c*. Es como si la secuencia de Fibonacci no fuese 1, 1, 2, 3, 5, 8, 13, 21 sino 1, 1, 25, 3, 5, 8, 13, 21, y la clave correspondiente no sería AABCEHMU sino AAYCEHMU.

¿Por qué hizo esto el juez Smith? Sólo podemos conjeturar, pero mi hipótesis es que ese día se sintió muy creativo y decidió que la primera transformación no trivial sería diferente: en lugar de descifrar yendo una letra a la derecha, hay que irse una letra a la izquierda. Así, *E* no se convertiría en *f* sino en *e*. ¿Y por qué entonces cambió el paso, de una letra a dos? Imagino que se equivocó al contar. O bien, sencillamente, se confundió.

Cualquiera que sea el caso, es fácil corregir el mensaje. Hay dos errores más: la décima letra del texto cifrado debería ser *h* en lugar de *t*; y el texto cifrado debería acabar en una *t*. No es evidente en este punto, pero se verá claramente enseguida. Lo último que nos queda por hacer es introducir un espacio entre cada palabra para que quede más legible, y así obtenemos el resultado final:

```
Jackie Fisher who are you Dreadnought
```

que podemos traducir como "*Jackie Fisher, quién eres tú, dreadnought*". Esta frase es clara para un inglés. John (*Jackie* en diminutivo) Fisher fue un almirante de la Armada Británica a finales del siglo XIX y comienzos del XX. Reconocido como uno de los más grandes marinos ingleses (algunos lo colocan tan sólo detrás de Nelson), contribuyó a aumentar el poder de la Royal Navy de forma beligerante, como indica uno de los proverbios que se le atribuyen: *never explain, never apologise* (nunca explicar, nunca disculparse).

Fisher alcanzó en 1903 la jefatura de la armada (*First Sea Lord*), desde donde fomentó diversos avances técnicos como el torpedo y el submarino. Uno de tales avances fue un nuevo acorazado de diseño revolucionario, que embarcaba una potente y precisa artillería y estaba propulsado por turbinas de vapor: el *HMS Dreadnought*. Fue botado en 1906, justamente un siglo antes de la decisión judicial que nos ocupa. Creado para contribuir a frenar la creciente amenaza de la Armada alemana, su nombre proviene de que su poderío le permitiría imponerse a cualquier adversario y no "temer a nadie" (*dread nought*).

El nombre *Dreadnought* ha sido usado por diversos buques de la armada inglesa, desde un navío de 41 cañones de 1573 hasta el primer submarino nuclear británico, botado en 1955. Irónicamente, el *HMS Dreadnought* de Fisher, diseñado para frenar el poderío naval alemán, no pudo participar en la batalla decisiva (Jutlandia), y su único hundimiento tuvo lugar a la antigua usanza: se abalanzó contra el submarino alemán U-29 y lo hundió de un topetazo. En 1923 fue finalmente desguazado. El propio término dejó de usarse para describir un tipo de buque, una vez que los navíos de línea "pre-dreadnought" fueron retirados del servicio. Con todo, el término *dreadnought* ha quedado en el folclor inglés para indicar algo o alguien que reina supremo en su campo sin temer a nadie. Una especie de Chuck Norris, si me permiten la comparación.

El propio juez Smith reconoció que no le interesaban los códigos ("*odio los crucigramas, y no hago sudokus porque no tengo paciencia*," afirmó posteriormente), y el "código Smithy" lo muestra en algunos fallos. Pero al parecer, la oportunidad era demasiado buena para dejarla pasar. No se puede reprochar nada al juez, quien por otro lado emitió un fallo judicial impecable y bien elaborado desde el punto de vista puramente legal. Los demandantes, descontentos, recurrieron el dictamen del juez, llegando incluso a afirmar que "*[el juez Smith] fue estimulado por el extenso uso de códigos en El Código da Vinci, e indudablemente por su propio interés en tales asuntos, para incorporar un mensaje codificado en su sentencia judicial*," en un intento de sugerir que quizá el juez se había precipitado en su decisión y debería haberse tomado más tiempo.

En el fallo del tribunal de apelación, uno de los magistrados emitió un voto particular lamentando que Smith no hubiera presentado sus argumentos con mayor claridad y orden, lo que hubiera beneficiado el dictamen. A pesar de ello, ese mismo magistrado reconoció que Smith "*hizo un buen trabajo con una sentencia tan larga y compleja en el corto espacio de tiempo que empleó*". En ningún momento se puso en duda la profesionalidad del juez o la conveniencia del dictamen[3]. El recurso fue desestimado en marzo de 2007. En cuanto a las aficiones criptográficas del juez Smith, no hay constancia de que las haya continuado.

2) EL CÓDIGO DA VINCI

Dan Brown publicó su primera novela en 1998. Pero fue la cuarta, publicada en 2003 con el título de *El Código da Vinci*, la que le proporcionó fama mundial. ¡Y de qué manera! Se vendieron cerca de cien millones de copias, y en 2006 llegó la adaptación cinematográfica, protagonizada por Tom Hanks. Tanto el libro como la película hicieron de Brown un escritor de fama mundial.

Es preciso puntualizar que, a pesar de su título, *El Código da Vinci* tiene poco que ver con claves, cifras o códigos secretos. La palabra código parece tener aquí la acepción de *conjunto de reglas o preceptos sobre cualquier materia*, o más bien la de *códice: libro o manuscrito de cierta antigüedad*[4]. Sin embargo, la publicidad de la película resaltó el carácter secretista y oculto. Anuncios de periódico a página completa incluyeron en grandes letras la leyenda "*a partir de hoy, el código será descifrado y el secreto revelado*". Incluso hoy, la web oficial de la novela nos invita a "*resolver acertijos… romper códigos… y desvelar un secreto perdido de da Vinci"*[5]. Ante llamada tan suculenta, solamente podemos aceptar.

Como de costumbre, no es mi intención fastidiarle a usted la lectura (o el pase en DVD). Tampoco es mi intención hacer una crítica literaria, así que no esperen comentarios sobre el estilo literario de Dan Brown o la capacidad interpretativa de Tom Hanks. Dicho esto, comenzaré con una breve sinopsis del argumento. El protagonista, Robert Langdon (que ya apareció en la anterior novela de Brown, *Ángeles y Demonios*), es un experto en simbología religiosa, de esos que dan veinte vueltas sobre las implicaciones culturales de por qué San Pedro partía el pan con la mano derecha en lugar de con la izquierda. Durante una estancia en París, es requerido por la policía para ayudarles a resolver el asesinato del director del museo del Louvre, muerto en extrañas circunstancias. El fallecido, antes de morir, tuvo tiempo de dejar escrito un críptico mensaje, que obviamente habrá de ser descifrado.

El criptógrafo encargado del caso es la bella Sophie Neveu, del Departamento de Criptografía de la policía francesa, quien además resulta ser la nieta del hombre muerto; éste, a su vez, estaba complicado en cierta sociedad oculta, la cual es a su vez perseguida por el Opus Dei. Para rematar la faena, el policía encargado del caso sospecha de Langdon, y le considera poco menos que el asesino. El resultado es una mezcla de persecuciones, investigaciones, pesquisas y descubrimientos más o menos sorprendentes.

Hay varios elementos criptográficos en la película. El primero, por supuesto, es que la coprotagonista de la peli es criptógrafa. Hasta ahora, y con la excepción de la película *Enigma*, la chica siempre es arqueóloga, actriz, profesora universitaria, científica, piloto de aeroplano, pero no criptógrafa. Esto, para los aficionados a la criptografía, representa una reivindicación: por fin los criptólogos no son extranjeros

con pintas raras y números en la cabeza, sino que aparecen como personas reales que comen en restaurantes, conducen coches, enamoran a gente normal y todo eso.

Hemos de hacer una objeción antes de comenzar: esta criptógrafa no se gana el sueldo. Sólo hay tres ocasiones en las que ha de hacer uso de sus habilidades de descifradora. En la primera, los protagonistas se encuentran con un anagrama, es decir, una frase con cuyas letras puede formarse otra. Lo lógico sería que la bella Sophie Neveu fuese la encargada de resolver el anagrama; pues no, fue Robert Langdon quien se encarga de ello. Afortunadamente, en seguida aparece un segundo anagrama, que es resuelto por Sophie. No voy a revelar el anagrama, pero advierto al lector que la película utiliza los anagramas en inglés original, que son diferentes a los del libro en español. Confío que el lector que haya visto la película (o el espectador que haya leído el libro) no se confunda por ello. Puedo decir, como mínimo, que los anagramas en la versión española son adecuados y están bien hechos.

Vamos con el segundo elemento cripto. Nevey y Langdon deben abrir la caja fuerte de un banco, para lo cual disponen de una llave y una clave numérica (lo que en la versión original da lugar a un juego de palabras, ya que la palabra inglesa *key* significa tanto llave como clave). La clave numérica parece ser el número de cuenta del banco, pero cualquier podría haberla leído. Sophie no las tiene todas consigo, y hace partícipe de sus sospechas a Langdon: "*Este número de cuenta no es correcto… es demasiado aleatorio*".

Puesto que el número de cuenta actúa como la contraseña, el razonamiento de Sophie parece absurdo. El propio Langdon lo razona así: "*los bancos aconsejan siempre a sus clientes que escogieran sus números secretos de manera aleatoria, para que nadie pudiera adivinarlos. Y, evidentemente, aquello no era una excepción*". Pero Sophie, lejos de ser una caprichosa, se da cuenta de que reordenando los números de la cuenta se obtiene la secuencia de Fibonacci, en la que cada número es la suma de los dos anteriores: 1-1-2-3-5-8-13-21. "*Es demasiado casual que los números de esta cuenta, supuestamente aleatorios, puedan reordenarse para formar la Secuencia de Fibonacci*".

Tenemos aquí un agujero de seguridad típico. En efecto, no sólo los bancos sino cualquier experto en seguridad mínimamente competente recomiendan que los números de cualquier clave sean aleatorios, pero los seres humanos somos algo penosos a la hora de recordar secuencias aleatorias. Sophie se debate en su razonamiento: su abuelo, hombre inteligente, escogió una secuencia que parecía aleatoria, pero no lo era. Nos encontramos en un momento crucial. La caja del banco les permite probar una sola secuencia numérica. ¿Tiene razón Sophie? ¿Se está pasando de lista? ¿Hay una tercera opción, y Langdon la tiene en la punta de la lengua? Les dejaré que lo averigüen ustedes mismos en el libro.

Por fin, el momento cripto por excelencia. Tras múltiples aventuras y carreras,

adquieren la posesión de un aparato al que denominan *criptex* (*cryptex* en la versión inglesa). Se supone que es una réplica de un dispositivo inventado por el propio Leonardo da Vinci. Consiste en un cilindro con cinco discos móviles que incluyen las letras del alfabeto. Funciona de forma parecida a los candados de bicicleta: si se giran los discos hasta la posición correcta, se abre. Puesto que el número de "claves" es igual al número de letras del alfabeto elevado a la quinta potencia, los autores deben usar el cerebro para deducir la clave.

¿Por qué no romper el criptex? Pues porque Leonardo, que no tenía un pelo de tonto, lo diseño con un dispositivo de autodestrucción. El mensaje, escrito en papiro, está colocado entre la pared interior del criptex y un núcleo de vidrio lleno de vinagre. Si se rompe el criptex, el vinagre disolverá el papiro y adiós al mensaje, en una anticipación de los mensajes de James Bond que se autodestruyen tras leerlos.

Antes de seguir, hemos de dejar clara una cosa: la existencia del criptex es una invención de Dan Brown. Hasta donde se sabe, Leonardo da Vinci nunca utilizó ningún tipo de sistema de cifrado. Sí es cierto que escribía con la mano izquierda, al revés, de forma que sus escritos sólo pueden leerse poniéndolos ante un espejo, pero no está claro si lo hacía como método criptográfico. Tanpoco consta que haya construido nada parecido a un criptex. En este punto, Dan Brown aprovecha la extremada versatilidad de Leonardo, de quien consta que diseñó planos de aparatos voladores, tanques e infinidad de invenciones. El criptex es, sencillamente, una licencia literaria, válida y bien usada.

En realidad, la ficción influyó en la realidad y el éxito de ventas de *El Código da Vinci* acabó dando vida propia al criptex. Para promocionar la película, la productora Sony Pictures y Google lanzaron en 2006 un concurso llamado *The Da Vinci Code WebQuests*. Los participantes debían resolver una serie de acertijos, y los diez mil primeros participantes recibieron una réplica del criptex[6]. Para los que se quedaron con las ganas de poseer un criptex, buenas noticias: la empresa norteamericana Cryptex Security le venderá uno por poco menos de 190 euros mas gastos de envío[7]. Y, puestos a aprovechar el nombre, la empresa española de software Global Duir vende un programa de cifrado de archivos llamado Duir Criptex[8].

Volviendo al argumento del libro, probablemente usted esté pensando en alguna forma de abrir el criptex sin conocer la clave. Me refiero a sistemas imaginativos como radiografiar el criptex para obtener la disposición correcta de los discos. Hace tiempo leí una idea ingeniosa: congelar el vinagre y arrearle un martillazo al cacharro. En cualquier caso, problema resuelto. En lugar de eso, los protagonistas las pasan canutas hasta que alguien da con la clave (no Sophie, me temo).

Dan Brown aprovecha también para compartir con nosotros sus conocimientos sobre criptografía. La idea es presentar a Sophie Neveu como una inteligente y capaz criptóloga. Para ser sinceros, yo me pregunto qué pinta una criptóloga en la obra

(aparte de hacer bonito como contrapunto del héroe masculino), si al final resulta que todos los códigos los revelan otros, todas las claves se descifran en otras mentes. El problema subyacente es que tampoco el escritor parece tener ni idea de criptografía.

Les daré algunos ejemplos. De todos los criptólogos que ha habido a lo largo de la historia, Sophie solamente cita a dos: Simmermann y Schneier. Uno de ellos no es criptógrafo, y además su apellido está mal escrito. Eso sí, ambos son expertos en el tema. Philip Zimmermann (con Z) es el creador del programa de cifrado PGP, usado por millones de usuarios para proteger mensajes de correo electrónico. La empresa que comercializa PGP fue vendida a Symantec en 2010 por 300 millones de dólares (Zimmernann, para entonces, se había desvinculado de su propio programa). En cuanto a Bruce Schneier, escribió a mediados de los años noventa *Applied Cryptography*, que durante años ha sido considerada la "biblia" de la criptografía, con múltiples ejemplos de algoritmos de cifrado de todo tipo[9]. Incluso tiene su sección de *Bruce Schneier Facts*, a semejanza de los famosos *Chuck Norris Facts*[10].

Estos ejemplos no indican más que un cierto descuido a la hora de escribir. Pero conforme profundizamos en la lectura, los problemas se agudizan. Al igual que hizo en *La Fortaleza Digital*, el autor sigue empeñado en referirse a un inexistente sistema de escritura cifrada llamado "la Caja del César". Avanzamos una línea más, y podemos leer lo siguiente: "*María Estuardo, reina de Escocia, creó un sistema mediante el cual unas letras podían ser reemplazadas por otras, y enviaba mensajes desde la cárcel*"". Esto duele. Porque el caso de la reina María Estuardo está muy bien documentado, incluyendo el uso de una cifra que ella misma solicitó en ocasiones que fuese modificada; pero de ahí a decir que la propia reina creó la clave ella solita va un buen trecho. Ni siquiera cifraba ella las cartas, sino su secretario. Y aunque fue confinada en su heredad de Chartley Hall, deberíamos hablar de arresto domiciliario más que de encarcelamiento propiamente dicho.

Dan Brown cita casi correctamente al científico árabe Abú Yusuf Ismail al-Kindi, de quien afirma que "*protegía sus secretos con códigos cifrados polialfabéticos*". Al-Kindi, en efecto, fue un matemático del siglo IX que escribió casi 300 obras sobre medicina, astronomía, matemáticas, lenguaje y música. En 1987 se descubrió un tratado suyo titulado *Sobre el desciframiento de mensajes criptográficos*[11]. El descubrimiento del análisis de frecuencias por parte de Al-Kindi precede en varios siglos a los "descubridores" italianos, y algunos autores lo consideran el padre del análisis estadístico. Sin embargo, hasta el momento no hay constancia de que Al-Kindi conociera los cifrados de tipo polialfabético, y no hay prueba alguna de que lo utilizase[12]. Es posible que futuras investigaciones arrojen más luz al respecto[13].

Hasta el momento, afirmar con tanta rotundidad que Al-Kindi protegía sus secretos con códigos cifrados polialfabéticos constituye una afirmación arriesgada. Por supuesto, podríamos apuntarlo como otra licencia literaria, pero más bien parece

un fallo más. Demasiados puntos negros para un autor que cinco años antes había escrito *La Fortaleza Digital*, una novela donde la criptografía jugaba un papel fundamental. Por supuesto, tenemos que leerlo y evaluarlo. Vamos allá.

3) LA FORTALEZA DIGITAL

Aunque Dan Brown se hizo mundialmente famoso en 2003 con el éxito de ventas *El Código da Vinci*, su primera novela fue *La Fortaleza Digital*. En ese libro, publicado en 1998, la criptografía y los códigos secretos no son un añadido más para dar emoción, sino que constituyen el núcleo del argumento. Por supuesto, cuando Brown fue catapultado a la fama, sus anteriores obras fueron inmediatamente re-editadas y elevadas a la categoría de bestsellers.

Este que escribe no supo de la existencia de ese libro hasta 2007, y fue necesaria una aburrida estancia en un aeropuerto alemán para animarme a adquirir una copia en inglés. Ya saben ustedes lo que pasa en los aeropuertos, las largas esperas y el hastío hacen que uno devore cualquier plato que le pongan ante las narices. Eso me pasó a mí con *La Fortaleza Digital*. Mi veredicto es el mismo que el de una comida de avión: apetitosa en apariencia, pobre en sabor y claramente insatisfactoria. A pesar de ello (o precisamente debido a ello), creo que puede servirnos para comentar algunos aspectos sobre la criptografía del siglo XX.

El libro en sí me decepcionó bastante, tengo que reconocerlo. Y no me refiero sólo a la parte cripto. El argumento es en ocasiones aburrido y a veces predecible. Los personajes parecen calcados de otros libros. Algunos enigmas se resuelven fácilmente sin más que traducirlos al español (yo leí la versión en inglés, así que no sé cómo se las habrá apañado el traductor para dar sentido sin desvelar el truco). Comentario aparte merece la descripción que hace Brown de Sevilla (donde transcurre parte de la acción), más propia del siglo XVII que del XXI. Es evidente que el autor nunca ha estado en Sevilla, y si en el futuro se atreve a visitarla yo le recomendaría una fuerte escolta… y un pseudónimo.

Muchos críticos han despellejado a Dan Brown desde el punto de vista meramente literario. Incluso dejando de lado sus muchos fallos, me resultó bastante menos interesante que *El Código da Vinci* o *Ángeles y Demonios*. Pero eso es la opinión particular de un lector, y seguro que usted espera algo más. En efecto, vamos a estudiar *La Fortaleza Digital* desde el punto de vista criptográfico.

En este punto, debo advertir al lector de que habrá algunos “spoilers”. Por lo general, cuando critico un libro procuro no revelar el argumento, de forma que el lector pueda decidir si quiere leerlo o no. Pero en este caso, todo el libro gira en torno al tema que estamos aquí tratando, así que la tarea será difícil. En lo que sigue, me centraré en los puntos meramente criptográficos, y no daré detalles que descubran el argumento; así no le fastidiaré la lectura.

El libro gira en torno a la Agencia de Seguridad Nacional (NSA), encargada de proteger los códigos secretos del gobierno norteamericano y de romper los de otras naciones. El protagonista principal es ahora una mujer, Susan Fletcher, mano derecha

del jefe de uno de los principales proyectos de la NSA: el superordenador TRANSLTR, capaz de descifrar cualquier tipo de código secreto. El compañero de la protagonista, un profesor de universidad, también acaba metido en el fregado, junto con un misterioso asesino letal (que, incomprensiblemente, se convierte en un chapuzas total a la hora de acabar con un protagonista, y espero no estar revelando demasiado), y aderezado con diversos actores secundarios entre los que se encuentra el malo de turno.

El tono general del libro es, aproximadamente, que la *Electronic Frontier Foundation* son una banda de hippies tocapelotas, y la pobre NSA es una inocente e incomprendida agencia gubernamental que se dedica a interceptar las comunicaciones de los malos malísimos, y absolutamente de nadie más.

Los protagonista se ven sacudidos cuando descubren la existencia de un código secreto llamado Fortaleza Digital, creado por un ex-empleado de la NSA. Dicho código es indescifrable, y no sólo en el sentido criptoanalítico puro sino también en el de "fuerza bruta": incluso probando todas las claves posibles, las características del código hacen que el ordenador sea incapaz de descifrarlo. El creador del código pretende venderlo al mejor postor. Eso, por supuesto, sería una pesadilla para la NSA: nada menos que un código que no pueden leer. Y hasta aquí, lo que voy a revelar del argumento. Pasemos a los momentos cripto.

CIFRADO AL ESTILO DE CÉSAR

Al comienzo de la novela, Susan Fletcher explica a su novio algunas cosas sobre la criptografía, un antiguo truco de autor para introducir al lector en la temática del libro. En alguna ocasión, su perorata sobre la NSA parece sacada de los libros de James Bamford[14], pero por lo demás supone una buena introducción para el lector.

Apenas Fletcher habla de criptografía propiamente dicha, parece equivocarse. Comienza hablando de un sistema de cifrado llamado "cuadrado perfecto de César". Supuestamente, César escribía un mensaje en forma de cuadrado perfecto, y luego lo reordenaba cambiando filas por columnas. Con ello, está describiendo una cifra de trasposición: el texto se escribe por líneas y se lee por columnas.

El primer problema consiste en que César nunca usó este sistema de cifrado. El cuadro, o caja, de César, es tan sólo una invención de Dan Brown. Probablemente, el autor se basó en dos sistemas de cifrado conocidos desde la antigüedad, y atribuidos a César y Polibio.

La llamada **cifra de César**, fue descrita por vez primera por el historiador romano Suetonio en su *Historia y Vida de los Césares*:

Se han conservado, por otra parte, sus cartas [de César] a Cicerón y las que dirigía a sus familiares sobre asuntos domésticos. Cuando tenía que enviarles alguna información secreta la escribía en clave, esto es, disponía el orden de las letras de tal modo que no se pudiese reconstruir ninguna palabra si se quiere descubrir lo que

dice y descifrarlo, hay que sustituir cada letra por la que le sigue en tercer lugar, esto es, la D sustituye a la A, y así las restantes.

El segundo método, que probablemente sugirió a Brown la idea del cuadrado perfecto, es el llamado **cuadrado de Polibio**. Antes de la transmisión, se acuerda un cuadrado 5x5 en el que se disponen las letras del alfabeto. El emisor enciende dos antorchas como señal de que un mensaje va a ser enviado, a lo que el receptor responde con otras dos. Tras este "indicativo de llamada," se toma un conjunto de entre una y cinco antorchas en la mano izquierda, y otro conjunto de antorchas en la mano derecha. Eso nos da la fila y la columna en el cuadrado, lo que nos proporciona la codificación de la letra. Como ejemplo, supongamos el siguiente cuadrado:

	1	2	3	4	5
1	a	b	c	d	e
2	f	g	h	i	j
3	k	l	m	n	o
4	p	q	r	s	t
5	u	v	x	y	z

Según el esquema de Polibio, dos antorchas encendidas seguidas de otras cuatro nos daría el número 24, o fila 2, columna 4, lo que nos da la letra i. Para transmitir POLIBIO tendríamos que enviar los números 41-35-32-24-12-24-35.

La idea subyacente al cuadrado de Polibio constituye un hito por diversos motivos. Es uno de los primeros intentos por aumentar "la capacidad del canal", ya que hasta entonces las señales a distancias tan sólo podían señalar eventos binarios (hay fuego, o no lo hay), o como mucho una cantidad limitada de mensajes. En segundo lugar, su cuadrado permite una conversión cómoda entre letras y números, lo que constituye el punto de partida de muchos sistemas criptográficos. Su esquema de codificar un paquete de información en forma de dos variables tiene múltiples aplicaciones, ya sean coordenadas geográficas (longitud y latitud) o juegos tipo guerra naval ("H-2, tocado").

COMPUTACIÓN CUÁNTICA

Hasta ahora, Dan Brown no se ha lucido precisamente para crear el clima apropiado, pero en el fondo todo es una excusa para justificar la necesidad que tenía la NSA de construir TRANSLTR, el más potente y secreto ordenador de ruptura de códigos. De la descripción del autor podemos concluir que su eficacia descansa en tres principios: la implementación de paralelización masiva (tres millones de procesadores); "*avances en valoración de texto llano altamente confidenciales para adivinar las contraseñas y romper los códigos*"; y finalmente, nuevos descubrimientos en el campo de la computación cuántica.

Es posible que el lector haya oído hablar del concepto de **computación cuántica**. Se basa en el uso de sistemas cuánticos para poder resolver problemas computacionalmente muy complejos en poco tiempo. Los ordenadores clásicos operan bajo una lógica binaria, con bits que solamente pueden adoptar los valores uno o cero, pero los ordenadores cuánticos podrían utilizar "qbits" con un número de estados mucho mayor. Un ordenador cuántico podría, por ejemplo, utilizar el llamado algoritmo de Schor para descomponer un número N en factores primos, con una eficiencia y una velocidad impensable para un ordenador convencional. En tal caso, toda la criptografía de clave pública podría venirse abajo.

La computación cuántica es todavía un concepto, con muy pocas aplicaciones prácticas hasta ahora. Para darles una idea, les diré que en 2001 un grupo de investigación de IBM causó sensación al anunciar la creación de un ordenador cuántico de 7 qbits, con el cual pudieron factorizar el número 15[15]. Parece un resultado ridículo, pero lo importante fue demostrar que el concepto era factible. En mayo de 2011 la empresa canadiense D-Wave Systems afirmó haber vendido a la Lockheed Martin Corporation un ejemplar de su *D-Wave One*, el primer ordenador del mundo basado en lo que llaman "computación cuántica adiabática". El comprador fue la Lockheed Martin Corporation[16], y aunque la venta se llevó a cabo a finales de 2010, la noticia se hizo coincidir con la publicación de un artículo sobre computación cuántica en la prestigiosa revista *Nature*[17].

Una aplicación posterior, en agosto de 2012, utilizó un total de 81 qbits para resolver un problema de plegado de proteínas, pero con problemas: en 10 000 pruebas, el ordenador solamente dio con la respuesta correcta en 13 ocasiones[18]. Sin embargo, hay quien piensa que el ordenador cuántico puede ser la dirección que la informática tome en el futuro; es el caso del fondo de inversión In-Q-Tel, que en septiembre de 2012 anunció su apoyo financiero a la empresa D-Wave[19]. Por cierto, In-Q-Tel fue fundado para invertir en empresas de alta tecnología cuyos productos puedan ser de interés… para la Agencia Central de Inteligencia (CIA)[20].

Que un ordenador cuántico sea factible y práctico, por no hablar de que pueda romper códigos de forma más rápida y eficiente, queda aún por ver, pero los cimientos han sido puestos. Como muestra, baste recordar que el premio Nobel de Física de 2012 fue concedido a los investigadores Serge Haroche (Francia) y David J. Wineland (EEUU) por "*sus revolucionarios métodos experimentales para permitir la medida y manipulación de sistemas cuánticos individuales*"[21].

De momento, un ordenador cuántico es sólo una fantasía; en palabras de Wineland, "*pasará mucho tiempo antes de que podamos crear un ordenador así, pero creo que la mayoría de nosotros… creemos que acabará sucediendo; es cuestión sobre todo de controlar esos sistemas cada vez mejor*"[22]. Por supuesto, si una entidad como la NSA hubiera construido uno, le proporcionaría una ventaja enorme.

¿Cómo de grande?

BITS, BYTES Y CLAVES

Volvamos a *Fortaleza Digital* y veamos qué puede hacer Dan Brown (perdón, Susan Fletcher) con un ordenador cuántico. En su primera prueba, nos dice, localizó una clave de 64 caracteres en diez minutos. Muy impresionante… salvo que confundió caracteres con bits. Un pequeño pero importante detalle. Normalmente, las claves criptográficas son una sucesión de ceros y unos. Si deseamos usar letras, podríamos hacer una correspondencia con números binarios. El código ASCII, por ejemplo, realiza una transformación de ese tipo para representar texto en ordenadores y en comunicaciones.

Probar 2^64 claves en diez minutos es una hazaña respetable incluso con los mayores ordenadores de nuestros días. Sin embargo, hay muchos algoritmos de cifrado simétrico de 128 bits, y contra ellos incluso TRANSLTR sería inútil, ya que el tiempo necesario para probar las 2^128 claves de un sistema así le llevaría del orden de billones de años. Con lo que hemos avanzado poco.

Dan Brown da la impresión de creer que una clave de 128 bits es solamente el doble de resistente que una de 64 bits: Susan Fletcher comenta en una ocasión: "*tuvimos un mensaje interceptado hace unos meses, que nos llevó una hora, pero tenía una clave ridículamente grande, diez mil bits o así*". El problema consiste en que estamos hablando de aumentos exponenciales, no lineales. Seguro que usted, lector, tiene móvil. El PIN tiene cuatro dígitos, de forma que hay 10 000 posibles combinaciones; cinco dígitos elevaría el número de posibles valores del PIN hasta 100 000, diez veces más. En criptografía suelen usarse dígitos binarios, así que una clave de 65 bits es el doble de resistente que una de 64 bits.

Quizá a estas alturas piense usted que la potencia del ordenador TRANSLTR no depende tanto de su velocidad como de su capacidad criptoanalítica. Nadie en su sano juicio intenta probar todas las claves posibles salvo como último y desesperado recurso. En la práctica, es difícil crear un algoritmo de cifrado sin vulnerabilidades, de forma que un atacante puede intentar obtener la clave con menos cálculos de los que le exigiría un ataque de fuerza bruta. Por poner un ejemplo, supongamos que yo les digo que, debido a problemas con los algoritmos usados, los valores del PIN que desbloquean un móvil no pueden ser múltiplos de tres. Sabiendo eso, tan sólo necesito probar un 67% de los 10 000 posibles valores. Muchos algoritmos de cifrado tuvieron que ser descartados en el pasado porque no funcionaron con la efectividad teórica que prometían.

Podríamos pensar que es a eso a lo que Dan Brown se refería con eso de los "*avances en valoración de texto llano altamente confidenciales para adivinar las contraseñas y romper los códigos*". Evidentemente sería mucho más inteligente

combinar la potencia de cálculo del ordenador con procedimientos de criptoanálisis lineal, diferencial, estudios de entropía y todo tipo de trucos para acelerar los cálculos.

¡Pues no! El autor deja bien claro que TRANSLTR es un mero calculador tonto que prueba claves ciegamente sin aprovechar atajos. Fuerza bruta, al mejor estilo americano. Las sutilezas aquí no valen: "*para TRANSLTR, todos los códigos parecen idénticos, sea cual sea el algoritmo que les creó. No lo entiendo, —dijo— no estamos hablando de ingeniería inversa para alguna función compleja, hablamos de fuerza bruta*". Esto, de ser cierto, representaría un completo sinsentido. En primer lugar, hay que saber qué tipo de algoritmo de cifrado se está estudiando, aunque sea para saber cómo reproducirlo a la hora de probar todas las claves; y en segundo, las técnicas de criptoanálisis moderno permitirían obtener una solución de manera mucho más fácil y rápida. Es como si China dijese que, puesto que tiene muchos soldados, ya no necesita radares, tanques o armas nucleares. ¿Le convence a usted ese razonamiento?

EL CÓDIGO INDESCIFRABLE

En un momento dado, Susan Fletcher afirma rotunda: "*un código indescifrable es una imposibilidad matemática*". Lo siento, doctora, pero no es cierto. Hay un sistema denominado Libreta de Uso Único (OTP) que es completamente indescifrable, y además puede demostrarse matemáticamente. Patentado en 1919 por Gilber Vernam (de la AT&T) y Joseph Mauborgne (capitán del ejército de EEUU), una OTP es sencillamente una lista de caracteres aleatorios. Para cifrar un mensaje *M*, sacamos una ristra de caracteres *K* de la OTP, de tal forma que tanto *M* como *K* tengan la misma longitud. El cifrado consiste, sencillamente, en sumar las dos cadenas para obtener el mensaje cifrado *C=M+K*. Para descifrar, basta con restar la clave al texto cifrado para recuperar el mensaje: *M=C-K*.

Para aplicaciones prácticas resulta más cómodo utilizar una operación matemática ligeramente diferente a la suma, que se suele denominar XOR (*eXclusive OR*). La operación XOR, que denotaremos mediante el símbolo ⊕, produce este resultado cuando se aplica sobre dos bits:

$$0 \oplus 0 = 1 \oplus 1 = 0$$
$$0 \oplus 1 = 1 \oplus 0 = 1$$

La ventaja de la operación XOR sobre la suma es que es su propia operación inversa. Es decir, si X ⊕ Y = Z, también se cumple que Z ⊕ Y = X. Asimismo, se cumple que el cero es el elemento neutro de esta operación: X ⊕ 0 = X. Finalmente, cualquier operación xor de un elemento cualquiera consigo mismo nos da el elemento neutro: X ⊕ X = 0. Todo esto se cumple también si X es una cadena larga de bits.

Estas propiedades hacen que resulte más atractiva la operación xor que la suma

tradicional. El mensaje cifrado se obtiene sencillamente como:

$$M \oplus K = C$$

Cuando queramos recuperar el mensaje *M*, no tenemos más que “xorear” (y espero me disculpen el palabro) el texto cifrado con la clave:

$$\begin{aligned} C \oplus K &= (M \oplus K) \oplus K \\ &= M \oplus (K \oplus K) \\ &= M \oplus 0 \\ &= M \end{aligned}$$

De este modo, el atacante que quiera obtener el mensaje *M* deberá conocer la clave *K*. Ni siquiera podemos saber cuándo hemos acertado. Un mensaje cifrado de seis caracteres, digamos KEHQGZ, puede descifrarse como ATACAD con una clave, como MERINA con otra clave, y como ARTURO con otra. ¿Cómo saber cuál es el descifrado correcto? Sencillamente, no lo podemos saber. Ese detalle, por sí sólo, echaría por tierra la idea de “lo probamos todo” de TRANSLTR, ya que obtendría tantos mensajes descifrados potencialmente correctos que no se podría saber cuál es el correcto. Fletcher, inexplicablemente, no cae en la cuenta.

Apuesto a que en estos momentos el lector (usted) se estará preguntando: si la libreta de uso único es el sistema criptográfico perfecto, ¿por qué no lo usamos siempre? Bien, hay dos inconvenientes prácticos. El primero es que la clave ha de ser tan larga como el mensaje, y si el volumen de las comunicaciones es muy grande, genera grandes problemas de creación, distribución y uso de clave. Reutilizar la libreta no es una opción, porque dos mensajes cifrados con la misma clave de una OTP serían muy fácil de descifrar incluso sin conocer la clave. Supongamos dos mensajes X1, X2 cifrados con la misma clave Y: $X1 \oplus Y = Z1$ y $X2 \oplus Y = Z2$. Al “xorear” los dos mensajes cifrados, obtenemos:

$$\begin{aligned} Z1 \oplus Z2 &= (X1 \oplus Y) \oplus (X2 \oplus Y) \\ &= (X1 \oplus X2) \oplus (Y \oplus Y) \end{aligned}$$

Ahora bien, como hemos visto la operación XOR aplicada a dos bit iguales nos da cero. De ese modo, resulta que $Z1 \oplus Z2 = X1 \oplus X2$. Es decir, obtenemos una especie de “suma” de los dos mensajes en texto llano. Y, puesto que tanto X1 como X2 son mensajes redactados en una forma conocida (texto en español, por ejemplo), no es difícil obtener X1 y X2. La seguridad que proporcionaba la OTP ha desaparecido por completo.

El segundo problema de la libreta de uso único es que ha de contener solamente caracteres verdaderamente aleatorios. Antiguamente, se utilizaban mecanógrafos que

supuestamente tecleaban números al azar, pero no siempre el resultado podía ser considerado realmente aleatorio. En la actualidad, el problema continúa. Suena extraño, pero el hecho es que los ordenadores digitales, diseñados para ejecutar pasos electrónicos bien determinados, se comportan realmente mal frente al azar. Producir números aleatorios mediante programas informáticos es una tarea muy difícil.

Por ambos motivos, el uso de libretas de uso único es poco frecuente, y se limita a situaciones muy específicas. En 1963, tras la crisis de los misiles de Cuba, se estableció una línea segura entre el Kremlin y la Casa Blanca, una teleimpresora que utilizaba una libreta de uso único acordada por ambas partes. El famoso "teléfono rojo" fue instalado más tarde, durante los años setenta.

También fue una herramienta vital para los espías en territorio enemigo. No siempre surtieron el efecto deseado. Durante la Segunda Guerra Mundial, una red soviética de espías en Estados Unidos utilizó unas libretas de uso único que ya habían sido utilizadas anteriormente, lo que permitió a los criptoanalistas norteamericanos y británicos descifrar algunos mensajes. Esto condujo a la desarticulación de la red de espionaje, en un largo proceso de contraespionaje llamado operación VENONA que duró casi cuarenta años y cuyos detalles no fueron hechos públicos hasta 1995[23].

VIDA Y MILAGROS DE SKIPJACK

En otro momento del libro, el autor habla de un "algoritmo de clave pública" llamado Skipjack y de cómo el gobierno estuvo a punto de imponerlo como estándar de cifrado. La tesis del libro es que Skipjack tenía una "puerta trasera" que permitiría a la NSA acceder a los contenidos de cualquier mensaje cifrado con este sistema. No les diré más para no revelar parte del argumento, pero puede que le interese a usted conocer la verdadera historia de Skipjack. Porque en efecto existió, y estuvo cerca de convertirse en una pesadilla para la privacidad individual.

En 1990, la NSA terminó de evaluar un algoritmo de cifrado simétrico llamado Skipjack (Dan Brown debió pensar que "algoritmo público" y "algoritmo de clave pública" son lo mismo, pero no, no es un sistema de clave pública). Su clave de 80 bits lo hacía mucho más difícil de romper mediante fuerza bruta que el venerable DES. Se insertaba en un chip resistente frente a manipulaciones externas (*tamper-proof*) llamado Clipper, y llevaba un sistema de intercambio de claves basado en criptografía de clave pública (de ahí la confusión de Dan Brown), así como los sistemas más avanzados y seguros de firma digital. Los chips Clipper fueron diseñados en principio para proteger conversaciones telefónicas, pero también podían ser usados en otros sistemas mediante el uso de una cripto-tarjeta llamada Fortezza.

En principio, Clipper se diseñó como estándar federal, lo que significa que sería de uso obligado tanto para las agencias gubernamentales como para los contratistas que hiciesen negocios con el gobierno. A la larga lo habitual es que se acabe

convirtiendo en un estándar de facto para toda la industria electrónica, y luego para la sociedad en general. Eso ha sucedido en el pasado con algoritmos de cifra como DES o AES.

La NSA hizo público el algoritmo en 1998. Los criptoanalistas civiles tardaron poco en encontrar ataques matemáticos contra Skipjack. No consiguieron "romper" el algoritmo, pero incluso las rupturas parciales dejaron un mal sabor de boca, ya que se suponía que Skipjack era un algoritmo diseñado y probado durante años por la mayor agencia criptológica del mundo.

Hasta donde se sabe, Skipjack nunca ha tenido una "puerta trasera". El problema es que no la necesitaba, porque el gobierno norteamericano intentó algo aún peor. Para entenderlo, hemos de remontarnos hasta mediados de los años noventa. En aquella época, el gobierno de EEUU estaba tan preocupado por el uso libre de sistemas de cifrado que llegó a plantearse su prohibición total. En su defecto, intentó imponer un sistema denominado "depósito de claves" (*key escrow*) mediante el que cualquiera podría usar el sistema de cifra que quisiera… con la condición de dar al gobierno una copia de la clave de cifrado. La filosofía subyacente es que los ciudadanos o las empresas tiene perfecto derecho a esconderse sus secretos entre ellos, pero no a ocultárselos al Estado.

Un amplio e intenso debate circuló durante años acerca de esta iniciativa. Los problemas eran de privacidad y de seguridad. Un sistema donde el gobierno tuviese la clave a nuestros secretos era algo inconcebible en una democracia moderna. Era como si la policía nos exigiese dejar una copia de las llaves de casa en la comisaría más cercana, no fuese a ser que algún día tuvieran que entrar para registrarla. Incluso si eso nos pareciese bien, una base de datos con todas las claves y contraseñas de un país constituiría un objetivo de primera magnitud para cualquier organización criminal. Los problemas para mantener esa base de datos segura serían poco menos que insalvables[24]. Es posible encontrar situaciones donde un depósito de claves sea aceptable, por ejemplo en entornos empresariales, pero un depósito masivo a nivel nacional no lo es.

Ahora entenderán por qué el algoritmo Skipjack no fue muy querido. No es que fuese malo en sí, sino que el chip Clipper en que estaba instalado incluía también una opción de depósito de claves. Cada teléfono o dispositivo equipado con Clipper enviaba, de modo cifrado, una copia de la clave usada, que sería almacenada en depósito. Cuando el gobierno estimase necesario pegar la oreja, no tendría más que usar su copia. Habría salvaguardias legales, por supuesto, pero la infraestructura de espionaje masivo estaría lista y funcionando.

Afortunadamente para todos, el sistema de depósito de claves nunca llegó a imponerse. Poca gente estaba dispuesta a ceder su privacidad de una forma tan descarada. Había múltiples alternativas en el mercado. Y encima, el sistema de

depósito de claves de Clipper tenía fallos. Un atacante podría alterar la copia cifrada de la clave, de tal forma que el gobierno no pudiera descifrar las comunicaciones[25].

El sistema de depósito de claves basado en el chip Clipper, anunciado a bombo y platillo por la administración Clinton en 1993, estaba muerto a todos los efectos en 1996. Cuando el algoritmo fue desclasificado en 1998, los potenciales usuarios decidieron que su paciencia con la NSA se había agotado, y nunca le dieron una oportunidad a Skipjack. La industria se adaptó a los tiempos modernos, y adoptaron variantes de algoritmos como DES (que también era estándar del gobierno) y otros de la industria privada.

¿FALLOS O LICENCIAS LITERARIAS?

Para ser justo con el señor Brown, algunos de los “fallos” que le atribuyen sus detractores pueden clasificarse en el apartado de licencias literarias. El del “cuadrado perfecto de César” es uno de ellos. Podemos preguntarnos por qué Brown se molestó en inventar un sistema de cifra antigua inexistente y dar una referencia falsa a César, pero como autor está en su perfecto derecho. La protagonista Susan Fletcher menciona asimismo un *principio de Bergofsky*, según el cual, si probamos suficientes claves, forzosamente habremos de dar con la correcta. Este principio, que sencillamente describe un ataque de fuerza bruta, no existe con tal nombre, pero de nuevo puede atribuirse a una licencia por parte del autor para dar más emoción a la historia.

Otra licencia literaria es precisamente la “clave” de la trama del libro. El problema con el sistema de cifrado Fortaleza Digital es que ni siquiera probando todas las claves puede descifrarse el mensaje. El autor inventa un concepto llamado “texto llano rotatorio”. Se supone que el algoritmo de cifrado va cambiando el texto llano según una variable temporal. Es decir, el truco no es sólo encontrar la clave, sino de hacerlo en el momento oportuno. Podemos imaginarnos que, en ese caso, bastaría probar claves más largas (es decir, combinaciones de claves y tiempos), con lo que el número de posibilidades sería mayor pero asimismo finito. Sería muy discutible si eso podría ser factible, pero de nuevo estamos jugando en el campo de la licencia literaria. No tiene por qué ser cierto, sólo parecerlo.

Más peliaguda se pone la cosa cuando el autor habla de la criptografía de clave pública (PKC). La base de la PKC consiste en que hay dos claves diferentes, una para cifrar y otra para descifrar. Si usted quiere cifrarme un mensaje, puede usar mi clave pública[26] y usarla para cifrar. Yo luego tomaré mi clave privada, la activaré con mi contraseña y obtendré el mensaje. Sin embargo, Brown afirma que la PKC es un mero programa de software, creado por algunos “programadores empresariales” como respuesta a la pérdida de privacidad de los años noventa. Afirma en su libro: “*La única manera de descifrar el mensaje es introducir la frase de contraseña del*

remitente (pass-key) una serie secreta de caracteres que funcionan como un número PIN en un cajero automático". Parcialmente cierto, aunque también necesita la clave privada. Y además, son la contraseña y clave del destinatario, no las del remitente. Da la impresión de que Brown, sencillamente, no ha entendido el concepto de criptografía de clave pública.

Luego, como para mostrar lo difícil que se pusieron las cosas para la NSA, dice que "*los códigos que [la NSA] afrontaban ya no eran simples cifras de sustitución descifrables mediante lápiz y papel milimetrado, sino funciones hash generadas por ordenador, que combinaban la teoría del caos y los alfabetos simbólicos múltiples para cifrar mensajes…*"

En este punto de la novela (¡y estamos en el capítulo 4 de 128!) podemos ya dudar seriamente de los conocimientos criptográficos de Dan Brown. Ya no se trata de licencias literarias, sino de ignorancia manifiesta. Según este autor, todos los sistemas de cifrado anteriores a los años 90, la máquina Enigma, los algoritmos DES, IDEA, CAST, no llegan más que al nivel de "sistemas de lápiz y papel". No solamente eso, sino que introduce conceptos inexistentes en criptografía, como teoría del caos, y los mezcla con funciones hash, que se utilizan como ayuda en los protocolos de firma digital.

DECIDIDAMENTE, FALLOS

Criptográficamente hablando, el nivel del libro baja conforme avanzamos páginas. Mientras Susan Fletcher cavila sobre la estructura de Fortaleza Digital, repasa mentalmente otros algoritmos de cifrado: "*Había cientos de ellos en el mercado: PGP, Diffie-Hellman, ZIP, IDEA, El Gamal*". En un intento por da la impresión de que domina el asunto, la protagonista mezcla churras con merinas. Para que conste:

—IDEA es un algoritmo de cifra simétrico.

—El Gamal es un algoritmo de firma digital, con una variante que puede usarse como sistema de cifrado asimétrico.

—Diffie-Hellman es un protocolo criptográfico, pero no es un algoritmo de cifrado, sino de intercambio de claves.

—PGP no es un algoritmo en absoluto. Es un programa informático diseñado para cifrado y firma. Combina algoritmos de clave simétrica, criptografía de clave pública, y un sistema de firma digital.

—ZIP (y aquí es donde la cosa tiene más gracia) ¡es un programa de compresión de datos! Tiene una opción de protección mediante contraseña, para lo cual utiliza el algoritmo de cifra AES.

Es decir, en una sola línea ha mezclado algoritmos de cifra, de firma, de intercambio de claves, y dos programas informáticos, y pretende dar la impresión de

que tratan de lo mismo. Realmente, parece que el autor se ha limitado a mirar en el índice de un libro sobre criptografía y escoger los términos que le sonaban mejor. Sus conocimientos de informática rozan lo patético, pero prefiero no comentarlos aquí.

Cerca del final, los protagonistas se enfrentan a un mensaje cifrado: paquetes de cuatro letras. Susan Fletcher las examina y, con aires de entendida sentencia "*agrupaciones alfa de cuatro bits*". Creo que es un buen momento para explicar la diferencia entre bits y bytes. Un bit (**bi**nary digi**t**) es un número binario, esto es, un **1** o un **0**. Un byte (pronunciado "bait") es un paquete de varios bits, usado para codificar un carácter (letra, número, símbolo). El byte más usado es el de ocho bits, lo que permite codificar hasta 2^8=256 caracteres distintos.

En la afirmación de Fletcher tenemos varios problemas. El primero, evidente para el criptólogo más tonto, es que una letra no es un bit. En segundo lugar, quizá el lector no sepa qué es eso de una "agrupación alfa". Yo tampoco, pero imagino que con "alfa" se refiere a caracteres alfanuméricos, es decir, letras o números.

Incluso si lo hubiese dicho bien, yo señalaría una pega adicional, y es la pedantería implícita al hecho de referirse a lo evidente de forma tan complicada. Llamar "agrupación alfanumérica de cuatro bytes" a un sencillo paquete de cuatro letras es como si yo les dijese a mis compañeros de trabajo "¿os apetece una suspensión coloidal exotérmica?" cada vez que sea la hora del café.

No solamente eso, sino que Fletcher afirma a continuación que "*muchos sistemas de cifrado usan agrupamientos de cuatro bits*". Los sistemas de cifrado simétrico suelen ser de dos tipos: de flujo y de bloque. Los de flujo cifran la información bit a bit, y los de bloque se llaman así porque procesan la información en bloques. ¿Y de cuánto son esos bloques? Los más habituales son de 64 o 128 bits. Yo, la verdad, no conozco ni uno sólo de 4 bits. Ni uno. Puede que haya alguno por ahí, hecho como entretenimiento por algún criptólogo aburrido, pero no es la norma habitual ni mucho menos.

Podríamos ser generosos y suponer que Susan Fletcher, que no hace más que confundir bits con bytes, tal vez se estuviera refiriendo a bloques de cuatro bytes, lo que equivale a 32 bits. Tampoco es que haya muchos algoritmos de cifra que usen bloques de 32 bits, pero entra en el rango de lo verosímil. De hecho, aparecen un total de 64 letras cifradas, que alguien agrupa arbitrariamente en bloques de cuatro. Fletcher lo ve, sonríe y afirma: "*Me suena mucho. Bloques de cuatro. Igual que Enigma*".

En este punto, no puedo menos que detenerme y chillar. La máquina Enigma, usada por Alemania para proteger sus comunicaciones durante la Segunda Guerra Mundial, cifraba las letras una a una. Los mensajes cifrados se dividían en paquetes de idéntica longitud a efectos de facilitar la transmisión, y solamente para eso. Para rematar la faena, los mensajes cifrados con Enigma se enviaban en paquetes de cinco.

¡Cinco, señorita Fletcher, no cuatro! Eso no era un dictado de los criptoanalistas alemanes sino un convenio internacional según el cual las "palabras" cifradas debían tener una longitud de cinco letras.

En esta competición por bajar el listón de conocimientos criptográficos interviene hasta el jefe de Susan Fletcher: "*El director asintió. Enigma era la máquina de escribir en código más famosa de la historia, la bestia codificadora de doce toneladas de los nazis*". No tengo ni idea de a qué máquina se estará refiriendo, pero seguro que no tiene en mente la Enigma. En primer lugar no era "una máquina de escribir códigos". Aunque la Armada alemana tenía un accesorio llamado *Schreibmax* capaz de escribir los mensajes recibidos por una Enigma, lo habitual es que los mensajes fueran transcritos por los operadores de radio; la máquina Enigma no escribía nada por sí sola.

Reconozco que este detalle compite en pedantería con la afirmación de Fletcher sobre las agrupaciones alfanuméricas de cuatro bytes. Culpable. Lo que no admito es que un criptólogo mínimamente conocedor de la máquina Enigma la llame "bestia codificadora de doce toneladas". ¿Doce toneladas? El modelo militar tenía una masa aproximada de entre once y doce kilogramos. ¡Kilogramos, señor Brown, no toneladas! Y, a pesar de haber sido usado por todas las unidades civiles y militares de la Alemania nazi, no tiene aspecto de bestia. Al contrario, es un elegante artefacto en una caja de madera, con aspecto de inofensiva máquina de escribir. Yo mismo he visto algunas, y les aseguro que de bestia no tienen nada[27].

Hay otras referencias criptográficas, no erróneas sino ya totalmente disparatadas, pero prefiero dejárselas a usted, si es que aún le quedan ganas de comprarse el libro y leerlo. Sí hay un detalle que, aunque no es culpa del autor, sí desluce bastante el libro. Se trata de un problema de traducción. Todos sabemos que hay juegos de palabras en un idioma que resultan intraducibles en otro. En la película *Los Fisgones*, la Agencia de Seguridad Nacional (ASN) se convierte en la *Agencia Sin Nombre*, en alusión a un viejo chiste según el cual las siglas NSA realmente significan *No Such Agency* (no existe tal agencia), o bien *Never Say Anything* (nunca digas nada).

En este caso, Susan Fletcher sufre lo indecible intentando descifrar un mensaje de su novio. En inglés dice:

Please accept this humble fax / My love for you is without wax.

(Te ruego que aceptes este humilde fax / mi amor por ti es sin cera).

El juego de palabras está en las palabras *without wax* (sin cera). Dan Brown lo explica así en su libro: "*El secreto que ocultaba «sin cera» era demasiado tierno. Sus orígenes eran antiquísimos. Durante el Renacimiento, los escultores españoles que cometían errores mientras tallaban estatuas de mármol caras disimulaban sus defectos con cera. Una estatua que carecía de defectos y, por lo tanto, no necesitaba retoques era alabada como una «escultura sin cera». La palabra inglesa «sincere»*

provenía de la española «sincera», sin cera".

No soy lingüista, así que me resulta difícil juzgar en este tema. Según una hipótesis, la etimología de *sincero* es, efectivamente, esa, pero proviene del latín *sine cera* (sin cera), usado por los escultores romanos. Otra posible raíz etimológica afirma que se deriva del vocablo *sincerus* (limpio, puro). Punto para Brown. Por supuesto, podríamos criticar el hecho de que una criptóloga con un cociente intelectual de 170 no atine en un detalle tan evidente, pero ya sabemos que cuando uno está enamorado, no siempre razona con claridad.

El problema está en la versión española. *Sincera - sin cera* es una relación tan evidente que salta a la vista. Peor aún, amor es masculino y sincera es femenino, lo que enseguida nos sugiere la forma de resolver el enigma. Imagino que el traductor hizo lo mejor que pudo, dadas las circunstancias, pero el hecho es que un juego de palabras en inglés colocado como acertijo para el lector pierde toda su fuerza en la versión castellana.

Este libro me decepcionó mucho. De verdad. No se trata tan sólo del argumento, los personajes, las persecuciones y demás estereotipos. Se trata de que el autor ha creado un thriller de ficción criptográfica sin saber lo más mínimo sobre criptografía, y sin preocuparse por ello. Es como si *El Código da Vinci* ubicase el museo del Louvre en Uganda y convirtiese a Leonardo en un delantero centro del Arsenal.

A la vista de ello, me resulta sorprende leer la página de agradecimientos. Entre otros, cita a "*dos ex-criptógrafos de la NSA, quienes hicieron contribuciones valiosas mediante servidores anónimos, sin ellos, este libro no se habría escrito*". Brown intenta aquí convencer al autor de que el libro que está a punto de leer está inspirado en información confidencial procedente de profesionales. A la vista del resultado, una de dos: o los dos ex-criptógrafos fueron expulsados de la NSA por zoquetes, o ha sido una inteligente labor de sabotaje.

O bien, sencillamente, el autor es un zote. En 2007, cuando hice mi primer crítica al libro, entré en su web www.danbrown.com, sección de investigación/fotos. Dan Brown aparece en la plaza de San Pedro de Roma, acompañado por dos escoltas armados de la —cito textualmente en inglés— "*Vatican Guardia Civil*". Quisiera creer que el autor estaba siendo irónico, pero después de leer la descripción que hace de Sevilla en su Fortaleza Digital, me lo creo todo. No me atrevo a seguir mirando y cierro el navegador[28].

Si quieren leer otra interesante crítica sobre *Fortaleza Digital* desde el punto de vista criptográfico, les recomiendo el artículo "un libro prescindible" de Fernando Acero[29]. También él compró el libro en inglés, esperando en un aeropuerto. Lo único que espero es que jamás hagan la película.

EL CASO DEL CRIPTOANALISTA BRUJO

La criptografía y el ocultismo han estado unidos durante gran parte de su historia. Es algo inevitable. La criptografía se dedica a la ocultación de la información, y su uso implica un ritual (ahora lo llamamos algoritmo) que ha de llevarse a cabo sin cometer un solo fallo. Cuando funciona, consigue convertir la información clara en un batiburrillo ilegible; y a la inversa. A veces, parece cosa de brujería. La magia, por su parte, intenta "descifrar" los misterios de la naturaleza, por medio de una serie de pasos que también han de cumplirse escrupulosamente. Suele ir acompañada de palabras mágicas, y durante siglos los *abracadabra* y los *hocus pocus* han actuado como contraseñas de lo oculto.

De sobra conocida es la siguiente anécdota. A mediados del siglo XVI, el criptoanalista francés Francois Viète consiguió romper una cifra española. El rey español, muy molesto, denunció a Viète ante el Papa, acusándole de brujería y magia negra. Como pueden imaginarse, solamente le sirvió para ganarse las carcajadas de toda Europa. Esta historia/leyenda/anécdota/bulo apareció por vez primera en la Enciclopedia de Jaques Auguste de Thou, un historiador francés de finales del siglo XVI. La historia no está verificada, y hay motivos para dudar de ella (los franceses nunca nos han querido, y mucho menos en aquella época), si bien hay que tener en cuenta que de Thou llegó a consejero de Estado y, por tanto, tenía acceso a información de alto nivel.

Este episodio fue añadido a nuestra Leyenda Negra. Ahora, por el contrario, hay pruebas que demuestra que los servicios criptológicos de Felipe II eran de los mejores en aquella época, incluyendo el uso del criptoanálisis, y en uno de sus primeros actos como soberano ordenó cambiar las cifras de su padre. ¿Sería alguien así capaz de creer que la criptografía es asunto de brujas?

La hipótesis que me parece más probable es aquella según la cual Felipe II, buen conocedor del poder de la criptografía, quiso poner en un brete al mejor criptoanalista francés ante un enemigo tan poderoso como la Iglesia (recordemos que Viète trabajaba para el protestante rey Enrique IV, en lucha entonces contra la facción católica francesa). Puestos a acusar, era preferible no mentar el asunto del criptoanálisis y centrarse, en su lugar, en la acusación de brujería, delito ante el que podía actuar la Inquisición en cualquier país. En aquella época en la que poca gente sabía leer, magia negra y criptografía podían fácilmente confundirse.

Medio milenio antes, el papa Silvestre II (945-1003) re-descubrió un sistema de taquigrafía de tiempos romanos (con reminiscencias criptográficas) conocido con el nombre de notas tironianas. Se trataba de un hombre erudito, cuyos conocimientos le pusieron en ocasiones en un serio compromiso. El "papa del año 1000" fue incluso acusado de tener un pacto con el diablo y de probar disciplinas como la cábala.

La historia registra el nombre de otros erudito mitad criptógrafo mitad ocultista: Enrique de Aragón, señor de Villena (1384-1434), nieto bastardo de Enrique II de Castilla y miembro de la casa real de Aragón por vía paterna. Hombre de gran erudición, que destacó por la traducción de textos como la Divina Comedia de Dante. Su interés por las ciencias, desde la medicina hasta la alquimia, le valió fama de brujo; no en vano pasó a la historia con el apelativo de *el nigromántico*.

Hasta tal punto fue así, que a su muerte gran parte de su librería fue quemada por el obispo Lope de Barrientos, quien actuaba bajo órdenes del rey Juan II de Castilla. Sin embargo, sobrevivió un curioso libro denominado Libro del Tesoro, que contiene una parte cifrada. Una nota en la obra indica que se hizo porque *non fuese entendida salvo de ome bueno e sabio*. Se sabe que de 1412 a 1416 el señor de Villena trabajó como ayudante personal de su primo el rey Fernando. Allí pudo haber aprendido técnicas de cifrado, o bien haber instruido a la propia cancillería aragonesa en la materia.

En realidad, la ligazón entre criptografía y ocultismo nunca ha desaparecido del todo. Durante la Segunda Guerra Mundial, los alemanes protegían sus comunicaciones mediante la máquina cifradora Enigma. Teniendo en cuenta que la Real Academia define la criptografía como "a*rte de escribir con clave secreta o de un modo enigmático,*" el nombre no podía estar mejor escogido. El esfuerzo hecho por los Estados Unidos para romper los códigos japoneses fue ocultado con el nombre código *Magic*. Yo mismo fui entrevistado una vez por la revista de ocultismo *Más Allá de la Ciencia*; por qué su consejo editorial pensó que el tema de la criptografía durante la Segunda Guerra Mundial encajaría bien en una publicación de esoterismo es algo que se me escapa[1].

Es el momento de introducir a nuestro primer protagonista de hoy. James Randi, un antiguo mago, es conocido en el mundo escéptico como "cazador de brujos" y azote de los médiums. En los años setenta saltó a la fama al acusar de charlatán nada menos que al famoso mentalista Uri Geller. En 1996, fundó la Fundación Educativa James Randi (JREF, *James Randi Educational Foundation*) para intentar examinar las afirmaciones paranormales en condiciones controladas[2]. Desde hace años, Randi mantiene un desafío particular: dará un premio de un millón de dólares a cualquiera que pueda demostrar, bajo condiciones controladas de experimentación, ser poseedor de poderes ocultos o paranormales[3].

Hasta la fecha, nadie ha llegado a pasar siquiera las pruebas preliminares, no digo ya ganar el desafío. Sin embargo, hay una persona que afirma haberlo conseguido. Entra aquí nuestro segundo personaje del día: Matt Blaze, profesor de la Universidad de Pensilvania, experto en criptografía y seguridad informática. Y muy bueno: ya en 1994 descubrió una forma de saltarse la protección del chip Clipper, que la Administración Clinton intentaba vender como la solución para combinar las ansias

de privacidad de los internautas con las necesidades de las fuerzas de seguridad (contenía un algoritmo que incluía un dispositivo de “key escrow”, o depósito de claves, así que nos hizo un buen favor a todos al mostrar el fallo). También publicó estudios sobre el propio esquema de “key escrow” (su conclusión: es una mala idea), el sistema de interceptación "Carnivore" del FBI, y testificó en diversas ocasiones ante comisiones del Congreso de EEUU y del Parlamento Europeo.

Diez años más tarde volvió a irritar a las autoridades al narrar la odisea que le supuso cruzar la frontera de EEUU con un teléfono que utilizaba cifrado fuerte. Puesto que en aquella época eso requería un permiso de exportación especial, Blaze publicó su experiencia en un divertido artículo titulado “*mi vida como traficante de armas*”[4]. Con semejantes antecedentes, creo que hemos dejado claro que Matt Blaze tiene tanto talento como sentido del humor. Por ello, nadie sabía bien cómo tomarse su afirmación de que había ganado el “desafío del millón de pavos” de Randi.

Esto fue lo que sucedió. Exasperado ante las mentiras vertidas por un “visor remoto” (persona que afirma que puede realizar una proyección astral y enviar su alma a otro lugar), en enero de 2007 Randi lanzó de nuevo su desafío en una versión modificada: cualquier persona que se pudiese proyectar astralmente hasta su oficina y descubriese un objeto guardado en un archivador ganaría el millón de dólares[5]. Para animar a los posibles candidatos, incluyó la siguiente descripción:

```
0679
4388
66/27
5 -14
```

El motivo de revelar esos datos era que cualquiera pudiese verificar a posteriori que no había existido trampa alguna. Randi declaró que, doce meses después, revelaría la “clave de decodificación”. No contaba con Matt Blaze. El criptógrafo se “concentró,” y enseguida descubrió el contenido de la caja. El proceso fue descrito por el propio Blaze de la siguiente forma:

“Hemos [lo descubrió conjuntamente con Jutta Degener] visualizado con éxito el contenido de la caja desafío de Randi. Lo conseguimos desde más de 1500 km de distancia, tan sólo mediante concentración mental y la aplicación de nuestros talentos (o debería decir dones), y sin acceso físico o información interna. Podemos revelar ahora el contenido: una pequeña rueda, o disco circular, como un DVD o CD”.

¿Les pica la curiosidad? Bueno, la verdad es que ni Blaze es vidente (que sepamos), ni usó sus habilidades criptográficas. O no del todo, aunque es indudable que una vida dedicada al manejo de los números le habrá predispuesto. En cualquier caso, conjeturó que los primeros dígitos podrían corresponder a un número de

registro ISBN, de esos que identifican un libro. De ese modo, “0679 4388 66” se convierte en el ISBN 0-679-43886-6, correspondiente al diccionario *Random House Webster’s College Dictionary*, edición de 1995.

Los siguientes números 275 -14 pueden corresponder a la palabra situada en la página 275, línea 14 contando desde abajo (por el signo menos). ¿Y qué aparece en esa línea? La entrada correspondiente a la definición de “compact disc”. Randi confirmó la veracidad de esa afirmación: efectivamente, se trataba de un CD.

Blaze aprovechó la ocasión para escribir una divertida entrada en su blog. Bajo el título de “*James Randi me debe un millón de dólares*,” aprovecha la oportunidad para explorar algunas facetas criptográficas. Entre otras cosas, postuló que con un poco de imaginación el mensaje hubiera podido “descifrarse” casi de cualquier forma. Por ejemplo, tomando el número 14 no como un número de línea sino como “columna 1, definición 4”, concluiríamos que la caja contenía un ejemplar del Manifiesto Comunista; Randi, por su parte, podría haber aprovechado esta ambigüedad para negar el premio[6].

El desafío de Randi ilustra un conjunto de problemas en los que no solamente hay que mantener un secreto sino también verificar que no ha sido cambiado. Digamos que Alicia, agente de bolsa, intenta convencer a Benito de que contrate sus servicios. Benito no se fía y quiere poner a prueba a Alicia, así que le pide una predicción sobre algunas empresas que vayan a ir bien en bolsa. Alicia está convencida de que Repsol y Telefónica van a subir mucho, pero ella tampoco se fía y piensa “si se lo digo, él podrá comprar las acciones por su cuenta y no necesitará de mis servicios”. Prefiere esperar un mes, y entonces le revelará a Benito cuál fue su elección. Ahora bien, cuando Repsol y Telefónica hayan subido y Alicia diga que esas eran las acciones que habría escogido, Benito no tiene ninguna posibilidad de verificar la validez de esa afirmación.

Sabemos que es muy fácil hacer predicciones a toro pasado; más difícil es hacerlas antes de que sucedan; y todavía más difícil es convencer a los demás de haber hecho una predicción sin revelar los resultados de esa elección. Para simplificar nuestro ejemplo, el secreto va a tomar una forma binaria. Podría ser una respuesta a la pregunta “¿Es Telefónica una buena inversión?” Sólo habría dos respuestas posibles: sí (1) y no (0). Ese secreto, que representaremos con la letra *b*, es lo que Alicia tendrá que proteger y Benito deberá verificar.

Vamos a describir un esquema que utiliza algoritmos de cifrado simétrico. Alicia cifra su secreto *b* por medio de un algoritmo de cifrado *E* que utiliza una clave *k*. El resultado, *Ek(b)*, queda bajo la custodia de Benito. Un mes después, Telefónica ha subido como la espuma. Es el momento en que Alicia llama a Benito y descifra el mensaje en su presencia. El resultado es *b*, que vale 1: Telefónica sí es una buena inversión. Puede que Benito siga sin convencerse, porque a fin de cuentas la subida

de Telefónica en bolsa puede haber sido cuestión de suerte. Pero Alicia puede hacer el mismo proceso con varias acciones. Que Alicia acierte en el comportamiento de todas ellas ya resulta mucho menos probable. Finalmente Benito se da por satisfecho y contrata los servicios de Alicia, convencido de haber hecho un buen negocio.

Por desgracia, lo único que esta "demostración" prueba es que Alicia es, o bien una gran agente de bolsa… o bien una timadora de primera. El problema aquí consiste en que Alicia controla el proceso de cifrado, y eso incluye la elección de la clave para cifrar *k*. Recordemos que su predicción es que Telefónica subirá, lo que significa que *b*=1. Ella en principio ha escogido una clave *k* al azar, y lo que le ha entregado a Benito es *Ek(b)*. La cuestión es la siguiente. Si usamos la misma clave para descifrar, evidentemente recuperaremos el mensaje original, ya que es un algoritmo de cifra simétrica; de ese modo, al descifrar obtenemos *Dk(Ek(b))=b*. O dicho en palabras: ciframos con una clave el bit 1, lo desciframos con la misma clave y obtenemos 1.

¿Pero qué pasaría si descifrásemos el mensaje con otra clave distinta? Si el algoritmo es bueno, su cifrado será indistinguible de un conjunto de bits aleatorios. Eso significa que, en promedio, la mitad de las veces que intentemos descifrar el mensaje nos saldrá el bit 1, y la otra mitad nos saldrá el bit 0. Alicia se aprovecha de ello, y cuando está sola en su casa descifra el mensaje con otra clave distinta. Es decir, efectúa la operación *Dk'(Ek(1))*. El truco consiste en ir probando claves k' hasta que una de ellas nos de cómo resultado un bit cero, es decir, *Dk'(Ek(1))*=0. Y ahora Alicia vuelve a cifrar el resultado, pero esta vez con la clave *k'*. Al hacerlo, tenemos que:

$$Ek'(Dk'(Ek(1))) = Ek'(0)$$

Ahora Alicia tiene dos claves que toman dos bits distintos y los cifra de la misma forma: *C=Ek(1)=Ek'(0)*. Benito tiene el mensaje cifrado *C*, pero no sabe si es el resultado de cifrar 1 con la clave *k* o 0 con la clave *k'*. Ese es el detalle que usará Alicia para timar al incauto Benito. Digamos que Telefónica ha subido en el mes de prueba. A Alicia le interesa que el descifrado del mensaje sea *b*=1, así que efectúa la operación de descifrado con la clave *k*. El resultado es *Dk(C)*=1. Justo como había predicho. Si, por el contrario, Telefónica se ha estrellado contra el parqué, lo que hace es descifrar con la clave k' y obtener Dk'(C)=0. Justo como había predicho. En ambos casos, el mensaje cifrado "prueba" lo que Alicia quiere que pruebe.

El truco de Alicia es una variante de lo que se conoce como *cifrado negable (deniable encryption)*, una técnica que permite negar la existencia de un mensaje cifrado. Tiene dos variantes. En una de ellas, un mensaje cifrado adopta una forma que impide reconocerlo como tal, de manera que no se puede demostrar la existencia del mensaje sin la clave adecuada; en otra, un mensaje cifrado se puede descifrar con

dos claves distintas, arrojando dos textos llanos distintos.

Imagínese el lector, por ejemplo, un mensaje enviado por una multinacional a su filial en España. Cuando los directores regionales reciben el mensaje, lo descifran con la clave A y leen “están ustedes haciendo un gran trabajo”. Eso tranquiliza a la plantilla. El consejo de dirección, sin embargo, utiliza la clave B y obtienen el mensaje “hay que hacer recortes, cierren las delegaciones de Málaga, Zaragoza y Valencia”. Dos públicos reciben, a partir del mismo texto cifrado, mensajes muy distintos.

En el caso de Alicia y Benito, el protocolo puede mejorarse. La forma de hacerlo es hacer que Alicia no cifre solamente el bit *b*, sino una combinación *R+b*, donde *R* es una cadena de bits aleatorios. Para hacer trampas, Alicia tendría que encontrar dos claves *k, k’* tales que *Ek(R+1)=Ek’(R+0)*, algo inviable si R es aleatorio y de tamaño suficiente. Cuando Alicia descifre el mensaje cifrado, Benito podrá leer “*R+b*,” lo que le revelará la elección de Alicia al tiempo que le permite reconocer su propia *R*.

El dilema de Alicia bolsista y Benito inversor puede también verse como ejemplo de una serie de problemas llamados “de conocimiento cero”. En este tipo de problemas, Alicia tiene que convencer a Benito de que conoce un secreto. La forma evidente de hacerlo es contárselo, pero en el momento que lo haga dejará de ser un secreto. Un campo de aplicación es el de las contraseñas. Vivimos en un mundo donde la forma de demostrar nuestra identidad es mediante contraseñas: en el cajero automático, en el móvil, en el acceso al trabajo. El usuario no suele pensar que los verificadores pueden no ser quienes afirman ser. Una web bancaria puede ser falsificada, o un cajero trucado, y en esos casos, el cliente estará facilitando su contraseña a terceros no autorizados. Los verificadores legítimos (bancos, empresas de telefonía) concentran sus esfuerzos en “generar confianza,” pero el hecho es que al cliente le puede quedar siempre la duda: ¿será esta gente quien realmente afirma ser?

Las pruebas de conocimiento cero (ZKP) pueden ayudar en estos casos. Vamos a ver un ejemplo de cómo una persona puede demostrar la existencia de un secreto sin revelarlo en absoluto. El artículo original, publicado en 1990, está escrito en forma de cuento de las Mil y Una Noches[7]. Si no les importa, voy a actualizarlo levemente.

Éranse una vez dos arqueólogos, Indy y Lara, que buscaban el secreto de la Sala Oculta. En las profundidades de la selva, el Monte Perdido tiene una entrada (A), que tras un largo pasillo conduce a un vestíbulo (B). De allí arrancan dos pasadizos (C y D), que aparentemente acaban en una pared. Para un observador casual son vías sin salida, pero quien se plante al final de uno de los pasadizos y grite las palabras mágicas se encontrará con que la Sala Oculta aparece de la nada, conectando ambos pasadizos.

Indy conocía las palabras mágicas que revelaban la existencia de la Sala. Lara deseaba conocer ese secreto, así que hizo un trato con Indy: le financiaría su próxima

expedición a cambio del secreto. Indy aceptó el trato, pero conforme se acercaban a la Sala empezó a dudar. ¿Y si él cumplía su parte pero Lara se echaba atrás? Por su parte, Lara no estaba dispuesta a soltar el cheque hasta comprobar que las palabras mágicas funcionaban; que Indy era perro viejo y tal vez guardase un as en la manga.

Por fortuna, ambos eran personas razonables, y cuando llegaron a la entrada del Monte Perdido ya habían diseñado una prueba de Conocimiento Cero. El primero en entrar fue Indy. Con paso firme, llegó al vestíbulo y escogió uno de los pasillos. Después entró Lara, quien se quedó en el vestíbulo. Desde allí gritó con voz alta y clara "Indy, vuelve por el pasillo C". Indy así lo hizo. A continuación, repitieron el proceso de nuevo, escogiendo el mismo pasillo o bien otro distinto (a capricho de Lara). Luego otra vez. Y otra más. Recordando la historia de Alí Babá y los cuarenta ladrones, hicieron este proceso un total de cuarenta veces. Finalmente, Lara se convenció de que Indy conocía el secreto que llevaba a la Sala Oculta.

La clave del protocolo consiste en que ni Lara sabe qué pasillo ha tomado Indy, ni Indy sabe qué pasillo de vuelta le va a indicar Lara. Si las elecciones se toman al azar, hay un 50% de probabilidades de que Indy vuelva por el mismo pasillo por el que entró, lo que ciertamente no prueba nada; pero también hay un 50% de probabilidades de que tenga que salir por un pasillo distinto, y eso no podría hacerlo más que diciendo las palabras mágicas y atravesando la Sala Oculta.

Un Indy tramposo podría haberse arriesgado y confiar en la suerte. Lara lo sabía y no se hubiera quedado satisfecha con la prueba. Pero si Indy repitiese con éxito el proceso un total de 40 veces, la probabilidad de haber "tenido suerte" hubiera sido de una entre un billón. Demasiada suerte incluso para él. Lara concluyó que Indy realmente conocía el secreto de la Sala Oscura y cumplió su parte del trato.

Algún tiempo después, tuve la oportunidad de cenar con ambos arqueólogos (recuerden, esto es un cuento). Tras un magnífico Oporto, me contaron toda la historia. Yo, que cuando he pillado el punto suelo volverme algo malicioso, les dije: "Entiendo por qué ese proceso tan complicado funciona, pero hay algo que aún se me escapa. Lara, ¿por qué no te limitaste a enviar a Indy por un pasillo y esperar a que apareciese por el otro?" Debí haber grabado la cara que pusieron los pobrecitos. Realmente, se hubieran ahorrado una buena caminata por el interior del Monte Perdido.

En su defensa, debo decir que ellos no son criptógrafos profesionales y se limitaron a copiar la idea de un experto y aplicarla a su caso particular. En general, las pruebas de conocimiento cero del tipo que hemos visto requiere una serie de desafíos por parte de la persona que quiere verificar un secreto, seguidos por una elección hecha por el poseedor del secreto. Cada paso del proceso solamente puede hacerse con seguridad si realmente se posee el secreto, en tanto que un timador solamente tiene una probabilidad de acierto debido al azar.

James Randi tampoco es criptólogo profesional, y seguro que no pensó que tendría que enfrentarse a uno. Afortunadamente, en esta ocasión no tuvo por qué preocuparse. Caballero ante todo, Blaze mostró su apoyo al trabajo de la fundación Randi. ¿Y saben lo mejor? En un arrebato de generosidad, renunció al premio. "*Por esta vez,*" dijo. Si hubiese sido un vidente de esos que salen en la tele, podría haberlo "revelado" en un programa de máxima audiencia para hacerse mundialmente famoso. Sin embargo, no sólo no hizo tal cosa, sino que renunció al premio de un millón de dólares. Hasta la fecha, sigue sin reclamarlo.

Para su sorpresa, el caso acabó en los tribunales. No, bueno, quiero decir de modo simulado. El asunto Randi-Blaze fue aprovechado por un profesor de la Facultad de Leyes de la Universidad Stetson, en Florida, quien tuvo la genial ocurrencia de explicarlo a los alumnos, aderezarlo un poco y utilizarlo como cuestión en un examen. Los alumnos debían explicar quién puede demandar a quién, por qué motivos, y quién ganaría[8].

Por su parte, Randi se lo tomó con humor. Según contó en su revista electrónica, cambió el contenido del archivador, y ahora nos da la siguiente pista[9]:

```
BSQN ISIO QWOD QPIE
```

Que yo sepa, nadie ha descifrado este reto ni reclamado el millón de pavos. ¿Se atreve usted a intentarlo? Le deseo suerte. Y, por supuesto, en caso de éxito espero mi porcentaje.

A CORAZÓN ABIERTO

Estamos tan acostumbrados a ver objetos interconectados electrónicamente que ya ni nos llama la atención. Escribo estas líneas rodeado de todo tipo de aparatos que emiten y reciben información de forma inalámbrica, desde móviles a altavoces, pasando por la estación meteorológica o la llave que me permite abrir mi coche a distancia.

Pero hay un instrumento cuya conectividad solamente ahora se está comenzando a aprovechar. Los beneficios potenciales son enormes, pero también las implicaciones de seguridad. Las soluciones actuales están siendo desarrolladas y probadas, y su correcta aplicación es literalmente un asunto de vida o muerte. Porque ese instrumento es el propio ser humano. Bienvenidos a la seguridad en el campo de los marcapasos, desfibriladores, bombas de insulina y demás aspectos de la biónica moderna.

Como todos sabemos, los dispositivos genéricamente denominados marcapasos son necesarios para que los corazones de miles de personas sigan latiendo con normalidad. El problema con los marcapasos es el mantenimiento. Es evidente que abrir el pecho del paciente cada vez que haya que ponerle una pila nueva no es una buena idea. Felizmente, el fenómeno físico conocido como inducción magnética resuelve ese problema; y si no, las baterías actuales pueden mantener el aparato funcionando durante años. La construcción de marcapasos evita, o minimiza, la mayoría de fallos por funcionamiento mecánico o eléctrico.

Los marcapasos tienen su propio software (*firmware*), y algunos pueden ser configurados, lo que los hace susceptibles a errores de programación. Un estudio realizado en Estados Unidos entre 1990 y 2000 mostró que más del 40% de los fallos de los marcapasos y cardio-desfibriladores implantables se debieron a fallos en el firmware, lo que constituye en términos absolutos casi un cuarto de millón de casos[1]. Y eso en aparatos que apenas pueden programarse.

Quizá fuese mejor no programarlos en absoluto. Pero eso no es una opción. Lejos quedaron los tiempos en los que el único control posible sobre un marcapasos era darle al botón de encendido. Los marcapasos actuales son programables y permiten al médico escoger el ajuste óptimo para cada paciente. Los desfibriladores automáticos implantables (DAI) detectan arritmias y las corrigen proporcionando al corazón un “chispazo” eléctrico. Los modelos más modernos pueden incluso conectarse al médico o al hospital, enviando telemetría y permitiendo un seguimiento a distancia de la evolución del paciente.

Los marcapasos son solamente la punta del iceberg. Las aplicaciones de pequeños dispositivos insertados quirúrgicamente en el cuerpo se multiplican. Un medidor de glucosa subcutáneo indicaría automáticamente el estado del paciente, y podría

comunicarse con la bomba de insulina del paciente para regular así el flujo de insulina según las necesidades del organismo. Una cadena de sensores colocados en el cuerpo de un paciente proporcionaría información sobre sus parámetros vitales. Un chip insertado en la piel podría ser usado para almacenar la historia médica del paciente, como el famoso VeriChip[2], puesto en circulación entre 2004 y 2010, que llegó a ser usado en una discoteca de Barcelona como sistema de control de acceso[3]. Chips de otro tipo podrán efectuar liberaciones de fármacos de forma controlada en el cuerpo, ayudando a tratar enfermedades como la osteoporosis[4].

Las posibilidades están limitadas tan sólo por nuestra imaginación, pero para que puedan ser aceptadas y usadas de forma eficaz es esencial que los problemas de seguridad queden primero resueltos. Los requisitos de acceso para un marcapasos no son los mismos que para un sensor de glucosa. Puede haber casos en los que la compañía de seguros tenga una necesidad legítima de acceder a datos (para fines de calcular la mensualidad de la póliza, por ejemplo) y casos en las que no. Una serie de reglas de restricción, útiles en circunstancias normales, pueden resultar contraproducentes en una emergencia.

Sin intención de ser exhaustivo, podemos establecer que un sistema de control de acceso a un implante médico (sea para enviar instrucciones o para recibir datos) debe cumplir propiedades como:

—Permitir un acceso sencillo y preciso a las partes autorizadas, sea un médico en la consulta o un sanitario en un accidente de tráfico.

—Proteger la privacidad de dichos datos frente a terceros no autorizados y permitir la identificación correcta del dispositivo en cuestión (no queremos manipular el marcapasos equivocado, ¿verdad?)

—Permitir al personal autorizado configurar el sistema e impedirlo a otros.

—Ser discreto, impidiendo a terceras personas tener conocimiento de su existencia.

—Garantizar su acceso a otros dispositivos autorizados (por ejemplo, un sistema de liberación de insulina debería poder acceder a los datos del medidor de glucosa para poder ajustar la dosis de forma automática)

—Respetar la decisión del paciente sobre quién ha de tener acceso a los datos, y en qué condiciones.

—Ser eficiente en términos de tamaño, tiempo y energía[5]

El problema se complica si tenemos en cuenta que los posibles atacantes abarcan casi todo el rango de posibilidades, desde el asesino a sueldo hasta el hacker curioso, y con motivaciones que van de la ganancia económica al inocente “porque puedo hacerlo”. Imagine un usuario no autorizado que programe un marcapasos para que se detenga. Una orden sencilla, y un IDC envía un chispazo al corazón cuando éste no lo necesita. O bien un sensor de glucosa envía una falsa medida y como consecuencia

no se envía una dosis de insulina[6]. Si el presidente de Estados Unidos lleva uno de esos aparatos, ya puede echar a correr Bruce Willis para salvarlo.

Si piensa que es un argumento de película muy cogido por los pelos, le doy a usted la razón. Dudo que el usuario típico de marcapasos sea objetivo de un complot internacional. Pero no hay que ir tan lejos. Los posibles móviles son múltiples, como indica un estudio de 2010: *"obtención de información privada para ganancia financiera o ventaja competitiva; daño a la reputación del fabricante del dispositivo; sabotaje por parte de un empleado disgustado, un cliente insatisfecho o un terrorista para producir daños personales o financieros; o simplemente la satisfacción del ego del atacante"*[7].

No es una preocupación teórica. En al menos dos ocasiones, hackers desconocidos asaltaron webs e introdujeron imágenes con colores rápidamente cambiantes, capaces de provocar ataques a personas epilépticos. Lo grave del caso es que las webs atacadas albergaban foros de apoyo a personas con epilepsia. La identidad de los atacantes se desconoce, aunque la descripción que dio un administrador de sistemas es, en mi opinión, bastante certera: *"una panda de gente muy inmadura, que se regodea en sus intentos de causar daño a la gente"*[8] [9].

Incluso ataques en apariencia inocentes pueden conllevar graves consecuencias. Recordemos que cuando hablamos de "ataque criptoanalítico" no nos referimos necesariamente a un asalto físico. Hablamos de tiempo de cálculo, espacio de claves, emisión de mensajes, captación de paquetes de datos, ese tipo de cosas. Interrogar electrónicamente a un sistema que tiene una batería limitada reducirá su vida útil. Cada vez que un paciente se suba al autobús, entre en un edificio con control de acceso o pase por un control aeroportuario, un lector de radiofrecuencia puede estar interrogando a su marcapasos. Si se diese el caso que su dispositivo implantado operase en la misma banda de frecuencias, tendría que responder, aunque fuese para decir "no, no soy un bonobús", y en cada respuesta el sistema pierde energía. Peor aún, mientras se encuentra en dicho estado no puede enviar información al médico o ser re-programado, lo que nos plantea un problema de Denegación de Servicio (DoS) especialmente peligroso.

Creo que a estas alturas está bien claro el problema. Vamos con la solución. Aunque lo que sigue puede aplicarse a cualquier dispositivo médico implantado, en lo que sigue supondremos que estamos hablando de marcapasos. Como hemos visto, hay muchos intereses que regular, desde el derecho de privacidad del paciente hasta la necesidad de acceso por parte de los equipos de emergencia. Olvidemos los detalles por un momento. Si los datos salen y entran, habrá que protegerlos, y eso significa criptografía.

Consideremos, en primer lugar, un cifrado simétrico, rápido y eficiente, con dos claves: una en el marcapasos, otra en posesión del paciente. En condiciones normales,

éste puede sacar su clave, insertarla en el aparato adecuado, y las lecturas saldrán en texto llano para ser utilizadas por el personal autorizado por el paciente.

Soluciones las hay, ya sea dar una copia de la clave a un tercero de confianza, guardar la clave en una tarjeta o habilitar una puerta trasera para equipos de emergencia. Todo ello crea nuevos puntos de vulnerabilidad. Las claves han de ser transmitidas, el tercero de confianza puede no estar disponible en un momento crítico (¡o no ser tan de confianza!), y las puertas traseras pueden ser mal utilizadas. Además de ello, requiere tiempo y calma, algo de lo que no suele andarse muy sobrado en situaciones de urgencia.

El principal punto débil de la criptografía simétrica consiste en que la clave de cifrado y de descifrado son la misma. Cualquier persona con acceso a dicha clave, sea un hacker o un médico autorizado, tendrá control del sistema. En ese caso, quizá fuera mejor considerar las ventajas de la criptografía de clave pública, donde las claves para cifrar (clave pública) y para descifrar (clave privada) son diferentes. La clave pública sería accesible para cualquier persona que la solicitase. Con una infraestructura de clave pública adecuada, cada hospital puede tener las claves públicas de todos los pacientes con marcapasos del país.

En cuanto a la clave privada, se almacenaría en el marcapasos, combinada con algún procedimiento para activarla a voluntad: una contraseña numérica escrita en papel, guardada en una tarjeta de memoria USB, en un colgante de ayuda médica, en un collar de plata con cristales de Swarowski, a elección del paciente.

Con la clave pública del paciente, el dispositivo de urgencias puede transmitirle al marcapasos una clave temporal para cifrar los datos que salen o las instrucciones que entran. La integridad de las comunicaciones puede garantizarse mediante sistemas de firma digital, lo que nos garantiza que ni el marcapasos ni el equipo del hospital intercambian información errónea o alterada. También permite garantizar que los interlocutores son solamente entidades autorizadas. En principio, problema resuelto con final feliz, particularmente en las áreas de autenticación e intercambio de claves.

Pero cuando usted lea “en principio”, debe acostumbrarse a traducirlo como “ni se le ocurra pensar que esto es tan fácil”“. La búsqueda de soluciones matemáticas teóricas no debe hacernos olvidar que, en el mundo real, estamos tratando con dispositivos que tienen graves limitaciones de tamaño, peso, energía y memoria. La criptografía de clave pública es onerosa en términos de tiempo de computación, y por tanto de energía, y un marcapasos tiene una capacidad de memoria y una energía limitadas. No se trata de un móvil con gran capacidad de cálculo, batería de tamaño respetable y un cargador con enchufes siempre a mano. Podemos aumentar la memoria del marcapasos merced a los avances de la técnica, pero no tanto la capacidad de la batería.

Es aquí donde entra una tecnología que parece creada a medida para la ocasión: la

Identificación por Radiofrecuencia, RFID (*Radio-Frequency IDentification*). Se trata de un sistema de almacenamiento de datos basados en pequeños y sencillos chips, que pueden interaccionar mediante ondas de radio de corto alcance. Los chips RFID son pequeños y muy baratos de fabricar, lo que facilita su comercialización en masa; en el apartado negativo, disponen de escasa capacidad para almacenamiento de datos. Su campo de aplicación inicial fue el de identificación de productos, actuando como etiquetas programables. Permite el seguimiento de paquetes en entornos logísticos complejos como servicios de mensajería o grandes superficies. Sirven para etiquetarlo todo, desde pantalones vaqueros a mascotas, pasando por entradas a grandes eventos. También se utilizan como medios de pago en transportes u otros sistemas de micropagos. Yo tengo uno en mi cartera, y me permite pagar el billete del autobús sin sacar la tarjeta. Incluso los pasaportes españoles actuales incorporan un chip de radiofrecuencia.

Los sistemas RFID activos pueden disponer de su propia fuente de energía, pero otros muchos modelos funcionan en modo pasivo, activándose mediante la señal de radiofrecuencia que recibe del exterior. Esto los convierte en buenos candidatos para un sistema de comunicación con dispositivos médicos implantados. Antes de considerarlos, han de solventar dos problemas para ser aceptable, ya no en el campo médico, sino en general: privacidad y seguridad.

Del mismo modo que no queremos una tarjeta chismosa que filtre al exterior el saldo que nos queda, no es buena idea que el RFID transmita información médica confidencial. Ya ha habido casos en los que terceros no autorizados intentaron acceder a la información de tarjetas RFID para su propio beneficio. Por poner un ejemplo, en marzo de 2008 se supo que los servicios de inteligencia británicos habían solicitado acceso total a la información transmitida por las tarjetas de transporte Oyster que se utilizan en Londres. En ese caso, lo que a James Bond le interesa saber no es cuántos viajes de metro te queden, sino dónde estás: rastrear las tarjetas Oyster proporciona gran cantidad de información sobre posición y hábitos de sus usuarios[10].

Ese mismo motivo —evitar el rastreo y preservar la propia privacidad— provocó en noviembre de 2012 un enfrentamiento entre las autoridades escolares de Tejas y una alumna que rehusó llevar una tarjeta con RFID. El colegio utilizaba este procedimiento para rastrear la posición de todos sus estudiantes, y cuando una alumna se negó a ello fue expulsada de modo fulminante[11]. El colegio afirmó que el sistema era necesario para reducir el absentismo escolar; por su parte, los padres de la joven han llevado el caso a los tribunales[12]. Ese es el gran problema de los dispositivos RFID: sea cual sea su propósito permiten el rastreo de la persona que los lleva, y de ahí se pueden deducir patrones de conducta.

Asimismo, debemos establecer salvaguardias para evitar que el chip sea hackeado

o modificado por usuarios no autorizados. Desde el momento en que se utilizan en medios de pago, cualquier posibilidad de alterar uno de estos chips de forma no autorizada permite ganancias económicas. Ya que estamos con James Bond, esto le permitiría viajar gratis en metro. O pasar por las autopistas sin pagar. O, sencillamente, hacerse con un pasaporte falso. Idénticas consideraciones podemos hacer con los dispositivos médicos, donde lo que uno se juega no es dinero sino la propia salud.

La amplia experiencia de que se dispone con los sistemas RFID nos facilitará la tarea de evaluar la forma en que se intentaron resolver los problemas de autenticación y cifrado, y sobre todo, de hasta qué punto se consiguió el éxito. La cuestión no es fácil. Las soluciones criptográficas habituales existen, por supuesto, pero no son sencillas en este entorno. Los dispositivos RFID son pequeños y sencillos, con una capacidad de almacenamiento de datos muy escasa. En la mayoría de los casos, apenas pueden hacer otra cosa que transmitir su número de serie cuando se les interroga desde el exterior. Incluso los modelos más sofisticados tienen una capacidad de cómputo muy restringida por motivos de espacio y energía. Por dichos motivos, los fabricantes de RFID han utilizado sistemas criptográficos hechos a medida; y a veces, mal cortados.

Uno de los fabricantes de tarjetas RFID es la empresa NXP Semiconductors, una antigua filial de Philips vendida en 2006. Según su propia publicidad, han fabricado más de 4500 millones de tarjetas con tecnología RFID incorporada. Uno de sus productos estrella es la línea de tarjetas Mifare, algunas de las cuales utilizan algoritmos de cifra propietarios. El uso de cifrado hecho a medida es, hasta cierto punto, una buena idea en este tipo de tarjetas, debido a las limitaciones que tiene. El inconveniente estriba en el hecho, contrastado por la experiencia en otros muchos campos, de que la criptografía de invención propia raramente da buenos productos.

Para demostrarlo, un doctorando de la Universidad de Virginia llamado Karsten Nohl se puso manos a la obra. Sus resultados, obtenidos en colaboración con el investigador alemán Henryk Plötz, fueron presentados en diciembre de 2007 en el *Chaos Communications Congress*, una famosa reunión de hackers alemanes[13]. El proceso de hackeo de la tarjeta les resultó lento y laborioso, pero una vez hecho afirmaron que en el futuro bastaría con un portátil, un escáner y unos pocos minutos para conseguir la clave criptográfica de una tarjeta Mifare Classic y duplicarla[14].

Vamos a examinar el proceso. Lo primero que hicieron Nohl y Plötz fue tomar una tarjeta e intentar deducir su funcionamiento. Por medio de un microscopio electrónico, obtuvieron la estructura básica del chip: un conjunto de miles de bloques (cada uno de los cuales podía ser una puerta lógica AND o una OR), distribuidos en varias capas. Por fortuna para los investigadores, los bloques correspondían a tan sólo setenta tipos de puertas lógicas distintas. A continuación, buscaron las zonas

criptográficamente importantes del chip. Suena más fácil de lo que fue en realidad, pero al final consiguieron reconstruir el algoritmo de cifrado, un sistema propietario que la empresa denomina Crypto-1.

El primer paso estaba dado. En palabras de Nohl, "*puesto que la seguridad de las tarjetas Mifare se basa en parte en mantener el algoritmo en secreto, creemos que las tarjetas son demasiado débiles para cualquier aplicación de seguridad puesto que el algoritmo puede hallarse sin mucho esfuerzo*". Crypto-1 resultó ser un sistema de cifrado en flujo (*stream cipher*) basado en un registro LFSR que se activa con una clave de 48 bits. El lector tiene una descripción detallada de los registros LFSR en el tema "Descifrando a Nemo," así que no me repetiré aquí.

Un algoritmo hecho en casa, por muy bueno que sea el equipo local, es campo abonado para los fallos. Lo hemos visto hasta la saciedad en este libro, y aquí tenemos otro ejemplo. Resultó que el generador de números aleatorios de la tarjeta, necesario para establecer una sesión de autenticación con el exterior, está mal diseñado y dista mucho de ser aleatorio, lo que significa que el proceso de autenticación puede ser falsificado. Mal comenzamos.

En cuanto al cifrado, una clave de tan sólo 48 bits no parece mucha protección, y puede resolverse mediante métodos de fuerza bruta, esto es, probar todas las claves posibles. Ni siquiera eso es necesario, ya que el algoritmo de cifrado tiene fallos. Para obtener un flujo de clave pseudoaleatorio (con el que luego se podrá cifrar la información), el registro LFSR controla un sistema denominado función de filtrado. Pues bien, el flujo de bits que proporciona esa función de filtrado no es aleatorio. Eso permitió a Nohl y Plötz diseñar una ataque: enviando a la tarjeta patrones de bits cuidadosamente seleccionados y examinando las respuestas de ésta, se pueden obtener más de la mitad de los bits de la clave secreta. La combinación de ambos factores permitió demostrar que bastaban unos pocos minutos para hackear la tarjeta.

La respuesta de la empresa fabricante no sorprendió, porque ya la hemos viso en otros casos. Básicamente vinieron a decir: a) las tarjetas Mifare Classic son un producto de baja gama, y no fueron diseñadas para entornos de alta seguridad como pasaportes o sistemas bancarios; b) sólo atacaron una de las muchas capas de seguridad que hay en el sistema; c) un ataque sería muy costoso y llevaría horas atacar una sola tarjeta[15].

Es cierto que una tarjeta contiene un equivalente económico escaso, pero si se cuenta el coste del sistema en su conjunto, los números marean. En aquel tiempo, Holanda había invertido ya entre mil y dos mil millones de euros en el sistema de tarjetas RFID (*OV-chipkaart*) para su red de transporte público, y tuvo mucho interés en que Nohl y Plötz comparecieran ante su Parlamento para dar testimonio acerca de su descubrimiento. La tarjetas Mifare Classic también se utilizaban en el transporte público de otras ciudades como Boston y Londres.

Dos investigadores de la Universidad de Radboud en Nimega (Holanda) convirtieron la ruptura teórica de Nohl y Plötz en un ataque real. En junio tomaron una tarjeta Oyster, de las que se usan en el metro de Londres (y que está basada en la Mifare), la hackearon y pudieron viajar a placer por el "Tubo" gratis. Los holandeses se tomaron el problema muy más en serio, y el Secretario de Estado, Tineke Huizinga, rogó a las autoridades universitarias que no publicasen los resultados, algo a lo que la universidad se negó. La empresa fabricante, NXP no estaba para sutilezas y acudió a los tribunales. Sin embargo, el juez holandés que vio el caso decidió que la libertad de expresión primaba en este caso sobre los intereses comerciales, y autorizó la diseminación de los resultados[16] [17] [18] [19].

Una semana después del fallo judicial, como si fuese un caso de justicia poética casual, el sistema de tarjetas Oyster del metro de Londres se vino abajo por "problemas técnicos" y los londinenses disfrutaron de un día de metro gratis. Como resultado de todo ello, dos semanas después el alcalde de Londres anunció la terminación del contrato con Transys, el grupo de empresas que proporcionaba las tarjetas Oyster. El motivo aducido era que así se ahorraría dinero (el sistema Oyster costaba, tan sólo en mantenimiento, 100 millones de libras esterlinas al año). También el gobierno holandés se pensó dos veces antes de seguir adelante con sus planes. Finalmente, el despliegue del sistema de tarjetas OV-chipkaart continuó; sospecho que, sencillamente, no tenían alternativa después de haberse gastado una suma tan enorme. En octubre de 2011, se anunció la sustitución de las tarjetas Mifare Classic de NXP por un sistema diferente de otra empresa.

La debacle de las tarjetas de NXP cruzó el Atlántico. La *Massachusetts Bay Transit Authority* (MBTA), encargada de gestionar el transporte público en la zona de Boston, lanzó su propio ataque legal contra tres estudiantes del Instituto de Tecnología de Massachusetts (MIT) por atreverse a redactar un artículo para la reunión sobre seguridad DEFCON, que se celebraría en agosto de 2008. La víctima esta vez era la tarjeta CharlieCard, cuyo sistema RFID utiliza la tecnología de (sorpresa, sorpresa) Mifare Classic. Aunque la MBTA consiguió impedir la presentación de resultados en DEFCON gracias a una orden restrictiva, dicha orden no fue prorrogada, y ni que decir tiene que ahora los datos están libremente disponibles en Internet[20]. Por cierto, que ese mismo mes casi 5000 expertos se reunieron en Las Vegas para una de las más famosas reuniones sobre seguridad, la Black Hat. Todos esos asistentes llevaban una placa identificativa que contenía una tarjeta RFID. ¿Saben qué sistema usaba dicha tarjeta? ¡La Mifare Classic! En casa del herrero…[21]

Últimamente se está desarrollando una tecnología para micropagos llamada Comunicaciones de Campo Cercano o NFC (*Near Field Communications*). Viene a ser una especie de RFID con dos diferencias fundamentales: tiene un alcance menor

(pocos centímetros), y un nombre distinto. La primera propiedad le asegura mejor protección frente a fisgones no autorizados, y en cuanto a la segunda le permite dejar atrás la mala publicidad ligada a los sistemas RFID que acabamos de ver.

Por desgracia, no deja de tener sus fallos de seguridad. En octubre de 2012, dos investigadores especializados en seguridad de móviles hicieron un descubrimiento fortuito. Las tarjetas de autobús de ciudades como San Francisco son Mifare Ultralight y están basadas en NFC; quizá pudieran ser leídas por un Google Nexus S, un móvil que también incorporaba esa tecnología. Lo intentaron y lo consiguieron. Lo interesante es que, según comprobaron, los bits que indicaban cuántos viajes quedaban en la tarjeta no solamente se podían ver, sino también modificar. No había más que crear una aplicación para Android que efectuase esa modificación, y de repente tenemos el equivalente a una tarjeta con viajes infinitos[22]; y eso a pesar de que una salvaguardia de seguridad de las tarjetas debería haber impedido esa modificación.

Eso es precisamente lo que hicieron los autores, pero no espere encontrar la aplicación (llamada *UltraReset*) en la tienda online. Sí puede usted descargarse *UltraCardTester*, de los mismos autores, similar a la anterior pero que no permite re-escribir la tarjeta[23]. Los autores no querían problemas legales, pero como dijo uno de ellos: "*no soy un desarrollador de software, así que si alguien sabe lo que hace es muy fácil hacerlo*"[24]. El problema se hace especialmente grave por la facilidad del ataque. Nada de ordenadores ni lectores de tarjetas, tan sólo un móvil dotado con tecnología NFC y una aplicación software.

La pobre NXP, que vio cómo la historia de la Mifare Classic se repetía, no pudo menos que reconocer la validez del ataque, al tiempo que ofrecía como solución la Mifare Ultralight C, que incorpora seguridad criptográfica[25]. Yo no sé cuál es el tipo de tarjeta que utiliza el servicio de autobuses urbanos de mi ciudad, pero sí les puedo decir una cosa: antes de cambiar a las tarjetas RFID, nunca vi un solo revisor en un autobús, y ahora es raro el trimestre que no me encuentro con uno.

Si les parece bien, volvamos al hospital, que tenemos allí un problema esperándonos. Como hemos visto, la tecnología RFID parece prometedora pero tampoco es la solución mágica. Eso no significa que no podamos aplicarla a dispositivos médicos, sino que la seguridad y la privacidad han de ser reforzadas. Pero incluso si se resuelven los problemas del cifrado y la autenticación, seguimos teniendo serias restricciones. Cruzarse de brazos y tirar la toalla no es una opción, y por ello un equipo de investigadores de las universidades de Washington, Massachusetts-Amherst y de la Escuela Médica de Harvard realizó en 2008 un estudio sobre un desfibrilador automático implantable (DAI) comercial que se comunica en forma inalámbrica con el exterior[26].

En la primera parte del estudio, demostraron que es posible, usando osciloscopios

y equipo técnico diverso, interrogar por vía inalámbrica al DAI y, mediante técnicas de ingeniería inversa, determinar cómo funciona. Algunos resultados son preocupantes. Por ejemplo, se supone que el DAI que utilizaron solamente transmitía datos al acercar un imán exterior para cerrar un interruptor magnético. Sin embargo, el equipo consiguió activar la transmisión telemétrica sin imán, utilizando solamente una orden de radiofrecuencia.

El trabajo muestra que el DAI cotilleaba más que una portera. La información que transmitía en claro (sin cifrar) incluía el nombre del paciente, su fecha de nacimiento, número identificador médico e historia clínica. Con los datos extraídos, fue un juego de niños deducir datos como el nombre y número de teléfono del médico, la fecha de implantación del DAI, modelo y número de serie. Las medidas de telemetría revelaron el ritmo cardíaco. Y menos mal que los autores no se lanzaron a un estudio a fondo de los protocolos de comunicación. Quién sabe que habría revelado un estudio de ingeniería inversa completo.

A continuación, se dedicaron a jugar a los ataques, escogiendo los que requerían poca complicación y violaban la confidencialidad o integridad de los datos. De nuevo, los resultados asustan. No solamente consiguieron sonsacar al DAI los datos mencionados en el párrafo anterior, sino que llegaron a cambiar algunos de los datos que almacenaba, como el nombre del paciente o el reloj interno. También consiguieron interferir con las respuestas a eventos peligrosos, de modo que no se activase el "chispazo" de 700 voltios cuando lo necesitase. Y a la inversa, consiguieron saltarse las salvaguardias contra chispazos accidentales y hacer que el DAI indujese una fibrilación ventricular no autorizada. También reprodujeron con éxito un ataque de denegación de servicio, haciendo que el DAI transmitiese datos continuamente.

Los investigadores eran conscientes de que las soluciones criptográficas, como hemos visto aquí, son onerosas en términos de energía, y pueden provocar problemas en casos de emergencia. Por ese motivo, ofrecieron lo que ellos llaman **soluciones de potencia cero**, que aumentan la seguridad del sistema sin requerir energía extra.

El primer paso consiste en un **sistema de alarma**, en la forma de un sensor piezoeléctrico que cuando recibe una señal de radiofrecuencia emite un pitido de aviso (que puede sustituirse por una vibración). Dicho sensor está unido a un chip RFID especial llamado WISP (*Wireless Identification and Sensing Platform*) que extrae energía de una señal de radiofrecuencia externa, de modo que no requiere energía extra del dispositivo. Para comprobar la eficacia del sensor, los autores hicieron una prueba "en vivo," aunque el término no es literal, por supuesto: las pruebas se hicieron con un DAI comercial insertado dentro de una bolsa llena de bacon y carne de ternera.

La segunda línea de defensa es un **esquema de autenticación**, y aquí entra la

criptografía. Partimos de una clave maestra *Km* almacenada en una forma segura dentro de los dispositivos que puedan interactuar con un dispositivo implantado. A estos dispositivos se les suele llamar programadores (no confundir con los programadores humanos). Por su parte, cada DAI tendrá un número de serie identificativo *I* y una clave propia *K* calculada como función de ambos factores: *K = f(Km,I)*. En principio, *f* será una función criptográficamente fuerte y pseudoaleatoria (los autores proponen el algoritmo de cifra simétrica AES).

Para iniciar la comunicación, el programador le envía al DAI una petición de autenticación, una especie de “hola, responda”. Es habitual en los protocolos de autenticación intercambiar un número aleatorio para evitar que un atacante pueda usar una comunicación grabada con anterioridad. El DAI responde al (de momento, presunto) programador enviando dos datos: su identificador *I* y un número *N* generado de forma aleatoria. El programador calcula *K=f(Km,I)* para obtener la clave del DAI, y a continuación utiliza un algoritmo de cifra en bloque (los autores proponen el algoritmo RC5) usando *K* como clave y *N* como mensaje. El resultado *R = RC5(K,N)* se envía al DAI. Éste, por su parte, calcula el resultado de usar RC5 usando *K* como clave y *N* como mensaje, es decir, *R’ = RC5(K,N)*. El intercambio de datos vendría dado de esta forma:

```
Programador  IDC
Hola -------------------->
       <----- (I , N) ----
f(Km,I) = K
RC5(K,N) = R
     ----- (R) ---------->
              RC5(K,N) = R’
              ¿R = R’?
```

Este esquema puede parecer complicado, pero cumple bien su cometido. En primer lugar, durante el proceso no hemos intercambiado ningún dato sensible. Un atacante tan sólo podrá captar *I* (un número de identificación que no es secreto), *N* (un número aleatorio sin ninguna particularidad especial) y *R* (resultado de cifrar *N* con una clave desconocida). Ni *K* ni *Km* han abandonado en ningún momento sus contenedores.

Por otro lado, la respuesta del programador indica al DAI si es de fiar o no. Imaginemos por un momento que un interlocutor desconocido intenta ligar con el DAI, y, tras el intercambio de datos, R y R’ no coinciden. Si eso sucede, puede ser por uno de los siguientes motivos.

—El interlocutor no conoce la clave maestra *Km*, lo que significa que no está autorizado para intercambiar información.

—La respuesta *R* del interlocutor está basada en un número aleatorio *N’* distinto

de *N*, lo que indica que está reproduciendo un intercambio de datos grabado en el pasado.

—La respuesta está basada en un identificador *I'* distinto de *I*, esto es, está intentando colar la conversación que tuvo con otro DAI.

Es decir, sólo si *R'* = *R* podemos estar seguros de que el interlocutor conoce la clave secreta (*Km*) y ha respondido a la petición de identificación hecha en este momento y lugar.

En todo el proceso solamente se han utilizado algoritmos de cifra simétrica, que son más rápidos y eficientes que los de clave pública. Para simplificar el proceso de prueba, el equipo probó con un valor de *N* constante, lo que reduce la seguridad pero simplifica el proceso, y consiguieron ejecutar con éxito el algoritmo RC5 en el chip WISP, que recordemos toma la energía del exterior. Su conclusión es que este esquema funciona de forma adecuada para proteger al menos partes vitales de la información, aunque la pregunta de si puede usarse para cifrar todo el flujo de datos permanece sin respuesta todavía. Tampoco resuelve el problema de la gestión de claves (cómo crear, administrar y diseminar todas las *Km*, almacenarlas en lugar seguro y revocarlas si es preciso) en entornos con grandes cantidades de claves revoloteando por ahí, si bien hay que tener en cuenta que ese no era el objeto del estudio.

Finalmente, la tercera solución de potencia cero consiste en **utilizar un canal distinto** para transmitir la clave criptográfica. En lugar de enviar la clave por radiofrecuencia, se transmitiría mediante un canal de comunicación basado en ondas acústicas o ultrasónicas. El aparato programador, situado en el exterior, tendría un micrófono que solamente podría captar la información en el caso de que estuviese físicamente en contacto con la piel del paciente; de otro modo, la señal sería demasiado débil para ser captada.

Es decir, la triple solución de potencia cero se basa en a) avisar de ataques no autorizados, b) usar criptografía, al menos en forma simplificada, y c) decir la clave al exterior en voz muy baja.

Hay otras soluciones en marcha. Un grupo del MIT y de la Universidad de Massachusetts-Amherst propuso en 2011 utilizar un "escudo" que protegería un dispositivo médico implantado sin necesidad de modificarlo, evitando la decodificación de mensajes legítimos al tiempo que impide los intentos de comunicación no autorizados. El escudo emite una señal de interferencia que impide cualquier comunicación directa con el dispositivo. Solamente los datos que pasen por el escudo recibirán una "señal antídoto" que los descifrará[27].

La ventaja de este sistema está en que la seguridad del sistema no precisa de energía que tenga que ser drenada del dispositivo. El escudo es externo al cuerpo, puede llevarse colgado como un pequeño collar y en principio puede utilizarse en

cualquier tipo de dispositivo implantado. Es una forma elegante de “externalizar” el problema de la seguridad a un aparato físicamente desvinculado del dispositivo implantado. Ya ha sido probado con éxito en un DAI “en vivo” (entiéndase “en un paquete de bacon y carne de ternera”).

Otros investigadores han abogado por el uso de rasgos biométricos. En 2009, un grupo del Darmouth College en Hanover (EEUU) propuso utilizar patrones característicos del latido de un corazón con fines de autenticación. Los rasgos de un electrocardiograma, combinados con datos de acelerómetros, servirían como identificadores biométricos. Esta idea no incorpora solución de cifrado pero permite verificar la identidad del paciente a distancia, y ha sido planteada con fines de monitorización a distancia[28].

En el caso de que se adopte una solución criptográfica basada en una contraseña, es esencial garantizar un acceso a dos niveles. Habitualmente, el paciente la tendría bajo su control, pero en el caso de que perdiese el control (por estar inconsciente o gravemente herido), el equipo médico de emergencia debería poder acceder a ella. Un estudio reciente examinó diversas posibilidades, desde el punto de vista tanto de sus características técnicas como de su aceptación social[29].

El método más sencillo consiste en grabar la contraseña en un brazalete o chapa de alerta médica. Proporciona una seguridad aceptable en situaciones normales, y tiene la ventaja de que muchos pacientes ya están familiarizados con el concepto del brazalete médico. Como desventaja, resultan poco discretos. A algunos pacientes no les gustó que el brazalete fuese mostrando a todo el mundo que tiene una enfermedad. Para uno de ellos, el mero hecho de llevar un objeto que le recordaba constantemente su condición de paciente con marcapasos le resultaba molesto: “*me sentía como un inválido*”.

Como alternativa al brazalete, el paciente puede llevar una pulsera con un dispositivo llamado “ocultador,” que funciona de firma similar al “escudo” anteriormente mencionado pero con una diferencia: cuando se retira el ocultador, el sistema cambia sus reglas de acceso y autoriza la comunicación con cualquier aparato programador externo. En este caso, la seguridad estriba en el control físico que hace la persona. Los entrevistados expresaron su satisfacción por los modelos que incorporan opciones mejoradas (como los ocultadores que dan la alarma al 112 automáticamente cuando detectan una emergencia cardíaca), pero no mostraron simpatías hacia los modelos estándar por motivos similares a los de los brazaletes: ir expresando públicamente la propia debilidad mediante una pulsera no resulta agradable. Otros problemas incluían las molestias derivadas de tener que llevarlo a todas horas, recordar quitárselo en la ducha, temores a que se enganche con otros objetos, etc.

La tercera opción es un tatuaje. Sí, un tatuaje. La contraseña del sistema se

codifica en forma de un código de barras bidimensional QR y se tatúa en el paciente, preferentemente en una zona donde no sea habitual sufrir un accidente o quemadura. A pesar de su utilidad, la idea del tatuaje resultó ser la menos atractiva para los pacientes. Los motivos del rechazo fueron fundamentalmente de carácter cultural. A un participante le disgustaban los tatuajes porque los asociaba con borrachos; otra decía que no le gustaba la imagen que de ella proyectaba cara al público. Una persona judía fue brutalmente sincera en su comentario: "*para mí, un tatuaje en el brazo me recuerda a un campo de concentración*".

Los propios autores del estudio reconocen haber tenido "*reticencias acerca de incluir identificadores basados en tatuajes, especialmente considerando una posible asociación con el tatuado de prisioneros en campos de concentración durante la Segunda Guerra Mundial… teníamos la hipótesis de que ese sistema, si bien satisfactorio desde el punto de vista técnico, no sería satisfactorio desde el punto de vista del paciente*"".

Sin embargo, tatuar información de acceso (una palabra o frase de contraseña) es esencialmente una buena idea, y de hecho ninguno de los participantes en el estudio rechazó la alternativa por insegura. Por ello, los médicos ofrecieron otra opción, ya apuntada ese mismo año por Stuart Schechter, de Microsoft[30]. Consiste en utilizar un tatuaje con tinta ultravioleta. En condiciones normales es invisible al ojo humano, pero resulta perfectamente legible ante un lector dotado de luz ultravioleta.

El sistema de tatuaje también tiene sus inconvenientes. Algunos pacientes pueden sufrir irritaciones en la piel, o quizá algún riesgo a largo plazo para la salud. También son susceptibles a daños físicos, como quemaduras o amputaciones. Incluso manchas de grasa o sangre pueden entorpecer la lectura de la clave. Quizá fuese conveniente hacer copias del tatuaje en varias partes del cuerpo. Si los médicos tienen la sensibilidad suficiente para poder hacerlos sin que recuerden a los tatuajes de los campos de exterminio nazi (nada de ponerlos en el brazo, por ejemplo), pueden ser una alternativa válida no sólo desde el punto de vista de la seguridad, sino de la aceptación por parte de los pacientes.

En diversas partes de este libro he añadido una sección de consejos para aprender cómo protegernos de los diversos peligros criptográficos que nos amenazan. El caso de los dispositivos insertados en el cuerpo es tan novedoso que, como habrán visto, las soluciones de seguridad todavía se están desarrollando. En ese sentido, ni puedo recomendarles prácticas seguras a seguir ni puedo guiarles al respecto.

Pero sí puedo, y debo, dejarle algo claro a más allá de toda duda razonable: renunciar a dichos dispositivos NO es una opción. Por favor, no se asuste con los fallos técnicos que le he descrito aquí. Hasta la fecha, no se han detectado ataques contra marcapasos o desfibriladores implantados en un cuerpo humano. Si bien los problemas que hemos mostrado aquí son factibles tanto de forma teórica como

práctica, aún no se han dado en el mundo real.

En ningún caso un paciente debería plantearse dejar de utilizarlos por ese motivo. Los firmantes de uno de los estudios mencionados anteriormente lo afirmaron de forma tajante:

Creemos firmemente que nada en nuestro informe debe disuadir a los pacientes que necesiten estos dispositivos si se los recomienda su médico. El desfibrilador cardíaco implantable es una técnica probada que salva vidas. Creemos que el riesgo a los pacientes es bajo y que los pacientes no deberían alarmarse.

No puedo estar más de acuerdo con esa afirmación. Si su médico le recomienda un dispositivo desfibrilador, una bomba de insulina, un marcapasos o aparato similar, siga su consejo profesional. Son técnicas probadas que salvan vidas.

DESCIFRANDO A NEMO

Dejando aparte las fuerzas armadas y los servicios de inteligencia, la industria audiovisual es posiblemente la mayor usuaria de productos criptográficos del mundo. Año tras año, han desarrollado y desplegado protocolos de cifrado de todo tipo, destinados a controlar el uso y distribución de películas y discos musicales.

No siempre fue así. Antiguamente, las canciones y las películas se grababan y vendían en formato analógico: LPs, cassettes, cintas de vídeo. Cuando comenzaron a proliferar los dispositivos copiadores (magnetoscopios, cassettes de doble pletina), la industria se alarmó y exigió una regulación, llegando al extremo de intentar prohibir la tecnología del video doméstico.

Como la calidad de las nuevas copias era inferior a la del producto original, pronto se alcanzó un nuevo equilibrio. Los usuarios recibían permiso (implícito o explícito) para hacer una copia de menor calidad, lo que se llamaba “supuesto de bagatela,” en tanto que las copias múltiples con fines de reventa quedaban prohibidas. Por su parte, los vendedores recibían una compensación por las copias no vendidas. En España, dicha compensación fue introducida tras la aprobación de la Ley 22/1987. En la actualidad, el canon digital está en proceso de cambio.

Con la proliferación de los medios digitales y el desarrollo de Internet, la situación volvió a cambiar. Los usuarios ya podían realizar copias perfectas, indistinguibles del original, así como diseminarlas masivamente por todo el mundo en cuestión de segundos. Ya no sería necesario conformarse con una copia de menor calidad, ni esperar meses a que un disco publicado en Estados Unidos fuese lanzado en España. Ni siquiera hay que tener los conocimientos técnicos para realizar la copia, ya que basta que una sola persona sepa cómo hacerlo para que el resto de Internet aprenda la lección.

La industria audiovisual, por su parte, vio la ocasión para instalar nuevos sistemas de control y protección en sus productos. Tales sistemas, denominados genéricamente DRM (*Digital Rights Management*, Gestión Digital de Derechos) les permitiría no solamente restringir la copia indiscriminada, sino afinar en el control de sus productos. Se podría cobrar un producto en función del número de reproducciones; controlar cuántas veces al día puede reproducirse una película determinada; determinar lo que el usuario podría o no hacer. En la actualidad, hay operaciones prohibidas para el usuario, como saltarse los anuncios de inicio o el aviso del FBI de que piratear es ilegal, inmoral y engorda.

Durante años, la lucha entre los partidarios de los DRM y sus detractores se ha plasmado en la arena criptográfica: los primeros desarrollan todo tipo de sistemas de cifrado y firma digital para proteger sus productos, y los segundos utilizan técnicas similares para romper esas protecciones. Como resultado, hemos sido testigos de una

larga y apasionante lucha entre el escudo y la espada, entre la protección criptográfica y el ataque criptoanalítico. A lo largo de estas líneas, examinaremos algunos de los sistemas utilizados por la industria de medios audiovisuales, y comprobaremos hasta qué punto han logrado el éxito.

1) NO TOQUE ESA TECLA

Cuando la industria musical descubrió que un CD puede almacenar música de alta calidad, era también consciente de que resultaba muy fácil de copiar. El ordenador personal no solamente era un lugar lógico para efectuar copias, sino que los usuarios utilizaban cada vez más el ordenador para reproducir su propia música. El reto consistía en permitir la reproducción de discos originales y negar al usuario cualquier otra posibilidad.

Para ello, es necesario controlar el entorno en que se reproducen, es decir, insertar protecciones de tipo DRM en el propio sistema operativo. Eso es lo que intentó Microsoft en 2006 con Windows Vista, que básicamente es una nueva versión diseñada para, entre otras cosas, incorporar masivamente sistemas DRM a petición (digámoslo así) de Hollywood. Los resultados fueron desastrosos. El uso masivo de tecnología de protección se cobró un alto precio en prestaciones, y los usuarios le dieron la espalda. La revista *PC World* calificó Windows Vista como "*la mayor decepción de 2007 en el mundo de la técnica*"[1]. Cuando yo consideré adquirir el ordenador con el escribo estas líneas, el fabricante me ofreció dos posibilidades: usar Windows Vista o utilizar un disco de "desactualización" (*downgrade*) a Windows XP. ¡Y eso era considerado un regalo por parte del vendedor! A tenor de lo visto, ciertamente lo era. Por cierto, finalmente mi ordenador vino con Windows 7 de serie, así que nunca he tenido una versión de Vista en mis manos.

Pero Windows Vista fue anunciado en 2005, y los vendedores de CD no estaban dispuestos a sentarse y esperar que Microsoft resolviese sus problemas. Diversas empresas ensayaron soluciones dispares. Una de las más desafortunadas fue la perpetrada por la empresa SunnComm, que desarrolló software de protección para el gigante musical BMG (Bertelsmann Music Group). En septiembre de 2003, se anunció el lanzamiento del primer álbum musical protegido por el sistema MediaMax CD3 de SunnComm. Cada canción del disco tenía dos versiones; la primera sería reproducible en reproductores CD estándar pero no podría ser copiada en un ordenador, y la segunda podría reproducirse en un reproductor Windows Media dotado con protecciones DRM.

¿Cómo puede conseguirse una reproducción en Windows al tiempo que se impide la copia? La única solución viable consistía en que el CD instalase un paquete de software en el ordenador, y eso es lo que hacía. Para ello, el sistema MediaMax CD3 aprovechaba una opción de Windows llamada *autorun*, (ejecución automática). Cuando dicha opción está activada, cualquier disco que insertemos en la lectora de CD/DVD se leerá y ejecutará automáticamente. Esta opción es muy útil a la hora de, por ejemplo, instalar software o visualizar una película.

Cuando la opción *autorun* está activada, MediaMax CD3 ejecuta un archivo

llamado LaunchCD.exe, que proporciona el acceso a las canciones protegidas por DRM; al mismo tiempo, ejecuta un *driver* (controlador) para que se almacene en la memoria del ordenador. Si el usuario acepta las condiciones del contrato de licencia, el controlador permanecerá cargado en memoria para siempre, incluso si el ordenador es apagado y encendido de nuevo. La misión del controlador es "controlar" los contenidos de cada CD insertado en la unidad lectora, permitiendo su reproducción pero impidiendo su copia.

Resulta sorprendente que SunnComm pretendiese vender esta solución como segura y sólida. Y digo esto porque las contramedidas son tan sencillas que rozan lo absurdo. Voy a mencionarlas en orden creciente de sencillez. En primer lugar, el controlador puede ser deshabilitado fácilmente. El problema es que vuelve a instalarse en memoria, de modo que su erradicación permanente es más difícil.

Afortunadamente, hay una forma más sencilla de evitar la protección de MediaMax CD3, y es tan sencillo como desactivar la opción *autorun* del sistema operativo. Tras hacerlo, insertar el CD de música no activará el programa LaunchCD.exe, a menos que el usuario lo ejecute deliberadamente. De ese modo, se podrá copiar todo el contenido del disco.

Pero hay una tercera vía. La descubrió en octubre de 2003 J. Alex Halderman mientras trabajaba como profesor en el departamento de informática de la Universidad de Princeton. Podríamos pensar que este investigador de currículum impecable, trabajando en una de las mejores universidades del mundo, desplegó todo su saber en un ataque criptoanalítico brillante. Nada más lejos de la realidad. La receta de Halderman para saltarse el sistema MediaMax CD3 consiste en los siguiente pasos:

—Pulsar la tecla Mayúsculas (sí, esa que tiene una flechita hacia arriba y permite escribir *A* en lugar de *a*)

—Insertar el CD.

—Esperar unos segundos.

—Soltar la tecla Mayúsculas.

Eso es todo. El propio autor realizó una prueba empírica adicional sobre la robustez de MediaMax CD3. Razonó que, si el sistema de protección es bueno, pasaría bastante tiempo hasta que los discos que protege se filtrasen a las redes de intercambio. Para ello se fijó en un álbum en particular (*Coming from Where I'm From*, de Anthony Hamilton). Halderman lo encontró en Kazaa el día 27, apenas cuatro días después de ponerse a la venta[2].

Tras la divulgación del "truco de las mayúsculas"[3], la reacción de la empresa SunnComm fue tan decepcionante como predecible: amenazó al autor de la revelación con una demanda judicial. SunnComm afirmó que Halderman violó la DMCA al revelar el nombre del programa instalado y al deshabilitar el driver.

Alegando que los daños a su reputación provocaron una caída en su cotización en bolsa del 30%, la empresa reclamó al investigador la friolera de diez millones de dólares en concepto de indemnización[4].

Sin embargo, el daño estaba hecho. No fueron las acciones de Halderman las que provocaron la bajada en bolsa de las acciones de SunnComm, sino el hecho de que la técnica de MediaMax CD3 no cumplía las promesas de seguridad hechas en el pasado. Si algo podía dañar más aún la reputación de la empresa, sería un empeño en demandar a un investigador por decir a la gente que pulse una tecla del ordenador. La EFF (Electronic Frontier Foundation) se ofreció a defender a Halderman, y finalmente, prevaleció la lógica: SunnComm rectificó y retiró la amenaza de una demanda judicial[5].

El propio presidente de SunnComm, Peter Jacobs, comentó que "*son diez millones de pavos [perdidos], pero tal vez podamos reponernos*"[6]. En efecto, la empresa plegó velas y se justificó con el argumento de que su sistema de protección tan sólo se había diseñado "*para dar a los amantes de música honrados una oportunidad para hacer copias de uso personal, no para detener la piratería a gran escala*"[7]. Eso les llevó a otro escándalo aún mayor, con repercusiones serias para todos los implicados.

En 2004, BMG se fusionó con Sony Music Entertainment para formar Sony BMG. Cada una de ellas aportó su propio sistema de protección de información: MediaMax para BMG, y un sistema llamado XCP (*Extended Copy Protection*) para Sony. XCP había sido desarrollado por la empresa First4Internet, y se trataba de un paso más en la pugna por controlar el ordenador del usuario.

Como ya hemos visto, si se instala un sistema de protección que pueda desactivarse, resulta inútil. ¿Cómo evitarlo? First4Internet abogó por eliminar al usuario de la ecuación. Para ello preparó un paquete de software que no pudiera ser detectado ni desinstalado. XCP es lo que en jerga informática se llama un *rootkit*, un programa que toma el control del ordenador, atribuyéndose privilegios ilimitados al tiempo que permanece oculto al propio ordenador. El acuerdo de usuario final ni siquiera mencionaba su existencia.

Cualquier informático les dirá que un rootkit es una herramienta muy peligrosa. Al proporcionar acceso total y ocultación, es el arma perfecta para hackear un ordenador, robar sus datos, instalar virus o troyanos, cualquier cosa. Es el equivalente de tener a un espía del CNI como director de la CIA. Que una empresa decida utilizarlos de forma furtiva sin solicitar siquiera el permiso de los usuarios es una práctica cuando menos cuestionable, peligrosa y al borde de lo ilegal. Tales dilemas éticos no influyeron en las decisiones comerciales de Sony BMG, y en marzo de 2005 comenzó a vender CDs de música protegidos con el rootkit XCP.

Aunque entonces no se conocía el motivo, ahora se sabe que desde al menos

agosto de 2005 algunos usuarios tuvieron problemas de estabilidad causados por XCP: el sistema se colgaba y tenía que ser reiniciado. El fallo se debía a un archivo llamado *aries.sys*, que más tarde se descubrió formaba parte del rootkit.

El escándalo saltó el último día de octubre. Mark Russinovich, de Sysinternals, reveló la existencia del rootkit, que descubrió accidentalmente mientras probaba un programa para detectar rootkits. Al intentar eliminarlo, descubrió que los archivos necesarios para utilizar el lector de CD quedaban inutilizados[8]. La noticia se diseminó con gran velocidad [9].

La respuesta de Sony fue rápida, y en apenas dos días anunció la presentación de un parche que supuestamente eliminaba la capa de secreto. Aprovecharon para anunciar que "*[XCP] no es malicioso y no compromete la seguridad… esta actualización permitirá a los usuarios eliminar este componente de sus ordenadores*"[10]. Sin embargo, ni pidieron disculpas ni reconocieron lo peligroso de su actividad (por no decir ilegal). En declaraciones a una cadena de radio norteamericana el día 4 de noviembre, el propio presidente de la división de negocios digitales globales de Sony se enfrentó a Russinovich: "*La mayoría de la gente ni siquiera sabe lo que es un rootkit, así que ¿por qué debería preocuparles?*"[11]. Parecía como si el único pecadillo cometido hubiera sido dejarse coger con las manos en la masa.

Por supuesto, la gente que sabía del asunto estaba preocupada, no sólo por la peligrosidad del rootkit en sí, sino por la desfachatez con que Sony trataba todo el asunto. Ed Felten, en un excelente artículo publicado el día 3 en su blog, afirmó que la "actualización" que Sony hacía pasar por solución al problema no era sino más de lo mismo: incluía casi todos los archivos del rootkit original, y algunos nuevos[12].

Lo que resulta más aterrador es que las mismas características que hacían tan valioso el rootkit XCP para Sony BMG podría permitir a terceros el acceso no autorizado a cualquier ordenador. Mientras Felten descargaba su furia contra Sony, la empresa responsable del juego online *World of Warcraft* reconocía que un hacker podría usar XCP para entrar en su sistema de juego online y hacer trampas sin ser detectado[13].

Los vendedores de software antivirus tomaron cartas en el asunto casi desde el primer día en que el problema fue relevado. A algunos puede parecerle raro que empresas de software antivirus, que se ganan el pan buscando amenazas como rootkits, virus y troyanos, no hubiesen detectado este problema durante los meses en que Sony BMG estuvo vendiendo impunemente sus productos. Hay quien llegó a afirmar que la propia ley DMCA se lo impedía a estas empresas[14].

Sea así o no, la publicidad les obligó a pasar a la acción. El 5 de noviembre, Computer Associates lo calificó como un *spyware* (programa malicioso espía) con categoría de peligro alta[15] y el 8 modificó su programa antispyware *Pest Patrol* para

detectar y eliminar el rootkit[16]; McAffee hizo otro tanto[17]. El 13, Microsoft anunció que el rootkit XCP era una amenaza de seguridad para Windows y que modificaría su herramienta Windows Defender para erradicarlo[18].

El día 10 se descubrió la primera aplicación maliciosa (*malware*): una versión del troyano Breplibot que aprovechaba la tecnología del rootkit de Sony para instalarse sin ser detectado. Ya no se trataba de una vulnerabilidad potencial, sino de un fallo de seguridad probado de consecuencias potencialmente catastróficas. Enfrentado a una tormenta de críticas, Sony BMG claudicó. El 11 de noviembre anunció que suspendía la fabricación de CDs equipados con la tecnología de First4Internet[19]. Tres días después, retiró de la venta todos los CDs protegidos con XCP y anunció un programa para los 2,1 millones de discos ya vendidos, sustituyéndolos por copias nuevas sin rootkit[20].

Pero la pesadilla estaba muy lejos de terminar. Un estudio efectuado por el investigador Dan Kaminski reveló que el rootkit de Sony había infectado más de medio millón de redes. Eso se pudo averiguar porque Kaminski se dio cuenta de que el rootkit, tras instalarse, enviaba a Sony un mensaje. Es decir, XCP no solamente se instalaba en silencio sino que además filtraba información al exterior[21].

Para empeorar las cosas, el procedimiento establecido por Sony BMG para que los usuarios se librasen del rootkit tenía sus propios fallos de seguridad. El procedimiento pasaba por la instalación de una aplicación ActiveX llamada CodeSupport, que posteriormente autorizaría la descarga del programa de desinstalación del rootkit. El problema es que CodeSupport no verificaba de dónde provenía la descarga, lo que significa que en principio cualquier página web podía engañarle y hacer que descargase software malicioso[22].

Cuando parecía que las aguas volvían lentamente a su cauce (excepción hecha de la mala publicidad, las pérdidas económicas y las posibles demandas judiciales), llegó la tormenta perfecta. Con el escándalo del rootkit XCP en portada, algunos foros especializados advirtieron de la existencia de otro sistema de seguridad escondido en los CDs de Sony BMG. Se trataba de nuestro viejo amigo MediaMax, ahora en su versión 5. Sin ser realmente calificable como rootkit, MediaMax 5 compartía características similares: se instalaba sin aviso ni consentimiento, activaba un controlador incluso si el consumidor se había negado a aceptar el contrato de usuario final, no podía ser desinstalado fácilmente y transmitía información subrepticiamente a SunnComm[23].

Aunque el aviso se dio hacia el 11 de noviembre, la atención del público estaba puesta en el rootkit XCP. La propia existencia de MediaMax 5 pasó inadvertida durante un mes, y eso le vino muy bien a Sony BMG porque así al menos un elemento de seguridad podría permanecer oculto en sus discos. Salvo que el 6 de diciembre saltó de nuevo la noticia: MediaMax 5 tenía sus propios fallos de

seguridad. En este caso se trataba de lo que denominaron “ataque de escalada de privilegios,” gracias al cual en el sistema operativo se lleva a cabo un cambio que aparentemente resulta inofensivo pero que permite a un atacante hacer cambios reservados al administrador del sistema. Para entendernos, es como si el MediaMax 5 se hubiese dejado puestas las llaves del coche: cualquiera que pase por allí puede meterse dentro, robar la radio y darse un paseo gratis.

La nueva vulnerabilidad fue descubierta el 29 de noviembre por iSEC Partners[24] y hecha pública el 6 de diciembre, y en esta ocasión Sony BMG respondió con mayor rapidez y eficacia: al día siguiente habían preparado un paquete de actualización. Para su desgracia, la actualización resultó tener fallos de seguridad[25].

Podríamos pensar que Sony BMG recibió su merecido, pero no es así. Las demandas contra la empresa fueron liquidadas de un modo u otro, a base de indemnizaciones y de programas de intercambio de CDs. Queda para el usuario el amargo regusto de la diferencia de trato: lo que hizo Sony BMG hubiera llevado a la cárcel a un particular durante muchos años en Estados Unidos.

Sin embargo, los malos no quedaron del todo sin castigo. El caso de los rootkits puso de manifiesto las malas prácticas de las empresas discográficas a la hora de proteger sus contenidos, y convirtió a Sony en un enemigo declarado de la comunidad hacker. La mala publicidad, combinada con la facilidad con la que se desarrollaron contramedidas (algunas tan sencillas como la de pulsar la tecla de mayúsculas), fueron un factor determinante en el proceso que finalmente acabó con los intentos de las discográficas por controlar el uso de sus CDs. A lo largo de 2007, los principales sellos musicales anunciaron el abandono de los sistemas de gestión de derechos digitales (DRM): EMI en abril[26], Vivendi/Universal en agosto[27] y Warner en diciembre[28]. La última en abandonar el barco fue la propia Sony BMG, en enero de 2008[29].

Aunque los sistemas DRM mencionados aquí no estaban basados en principios criptográficos, tienen una característica común a muchos sistemas de cifrado mal empleados: el secreto mediante oscuridad. Este principio afirma que, para garantizar la seguridad de un sistema, lo mejor es que el atacante no conozca sus detalles. El problema consiste en que el atacante, tarde o temprano, conocerá esos detalles. La gente habla, los espías extraen información, los programas filtran datos. Tarde o temprano, el enemigo sabrá dónde se encuentra la puerta para entrar.

La seguridad mediante oscuridad se ha mostrado inútil una y otra vez a lo largo de la Historia. La seguridad realmente eficaz no consiste en esconder la puerta, sino al contrario, en hacerla visible. Si el enemigo sabe dónde está la puerta, cómo está hecha, qué vulnerabilidades tiene, y a pesar de ello no consigue abrirla, entonces estamos hablando de una puerta segura.

La industria discográfica probó con la seguridad mediante oscuridad, cifrando sus

esperanzas en el secreto: controladores furtivos, programas que se instalan sin aviso previo, rootkits que nadie sabe que existen. Finalmente aprendieron la lección de la forma más dolorosa y tiraron la toalla. Pero sus primos de Hollywood no estaban dispuestos a ceder tan fácilmente. Próximamente veremos cómo los principales estudios de cine desarrollaron y utilizaron sus propios sistemas de gestión de derechos, cometiendo nuevos errores en el proceso.

2) LAS GUERRAS DEL DVD

A finales de los años 70, la tecnología del vídeo doméstico comenzó a erosionar el control que los estudios de cine tenían sobre sus productos. Hasta entonces, la única forma que el usuario tenía de ver una película era ir al cine o esperar a que hubiese un pase por televisión. La popularización de las tecnologías de video (VHS, Beta) hizo que cualquiera pudiese comprar una cinta y verla en su casa cómo y cuando quisiese. Eso vino muy bien a Hollywood, que prontamente se apuntó a este nuevo modelo de negocio. El inconveniente surgió cuando se dieron cuenta de que un video también servía para grabar películas desde el televisor, e incluso hacer copias de cintas originales. La industria lo aceptó como un mal menor, convencida de que los beneficios potenciales de vender millones de títulos superaría el perjuicio de algunas copias pirata de baja calidad.

La situación cambió radicalmente con el advenimiento de las películas en DVD. Ahora la industria corría un peligro mayor, puesto que era potencialmente más fácil copiar películas mediante un ordenador, y la copia tendría la misma calidad que el original. Pronto se cayó en la cuenta de que la criptografía podía ayudarles, no solamente restringiendo la copia de películas sino permitiendo a un estudio el control de sus productos hasta límites insospechados. Las películas podrían en principio estar protegidas con controles de acceso ajustables hasta límites marcados por la imaginación, y los precios de venta podrían variar en función del número de visionados por semana, la frecuencia de acceso, cualquier cosa. Si el usuario quiere ver la película en dos reproductores distintos, se ajusta una tarifa; si ve la película con poca frecuencia, se puede conceder un descuento; si lo que se desea es solamente ver la película durante un máximo de un mes, el disco puede programarse para “autodestrucción” y se inhabilitaría pasado el plazo convenido.

Parece ciencia ficción, pero corresponde a la filosofía del mundo audiovisual. Un detalle que el comprador de un DVD no suele saber es que realmente no está comprando la película. Lo que en realidad adquiere es el derecho a ver el contenido. La compra no implica la posesión, sino la transferencia de unos derechos. Imagínese un mundo en el que el vendedor de un coche pudiese imponer al comprador limitaciones sobre el kilometraje que puede hacer al año o cuántas veces puede salir de la provincia con el coche.

La industria cinematográfica no ha podido llegar tan lejos, pero no ha sido por falta de ganas de intentarlo. Un DVD actual tiene operaciones prohibidas para el usuario, y a pesar de haber pagado por el disco (perdón, por el derecho a disfrutar el contenido del disco) sigue obligado a tragarse los anuncios del FBI o del estudio de cine sobre lo ilegal y peligroso que es hacer copias. Otras operaciones prohibidas al usuario pueden incluir la imposibilidad de saltarse anuncios o promociones de otras

películas, así como la obligación de reproducir el disco en una región del mundo determinada. Y las protecciones que incluye el DVD impiden su copia en un ordenador. Esto último implica, como puede imaginarse, criptografía a diversos niveles. De eso trataremos en este tema.

Las técnicas originales para controlar el acceso a los contenidos de un DVD llevaron a la adopción de un estándar internacional denominado CSS (*Content Scrambling System*). Se trata de un sistema de cifrado empleado en prácticamente todos los DVD con material cinematográfico, cuyo propósito es evitar tanto la copia como la reproducción en dispositivos no autorizados. Para conseguir dicha autorización, el fabricante de reproductores de DVD ha de solicitar la correspondiente licencia, y pagar cierta cantidad por ello a la entidad designada para ello, la *DVD Copy Control Association* o DVD-CCA[30]. De ese modo, CSS ayudó a crear un estándar controlado y protegido por la industrial.

La industria de contenidos cinematográficos (en adelante, "la industria") tenía frente a sí una tarea ímproba. Debía asegurar la protección de miles de películas distintas, que podrían ser reproducidas en aparatos hechos por centenares de fabricantes distintos. Y, por encima de todo, tenía que evitar los errores cometidos por sus primos de la industria discográfica. El aviso que aparece en la web de la DVD-CCA es una alusión a Sony y sus fracasos, tan sutil como una patada en el ojo:

CSS, la tecnología de protección de contenidos usada en muchos títulos DVD, no interfiere en modo alguno en el manejo del ordenador. CSS ha sido diseñado para ser completamente transparente y no invasivo durante la reproducción normal, y no instala ningún programa de software, o archivo de ninguna clase, cuando se utilizan en ordenadores o en cualquier otro dispositivo de reproducción.

El primer elemento del sistema CSS es un dispositivo llamado "módulo de descifrado". Este es, por así decirlo, el motor criptográfico. El módulo se encargará de todas las tareas de autenticación y cifrado, permitiendo o prohibiendo la reproducción al reproductor DVD, y almacenará en forma segura todas las claves que se requieran.

Lo primero que hace el módulo de descifrado es autenticar el reproductor de DVD. Es decir, debe asegurarse de que el aparato está autorizado para reproducir discos en formato DVD. Este paso permitirá establecer una "relación de confianza" entre el disco y el reproductor.

A continuación, comienza un proceso que al lector puede parecerle complejo, pero que es necesario para poder garantizar la seguridad del proceso. Para poder descifrar el disco, hay una jerarquía de claves. Tenemos en primar lugar la llamada "**Clave de Disco (DK)**". Si un atacante la consiguiera podría usarla para reproducir el DVD en cualquier sistema sin necesidad de módulo de descifrado. Hay que protegerla bien. No podemos dejar la clave del disco en cualquier sitio, del mismo

modo que no dejamos las llaves del coche colgando de la cerradura.

La clave de disco, necesaria para leer los contenidos del DVD estará a su vez cifrada con una clave maestra, denominada "**Clave de Reproductor (PK)**". Cada fabricante tiene una clave de reproductor distinta: Toshiba tiene una, Siemens tiene otra, LG otra, y así una para cada fabricante hasta un total de 409. Es decir, hay un total de 409 claves de reproductor.

Por su parte, el fabricante se asegurará de que sus reproductores tengan incorporada la correspondiente clave PK. Ni que decir tiene que esa clave ha de ser custodiada y protegida fuertemente en todo momento. Lo que hacen habitualmente los fabricantes es "incrustarla" en un chip de hardware, que si está bien diseñado resulta prácticamente inexpugnable. Por su parte, los fabricantes de discos toman la clave de disco, y la cifran con las 409 claves.

Cuando la autenticación ha concluido con éxito, el reproductor de DVD toma su clave de reproductor, y con ella descifra y obtiene la clave de disco DK. Lo que hacemos con ella es utilizarla para descifrar un paquete de datos que contiene las **Claves de Título (TK)**, que son las que finalmente descifrarán los contenidos del DVD. De ese modo, las claves secretas del reproductor (PK) y del disco (DK) se combinan para desbloquear los códigos necesarios para descifrar y ver la película (TK).

En el caso de reproductores software, la cuestión es similar, pero aquí hay problemas adicionales de seguridad. Si una empresa de software quiere diseñar un programa para ver películas DVD, necesita toda la información cifrada que hemos comentado. El problema es que la protección software es mucho más débil que la de hardware. Un atacante tiene, en principio, acceso a toda la información en su propio ordenador, de forma que mantener la seguridad es más difícil.

Aquí es donde comenzaron los problemas para los fabricantes de DVD. Hacia 1996, el sistema de protección CSS estaba en vigor y protegía los discos de los estudios de cine del todo el mundo. Los usuarios podían utilizar cualquier reproductor o programa de software que tuviese la licencia correspondiente.

Había una excepción: los entornos informáticos basados en Linux. Este sistema operativo funciona bajo los principios de software libre, y una de sus premisas es que cualquier elemento de software ha de ser revisable por cualquier programador. Por supuesto, la DVD-CCA no estaba dispuesta a permitir que sus claves criptográficas, piedra angular de la seguridad de todo el sistema, acabasen en el código fuente de un programa Linux, y rehusaron dar licencias a programas basados en ese sistema operativo.

Mala idea. Aunque el porcentaje de usuarios de Linux era entonces muy pequeño, constituían (entonces como ahora) una comunidad intelectualmente muy ágil y activa. Al contrario que un usuario de Windows o Mac, un "linuxero" avanzado considera

normal el hacer sus propios programas, y la filosofía de software abierto asegura que cualquier programa hecho por un miembro será rápidamente diseminado por la comunidad.

Visto en retrospectiva, resulta irónico que en la actualidad la DVD-CCA sí conceda licencias a fabricantes de software en Linux[31]. Seguro que desearon haberlo hecho antes. Negar un reproductor a la comunidad Linux era la mejor receta para asegurarse el enojo de un gran grupo de personas con talento, conocimientos técnicos y voluntad. Y no solamente el enojo, sino también una respuesta contundente.

Entre esos linuxeros enfadados se encontraba un noruego llamado Jon Lech Johansen, quien se hizo mundialmente conocido por haber roto el cifrado CSS. La historia es algo más confusa, ya que al parecer se trató de un trabajo conjunto de dos grupos de hackers. Johansen codificó los resultados de otros en un programa de software para reproducir discos DVD llamado DeCSS[32], y se hizo muy famoso por ello, hasta el punto de que desde entonces le apodan “DVD Jon”.

El *cracking* del sistema CSS es un buen ejemplo de las principales técnicas de seguridad informática, desde la criptografía hasta la ingeniería inversa. Por ello, le propongo que dediquemos algún tiempo a examinar el sistema; después veremos cómo puede vencerse.

Lo primero que hemos de mencionar es el modo en que se consiguió obtener la descripción de CSS. Hay una rama de la seguridad informática conocida como “seguridad mediante oscuridad,” de la que ya hemos hablado en este libro, que básicamente viene a decir que la mejor forma de dificultar la tarea del atacante es ocultarle los detalles de los medios de defensa. Según esta filosofía, un ladrón de cajas fuertes tendrá más problemas si no conoce el mecanismo de funcionamiento de la caja.

Este es un mal modo de implementar la seguridad. La experiencia con siglos de uso de la criptografía muestra una y otra vez que la seguridad mediante oscuridad no funciona. ¿Por qué? Muy fácil: porque los atacantes son muy ingeniosos. Un ladrón de cajas fuertes puede obtener información de múltiples fuentes. Puede usar desde ultrasonidos hasta sobornos a empleados de la empresa constructora. Puede recurrir a imágenes térmicas, utilizar un estetoscopio para analizar los sonidos que hace el mecanismo al girar, comprar una caja fuerte para estudiarla y despedazarla. Las posibilidades son muy amplias.

Para evitar este tipo de ataques, la seguridad del sistema no debe depender de ocultar los detalles del mecanismo. El ladrón puede tener los planos completos de la caja fuerte, puede desmontar diez cajas iguales, puede preguntar a los diseñadores hasta el más mínimo detalle; y si a pesar de ello es incapaz de abrir la caja, entonces sí la consideraremos segura. Mantener los detalles secretos puede ayudar en la seguridad, pero no garantizarla a largo plazo.

Los criptoanalistas utilizan desde hace más de un siglo uno de los llamados *principios de Kerckhoffs*: la efectividad del sistema no debe depender de que su diseño permanezca en secreto. La seguridad mediante oscuridad es una violación flagrante de este principio de Kerckhoffs. Hay muchos ejemplos en los libros de historia. Como mucho, uno puede utilizar la seguridad mediante oscuridad para dificultar la tarea del atacante, pero nunca como elemento esencial de la seguridad integral.

El sistema CSS tiene en la práctica un talón de Aquiles: las implementaciones en software. Como ya hemos dicho, los fabricantes de reproductores de DVD pueden almacenar claves en hardware, un proceso muy seguro. Sin embargo, con un programa de software, la cosa cambia. Existen técnicas de ingeniería inversa que permiten obtener gran cantidad de información sobre la forma en que un programa informático funciona; y es mucho más difícil esconder una clave en un programa informático.

Se cree que fueron los desarrolladores de un grupo llamado *MoRE* (*Masters of Reverse Engineering*) los que consiguieron obtener unas de las 409 claves de reproductor, concretamente la correspondiente al programa de software XingDVD[33]. A partir de ahí, y usando ingeniería inversa, consiguieron averiguar cómo funcionan los algoritmos criptográficos en CSS. Esto nos da una oportunidad de descubrir cómo funciona un sistema de cifrado en el mundo real, así que vamos a aprovecharlo.

El corazón de CSS es un algoritmo conocido como LFSR, siglas en inglés que podemos traducir como Registro de Cambio por Retroalimentación Lineal. Hola, ¿sigue usted aquí? Espero que no se haya ido, asustado por lo que acaba de leer. Reconozco que el nombre impresiona, aunque se queda algo mejor en inglés (*Lineal Feedback Shift Register*), pero cuando vea en qué consiste se le irá el miedo. La verdad es que se trata de un truquito muy ingenioso.

La idea es la siguiente. Queremos construir un algoritmo de cifra que tome la clave, la combine con el archivo sin cifrar y nos de un archivo cifrado; y que a la inversa, con la clave y el archivo cifrado nos vuelva a dar el archivo original. Más concretamente, vamos a utilizar lo que se denomina un **algoritmo en flujo**, que realiza la operaciones de cifrado bit a bit. Es decir, combinamos un bit de la clave con uno del archivo, y nos da un bit cifrado.

Para cifrar, utilizaremos una operación parecida a la suma, denominada XOR ($\oplus$). Se trata de algo parecido a una suma, pero con la propiedad de que es su propia operación inversa; es decir. Eso significa que podemos utilizar la misma operación para cifrar que para descifrar, lo que la hace muy útil en aplicaciones criptográficas. La operación de cifrado consiste simplemente en una operación xor de cada uno de los bits de esas variables; igualmente sucede con la operación de descifrado.

El proceso es idéntico al que vimos en el tema “Los códigos de Dan Brown” en relación a la llamada Libreta de Uso Único (OTP). El problema aquí consiste en que la clave *K* y el mensaje *M* han de tener la misma longitud, lo que lo convierte en un método poco práctico. En nuestro caso, tendríamos que comprar dos DVDs, uno con la película y otro con la clave. No solamente resultaría muy ineficiente, sino que en principio bastaría con conseguir un disco con la clave para poder descifrar cualquier película; a no ser que los vendedores hicieran una clave distinta para cada disco puesto en el mercado, una idea muy poco práctica. La segunda mejor opción es generar un flujo de bits que no sean realmente aleatorios, pero que lo parezcan a primera vista. Es lo que se llama **pseudoaleatoriedad**.

La forma más habitual de conseguir largas cadenas de bits pseudoaleatorios es mediante un FSR (*Feedback Shift Register*) o Registro de Cambio por Retroalimentación; dejaremos lo de lineal para más adelante. Un Registro funciona mediante un conjunto de *n* bits. En un principio, proporcionamos al sistema *n* bits iniciales, lo que se llama “semilla” (*seed*). A continuación el Registro calculará el bit *n+1* como función de los *n* bits iniciales, eliminará el bit número uno y guardará el nuevo bit. Hacemos lo mismo otra vez: calcular un bit nuevo y tirar a la basura uno antiguo. Y otra vez. Y otra vez. Eso nos da un flujo de bits que, si hemos hecho bien los deberes, se asemejará mucho a una cadena de unos y ceros tomados totalmente al azar. Esa será nuestra clave *K*.

En principio, un Registro que utiliza *n* bits podrá darnos una cadena pseudoaleatoria de cómo mucho *2^n* bits. Después volverá a repetirse una y otra vez, con lo que serán perfectamente predecibles, pero si *n* es grande, nos dará una clave muy larga.

Vamos a inventarnos un FSR para ilustrar el concepto. Para mayor comodidad, haremos *n*=4, es decir, será un registro de cuatro bits. Nuestra “semilla” será 1111. Mi función será la siguiente: si el número de cuatro bits, pasado a notación decimal, es primo, entonces añadiremos un uno, y si no, un cero.

Como 1111 es el número 15, y no es primo, voy a añadir un cero a la izquierda, y borraré el bit de la derecha. De esa forma, nuestro 1111 se transforma en 0111. Ahora tenemos el número once, que sí es primo, de forma que introducimos un uno, borramos un uno, y obtenemos 1011. De esa forma, iríamos obteniendo la siguiente secuencia:

Estado anterior	¿Primo?	Bit nuevo	Estado nuevo
1111 (15)	NO	0	0111
0111 (7)	SI	1	1011
1011 (11)	SI	1	1101
1101 (13)	SI	1	1110
1110 (14)	NO	0	0111

```
0111 ( 7)          SI        1         1011
1011 (11)          SI        1         1101
```

La columna “bit nuevo” nos dará nuestro flujo pseudoaleatorio. Pero esperen, ¿no notan algo raro? El estado segundo (0111) se repite en la línea sexta. De hecho, si seguimos dando vueltas a la manivela, este es el resultado:

`011101110111011101110111`…

Si este fuese un buen FSR, el flujo no se repetiría hasta después de 2^4 = 16 bits, pero aquí se me repite ya en el quinto bit. Aunque hubiese introducido otro valor de semilla, el problema aparecería igualmente. Esto no es un buen generador de números pseudoaleatorios. Por supuesto, este registro no se usa en la práctica, y el motivo por el que aparece aquí es porque sencillamente es el primero que me ha pasado por la mente. Como ven, no soy un buen criptógrafo. Pero otros pueden hacerlo mucho mejor.

Dependiendo de cómo calculemos el bit nuevo, los Registros tienen diferentes nombres. Si las operaciones que realizamos entre los bits del Registro son sumas xor, el registro se llama lineal, es decir, tenemos un LFSR. Si lo hacemos bien, podemos obtener LFSR que nos den cadenas de hasta 2^*n* bits. He aquí un ejemplo sencillo: produzcamos un bit nuevo como resultado de sumar los bits 1 y 4. Es decir, los cuatro bits (*b4, b3, b2, b1*) se convierten en estos otros cuatro:

(*b4, b3, b2, b1*) → (*b1*⊕*b4, b4, b3, b2*)

Si partimos de la semilla (valor inicial) 1111, obtendremos la siguiente cadena de bits.

`101011001000111101011001000111`…

que, como ven, tiene un período de quince, muy próximo al máximo teórico de 16. Una “semilla” inicial 0111 nos da esto:

`100011110101100100011110101100`…

es decir, la misma repetición que antes, pero con un comienzo diferente. Con un LFSR podemos obtener un buen montón de bits que, aunque no son realmente aleatorios, lo parecen. Un registro con 40 bits nos daría un total de 128 gigabytes, suficiente para cifrar una docena de DVDs. Un LFSR es sencillo de construir, ocupa poca memoria, es rápido y eficiente. Por esos y otros motivos, es una pieza clave en muchos sistemas de cifrado, entre ellos el CSS.

Concretamente, el sistema CSS utiliza no un LFSR, sino dos. El primero, de 17 bits, funciona haciendo xor entre los bits 1 y 15. Es decir, su estado cambia así:

(*b17, b16, b12…, b2, b1*) → (*b1*⊕*b15, b17, b16…, b3, b2*)

El segundo LFSR utiliza 25 bits, y por si les interesa, obtiene un nuevo bit mediante un xor de los bits 11, 19, 20 y 23. La semilla de los LFSR proviene de la clave de título (TK) que vimos antes. En lo que respecta a la elección de ese tipo de registros en particular, podemos imaginarnos que hicieron múltiples pruebas y esa resultó adecuada en términos de rapidez y seguridad, produciendo el equivalente de un solo LFSR con 25+17 = 42 bits. En la práctica son solamente 40 bits, ya que dos de los bits de la semilla se introducen al principio y no forman parte de la clave. En cualquier caso, los dos flujos se combinan en uno para producir la clave que descifrará los contenidos del disco.

Ahora veamos las cosas desde el punto de vista del atacante. Lo primero que hay que tener en cuenta es que, como cabe esperar, un atacante conocerá todos los detalles del sistema, incluyendo el funcionamiento de los LFSR. Lo primero que puede hacer es dejarse de sutilezas criptoanalíticas y limitarse a probar todas las claves, lo que se conoce como **ataque de fuerza bruta**. Si el número de claves es lo bastante grande, esto resultará inviable en la práctica. ¿Es el caso del sistema CSS? La verdad es que no. Las claves de título o de reproductor tienen una longitud máxima de 40 bits. Esto nos da un total de 1,1 billones de posibilidades. A finales de los noventa, resultaba una tarea ímproba y tediosa (habría que hacerlo para cada título por separado, ya que cada uno tiene su propia clave de título), y un ordenador personal típico tardaría aproximadamente una semana en probar las 2^40 claves posibles.

Una semana parece mucho para poder ver un DVD, pero basta con que el esfuerzo lo haga una sola persona y que luego comparta la película descifrada en una red p2p. La pregunta que surge de modo inevitable es: ¿por qué utilizar claves de sólo 40 bits? Seguro que pueden utilizarse registros mayores que esas birrias de 25 y 17 bits ¿no?

Técnicamente, no hay problema. El conflicto aquí está en el campo legal. Para evitar el uso de tecnologías occidentales por parte de la Unión Soviética y otros países considerados hostiles durante la Guerra Fría, Estados Unidos y sus aliados firmaron en 1976 los acuerdos ITAR (*International Traffic in Arms Regulations*). Su propósito era controlar la exportación de material de guerra, incluidas las llamadas "tecnologías de doble uso," que aunque destinadas al mercado civil puedieran ser reconvertidas para uso militar.

Encontrar los productos criptográficos en la lista de prohibiciones ITAR no es extraño puesto que tienen multitud de aplicaciones militares, y por eso ocupan un lugar prominente en la llamada Categoría XIII sobre Equipo Militar Auxiliar[34]. Lo que ya no resulta tan lógico es que, desaparecida la URSS y acabada la Guerra Fría,

las restricciones ITAR permaneciesen en vigor. Ciertamente Estados Unidos tenía enemigos (y los sigue teniendo ahora), pero restringir categorías enteras de productos porque un día pudieran ser utilizados por un enemigo con capacidad destructiva limitada no tiene mucho sentido.

Como dijo una vez el experto en seguridad Bruce Schneier en 1999, no hay muchas películas cifradas que los terroristas necesiten ver[35]. Las restricciones criptográficas eran tan ineficaces como absurdas. Yo mismo tengo en mi despacho una copia de uno de sus libros más celebrados, *Applied Cryptography*, edición de 1996. Un amigo me lo compró en Londres a finales de 1998. Se supone que incluía un CD con algunos de los programas criptográficos que describía el libro, pero por las normas ITAR nunca me llegó. Sin embargo, un apéndice del libro contiene cincuenta páginas de código fuente correspondiente a varios algoritmos de cifra. No tengo más que compilarlos y ya está. Lo mejor del caso es que el gobierno norteamericano no podía restringir la exportación del código fuente porque, al estar impreso en papel, se consideraba protegido por la libertad de expresión.

¿Absurdo? Por supuesto. Pero, con independencia de la lógica que contuviesen estas normas, cualquier usuario legítimo de productos criptográficos tenía que atenerse a ellas si quería vender sus productos fuera de Estados Unidos. Las normas norteamericanas permitían la exportación, pero solamente con la condición de que las claves de cifrado utilizadas fuesen de una longitud máxima de 40 bits. Este límite estuvo vigente hasta finales de los años noventa, cuando la explosión de Internet y el auge del comercio electrónico hicieron ver al Tío Sam que continuar con la prohibición perjudicaría fuertemente a su industria. En junio de 1997, el fabricante del navegador Netscape recibió autorización para exportar un navegador con claves de 128 bits[36]. Microsoft obtuvo un permiso similar para su navegador Internet Explorer.

Para comienzos de 2000, y con ciertas condiciones, se permitía ya la exportación de sistemas con cualquier longitud de clave a casi todos los países del mundo. Pero en 1996, cuando la industria cinematográfica acordó el formato DVD, las restricciones estaban en vigor. Eso les forzó a utilizar claves de 40 bits. Esto no es lo ideal, pero representaba un buen nivel de seguridad contra un usuario individual. Para desgracia de la industria, Internet permitía a un gran número de usuarios beneficiarse del trabajo de unos pocos. Algún linuxero con tiempo libre y ganas de enfrentarse a un reto, unas cuantas neuronas en acción, y ya tenemos una película descifrada lista para subir a una red p2p. O mejor aún, podemos disponer del programa de descifrado directamente, como sucedió con DeCSS.

Veamos cómo las mentes inquietas de la Red vencieron el sistema CSS. Comencemos por los LFSR. Un detalle que no les he comentado hasta ahora es que su carácter lineal los hace altamente vulnerables a un ataque criptoanalítico particular.

Imaginemos un registro de *n* bits cuyos detalles internos no conocemos. De algún modo, el fabricante ha conseguido ocultarnos esa información. Pero no le servirá de nada, porque un atacante puede reproducir el registro (o bien construir otro registro distinto pero equivalente) sin más que examinar *2n* bits de salida. Este resultado, que aprovecha el carácter lineal del registro, se conoce con el nombre de algoritmo de Berlekamp-Massey.

Los diseñadores de CSS probablemente tuvieron eso en cuenta, y es por eso que utilizan dos registros LFSR. El resultado de los dos se suma para dar lugar al flujo de bits que formará la clave de descifrado de la película, y eso previene el uso del algoritmo de Berlekamp-Massey. Ese debe de ser el principal motivo por el que no se utiliza un solo registro, sino dos en combinación.

Pero los registros lineales usados en CSS tienen otras vulnerabilidades. Para describirlas, usaremos el concepto de complejidad. Diremos que un algoritmo de cifra tiene una complejidad *m* si reventar el sistema nos requiere el mismo trabajo que probar *m* claves distintas. En un sistema perfecto con clave de *n* bits de longitud, la complejidad del sistema es exactamente *2^n*, lo que significa que no hay atajos. Cuando más vulnerable sea, menor será la complejidad y más fácil será obtener la clave y descifrar el texto protegido.

En ocasiones, la complejidad de un sistema disminuye si conocemos algo sobre él, y este es el caso de los LFSR de CSS. Supongamos, por ejemplo, que conocemos los primeros cinco bytes de salida del LFSR combinado (es decir, la suma de los dos que estamos usando). En ese caso, existe una táctica que permite recuperar la clave inicial de 40 bits, que recordemos se usa como semilla inicial de los registros.

En circunstancias normales, obtener esos cinco bytes no sería nada fácil. Sin embargo, en este caso es factible. Resulta que, cuando CSS está descifrando la claves de título, utiliza una operación denominada “exprimido” (*mangling*). Esa operación tiene un fallo, de tal forma que si conocemos el mensaje llano y el mensaje cifrado podemos obtener exactamente cinco bytes de la clave generada por el LFSR. Ambos fallos, el del LFSR y el del “exprimidor,” se combinan para crear una vulnerabilidad crítica. Como resultado, la complejidad del sistema se reduce a tan sólo 2^16. O dicho de otro modo, la seguridad de todo el esquema es equivalente a la de un sistema de cifrado con clave de 16 bits.

Otro ataque criptoanalítico, igualmente devastador, nos permite obtener la clave de disco **DK**. Este ataque fue descrito por Frank A. Stevenson en 1999[37] y no entraremos aquí en detalles. Solamente le diré que podemos obtener dicha clave con una complejidad de 2^25. Lo que llevaba una semana ahora puede conseguirse en apenas unos segundos.

Los esfuerzos de Johansen y otros permitieron hacer programas para visualizar discos DVD en entornos Linux. Cualquiera podría tomar un DVD original y hacer

una copia saneada (sin cifrado, operaciones prohibidas o códigos de zona) mediante programas como DVD Decrypter. A todos los efectos, la seguridad de CSS estaba totalmente rota antes de terminar la década de los noventa.

Eso dejó a la industria del video con una sola opción: el frente legal. DVD Jon fue llevado ante los tribunales noruegos por hacking informático en 2002, a petición de la *DVD-CCA*. Puesto que no había violado ninguna ley de su país, el juez emitió una sentencia absolutoria. El programa que redactó para ver videos en entornos Linux sigue en la red, y otros desarrolladores de software lo imitaron.

Una de las respuestas de la industria fue instar al gobierno norteamericano a cambiar la ley. El resultado fue la DMCA (*Digital Millennium Copyright Act*), que entre otras cosas prohíbe la producción y distribución de programas o sistemas diseñados para sobrepasar medidas anticopia. También limita el uso de los métodos de ingeniería inversa. Como resultado de ello, los creadores de DVD Decrypter tuvieron que retirar su programa en 2005, si bien existen copias en Internet[38].

La DVD-CCA intentó en diversas ocasiones mejorar el sistema CCS, lo que no es tarea fácil en absoluto. Cualquier solución deberá pasar no solamente por mejorar el sistema existente, sino también por asegurar su compatibilidad con los discos y reproductores ya existentes. En al menos tres ocasiones (1999, 2001 y 2005) intentaron evaluar sistemas de "marca de agua" (*watermarking*) para "*marcar contenido audiovisual para transmitir cierta información de control de copia… con el objeto de mejorar el sistema de protección de copia CSS con respecto a información audiovisual*".

A tenor de los requisitos descritos, la idea parece ser que los contenidos originales lleven una especie de marca digital, que no pueda ser borrada en caso de alteraciones del archivo original, y que incorporen códigos especiales capaces de identificar a) si se ha usado CSS como protección y b) si la copia está autorizada. Los reproductores de DVD futuros incorporarían dispositivos para impedir la reproducción de copias no autorizadas[39]. En los tres casos, el concurso para encontrar una solución viable quedó desierto.

Algunas empresas cinematográficas desarrollaron sus propios esquemas de protección adicionales en sus DVDs. La solución de Sony fue un sistema presentado en 2007 y denominado ARccOS (*Advanced Regional Copy Control Operating Solution*). Considerando el historial de Sony a la hora de proteger los discos con música, deberíamos creer que aprendieron del error. No parece ser el caso. ARccOS funciona creando sectores corruptos (por tanto, ilegibles) en ciertos lugares del DVD. De ese modo, los programas anticopia producen errores de copia y no pueden funcionar. Por desgracia, esto causaba que algunos reproductores de DVD no funcionasen correctamente[40]. Poco tardaron en aparecer programas copiadores que se saltaban esa protección.

Un segundo método, utilizado por Disney, Paramount y Warner, se conoce como Disney Fake o Disney X-Project. Reconozco mi incompetencia a la hora de encontrar información sobre este sistema. Sí puedo confirmar que la copia en DVD de *Piratas del Caribe 4: en Mareas misteriosas* no puede ser copiado con los "dvd rippers" habituales, si bien se anuncian en Internet programas que pueden hacerlo.

Sin embargo, no hay un estándar acordado para su uso por toda la industria. Un consorcio llamado *4C Entity*, formado por cuatro empresas de electrónica (IBM, Intel, Matsushita e Hitachi) propuso en 1999 un sustituto a CSS llamado CSS2, pero la propuesta fue rápidamente retirada. La solución pasaría por comenzar a diseñar desde cero un buen esquema de cifrado. El resultado, publicado en 2005, fue el sistema AACS (*Advanced Access Content System*). Pero nunca fue utilizado en discos o reproductores DVD. Los problemas de compatibilidad con los sistemas existentes lo impedían.

No, el AACS tenía otros destinatarios en mente: los nuevos formatos de alta definición HD DVD y Blu-ray. La industria subía el listón para asegurar la protección del sistema sucesor al DVD, haciendo todos los esfuerzos imaginables para corregir y evitar los errores del pasado. Por su parte, la comunidad internauta, vencedora de la batalla del DVD, estaba lista para afrontar los nuevos retos. La guerra de los medios digitales entraba en una nueva fase.

3) LA ALTA DEFINICIÓN

A comienzos del siglo XXI, la industria videográfica se planteó seriamente la conveniencia de actualizar las especificaciones existentes para el video digital (DVD) y dar el salto a un nuevo formato. Las razones fueron fundamentalmente tres. En primer lugar (sin ningún orden en particular), se deseaba poder introducir más información en los discos, aumentando así la calidad y definición de las imágenes. En segundo lugar, se confiaba en que la alta definición digital conllevase una nueva revolución en el consumo de productos audiovisuales, similar a la que generó el paso del formato analógico (VHS, Beta) al digital (DVD), lo que sin duda contribuiría a revitalizar la industria del ocio. Finalmente, un nuevo conjunto de especificaciones de seguridad bien hechas sustituiría al sistema CSS, que como hemos visto tiene más agujeros que un colador.

De forma similar a como VHS y Beta compitieron por el mercado de vídeo analógico, dos estándares pugnaron durante años por imponerse: HD DVD, patrocinado por Toshiba, y Blu-ray, impulsado por un consorcio liderado por Sony. Tras una guerra de varios años, Toshiba acabó cediendo y retiró su sistema, dejando a Blu-ray como único formato de discos compactos de alta definición.

Desde nuestra óptica de seguridad, poco importa el resultado, ya que ambos sistemas estaban dotados del mismo sistema de seguridad llamado AACS (*Advanced Access Content System*). Este estándar fue publicado en 2005, y representa un salto cualitativo respecto a su predecesor CSS. La industria hizo un serio intento por superar los fallos anteriores, y un análisis de AACS muestra que realmente se tomaron la seguridad en serio, abandonando casi totalmente la “seguridad mediante oscuridad”.

La primera mejora se observa en los algoritmos básicos de cifrado y firma digital. En CSS, se utilizaron algoritmos hechos por la propia industria, en lugar de confiar en sistemas seguros y probados. Por el contrario, el cifrado y descifrado en AACS utiliza el algoritmo AES (*Advanced Encryption Standard*), que ha sido sometido a multitud de pruebas durante años y se considera uno de los mejores sistemas de cifra existentes en la actualidad. AES no sólo sirve para cifrar, sino que también es utilizado para otras aplicaciones seguras como generar números pseudoaleatorios. Primer punto a favor de los propietarios de AACS, que por cierto es un consorcio de empresas agrupadas en la entidad *AACS Licensing Administratior* (AACS-LA).

Tampoco bajaron la guardia en otros aspectos criptográficos igualmente importantes, a saber: funciones hash y firma digital. En el primer caso, se utiliza una función con propiedades especiales para poder representar grandes archivos mediante cadenas alfanuméricas pequeñas; es un paso especial para poder realizar operaciones de firma digital. AACS utiliza SHA-1 como función hash, y EDCSA para firma

digital mediante criptografía de curva elíptica. Ambas funciones han sido ampliamente probadas y se las considera de lo mejorcito en sus respectivos campos. En todos esos casos, no hay seguridad mediante oscuridad que valga, ya que todos los detalles de estos algoritmos han sido hechos públicos y comprobados hasta la extenuación[41].

Hasta este punto, todo indica que la industria ha echado la casa por la ventana. Nadie podrá encontrar fácilmente una vulnerabilidad o un fallo. Esto es lo que deberían haber hecho desde el principio, y aunque metieron la pata a lo grande en el caso de los DVD, por lo menos hemos de reconocerles que aprendieron de sus errores. Casi dan ganas de aplaudir.

Pero escoger buenos algoritmos criptográficos es un paso necesario, no suficiente. Es necesaria una jerarquía de claves para asegurarse de que las claves no sean fáciles de extraer por terceros no autorizados. Hay que implementar mecanismos para evitar la copia no autorizada. Finalmente, sería deseable establecer un sistema de revocación que permita bloquear un reproductor que haya sido comprometido de algún modo; de esa forma, si un aparato Blu-ray ha sido usado para piratear películas, podrá ser bloqueado en el futuro.

El sistema AACS permite ambas cosas. Para ello, juega con las claves como si fuesen un conjunto de muñecas rusas, guardando unas dentro de otras. El esquema puede parecer caprichoso, pero está diseñado para garantizar diversos aspectos dentro de la seguridad global.

La primera diferencia que nos encontramos es que ahora no hay una clave de reproductor para cada fabricante. En el caso de CSS, había un total de 409 claves de reproductor, una para cada fabricante. Si un hacker consigue obtener la clave de un fabricante (digamos, la de Toshiba), puede utilizarla para descifrar todas las películas que quiera. En su lugar, podemos pensar en insertar una clave distinta para cada aparato reproductor individual. Es decir, si Toshiba vende diez millones de copias de un modelo en particular, cada uno de esos diez millones de reproductores llevará una clave diferente, llamada **Clave de Dispositivo** (*DK*).

Eso permite a la industria algo muy útil: la revocación. Digamos que uno de los diez millones de reproductores Toshiba ha sido usado para descifrar películas, o para copiarlas sin permiso. Una vez se haya identificado el culpable, su clave de dispositivo puede ser revocada, es decir, inutilizada. De esa forma, ese reproductor en particular no podrá volver a ser usado para reproducir discos (ni aunque sean discos legítimos), y el hacker tendrá que tirar su Toshiba nuevo a la basura.

El problema es que puede haber mil millones de reproductores Blu-ray (y HD DVD, aunque de esos apenas quedan ya) en el mercado, y como un disco no sabe en qué reproductor va a ser insertado, en principio tendríamos que almacenar en cada disco mil millones de claves para poder descifrar las películas. Una cantidad tan

grande de claves no cabría en el disco. Para evitar el problema, las especificaciones AACS utilizan un procedimiento denominado “árbol de diferencia de subconjuntos”. Los detalles son algo tediosos, pero se lo explicaré con un ejemplo.

Digamos que usted llega a vivir a una gran urbanización, donde cada puerta se abre mediante un código numérico de cuatro cifras, como el PIN de su tarjeta de crédito. En primer lugar, hay una puerta común para entrar en el complejo. A continuación, una segunda puerta guarda la entrada del edificio. Una tercera puerta impide la entrada a la planta. La cuarta puerta protege el pasillo, y la quinta es la entrada de su vivienda. Como ve, hay que atravesar cinco puertas, y por tanto usted necesitará cinco códigos PIN para llegar a su vivienda. Usted no paga las cuotas de la comunidad, y la junta ha decidido bloquearle el acceso. Para conseguirlo, ordena reprogramar el sistema, de forma que cuando se introduzcan los cinco códigos PIN de usted, la puerta no se abra. Cada uno de esos PIN abre una puerta de forma legítima, pero cuando está usted a punto de entrar en su vivienda, una pantalla cercana le indica “la combinación de códigos 1193, 9733, 0119, 6251, 8111 ha sido desactivada”. Esa combinación de cinco números PIN está en la “lista negra” y no se permite su uso, a pesar de que cada PIN individual abre perfectamente la puerta que se le ha asignado.

Un esquema algo similar sucede en AACS. Cada aparato reproductor recibe no una, sino un conjunto de **Claves de Dispositivo**, con las que se construye la llamada **Clave de Procesamiento**. A su vez, la Clave de Procesamiento descifra un bloque llamado MBK (*Media Block Key*), que contiene la información sobre los aparatos que están en la lista negra. Si en algún momento se detecta que un reproductor determinado “amenaza la integridad del sistema” (es decir, se ha usado para piratear o copiar una película), la información de ese aparato será incluido en el MBK y diseminado en las películas que se graben en el futuro. Más adelante, el dueño de ese aparato comprará una película original e intentará reproducirla, pero ese nuevo disco ya contendrá la información para revocar ese reproductor, y no permitirá el descifrado de la película. Es algo así como tratar con un tramposo que se va del bar sin pagar: hoy se sale con la suya, pero mañana todos los bares de la zona tendrán su descripción y no podrá repetir el truco.

Este esquema de revocación es algo que no se podía hacer en el sistema CSS de los DVD, donde todo lo más que se podía hacer era un “todo o nada”. Si se detecta que un reproductor Toshiba determinado ha participado en una actividad de copia no autorizada, lo único que se podía hacer era eliminar la clave de los reproductores Toshiba. El problema es que con ello estamos bloqueando todos los reproductores que Toshiba ha fabricado jamás, desde el primero al último. Dudo que los otros 9 999 999 propietarios se lo tomasen bien. Por el contrario, ahora podemos afinar la revocación hasta el punto de que podemos bloquear un aparato reproductor

determinado sin que los demás se enteren siquiera. Este procedimiento también se puede aplicar en software, aunque como ya hemos mencionado en otras ocasiones la seguridad en software es menor que en hardware y se precisan mecanismos adicionales de autenticación.

El siguiente paso en la cadena de seguridad impide la copia no autorizada. Una vez que hemos insertado nuestro disco, y se ha comprobado que nuestro reproductor no está en ninguna lista negra, se genera una clave llamada **Clave de Medio** (*Km*). Un código numérico llamado **Identificador de Volumen** se extrae del disco, y ambos datos se procesan mediante el algoritmos AES. El resultado, si todo va bien, es una nueva clave llamada **Clave Única de Volumen** (*Kvu*). El truco consiste en que el identificador de volumen no puede reproducirse fácilmente. No está almacenado en el disco como si fuera un archivo normal y corriente, y solamente se puede acceder a él si se utiliza una clave especial que está oculta en el reproductor. De esa forma, ni siquiera una copia del disco bit a bit permitirá su reproducción.

Si el sistema está satisfecho con nuestra honradez, nos habrá permitido crear la Clave Única de Volumen, ya nos estamos acercando al final. El disco almacena la llamada **Clave de Título** (*Tk*), que es necesaria para ver la película. Esa clave está protegida mediante cifrado, y la clave de descifrado es precisamente la clave única de volumen; así que tomamos esa clave de volumen, la usamos para obtener la clave de título, y será dicha clave la que descifrará la película. Puede que una película tenga más de una clave de título, pero a estas alturas esos detalles no nos importan. Lo único que debe ya preocuparnos es que las palomitas no quemen y que nadie nos importune mientras nos recostamos en el sillón a ver nuestra estupenda película en alta definición.

Los grandes estudios de cine comenzaron a producir y vender discos en alta definición, confiados en que AACS proporcionaban por fin la seguridad que deseaban. Nadie podría sobrepasar las barreras de AES, no había forma posible de descifrar una película y colgar los resultados en Internet. Se acabó la tontería. Lo único que había que hacer era esperar a que los usuarios se convirtiesen en masa a la alta definición, inundando las tiendas y llenando de dólares las cajas fuertes de Hollywood.

Para su desgracia, eso nunca sucedió del todo. Para cuando la larga guerra de la alta definición entre HD DVD y Blu-ray se saldó con victoria de esta última, muchos usuarios ya se habían pasado a sistemas de descarga directa (*streaming*), o bien permanecieron fieles al formato DVD, que aunque de menor calidad era muy sencillo de copiar y compartir. Incluso hoy día, el futuro de Blu-ray permanece en entredicho, y no está claro hasta qué punto logrará arraigar. Pero por lo menos, no se trataba de una decisión basada en la inseguridad del formato de alta definición. AACS era sólido como la roca.

Aunque a veces aparecía alguna pequeña grieta que daba a los fabricantes un susto ocasional. En julio de 2006, la revista alemana *c't* descubrió una forma de trampear el sistema. Uno de los reproductores de películas en Windows, un programa llamado WinDVD, permitía hacer una copia de la pantalla, es decir, extraía un fotograma en alta definición. Un hacker con mucha paciencia no tendría más que irlas guardando como archivos, uno tras otro, y luego agruparlos para recomponer el video. Un hacker con menos paciencia y más habilidad lograría automatizar esta tarea.

Aunque este tipo de actuación podía haber llevado a WinDVD a la lista negra, no fue necesario tomar una medida tan drástica. La AACS-LA reconoció que Intervideo, fabricante del programa WinDVD, había cumplido las especificaciones de seguridad AACS al pie de la letra. Una pequeña actualización al programa, y problema resuelto[42].

Esta anécdota difícilmente constituye algo que pueda considerarse como romper la protección AACS, pero fue un recordatorio de dos cosas: primera, que ningún sistema de seguridad es completamente impenetrable; segundo, que hay millones de personas ahí fuera con tiempo e imaginación, y basta una de ellas para arruinarte el día.

Ese día llegó con las navidades. El 27 de diciembre de 2006, un hacker con el seudónimo muslix64 publicó en Internet un programa llamado BackupHDDVD, que pretendía nada menos que descifrar contenidos protegidos con AACS. El anuncio, publicado en el foro forum.doom9.org[43], levantó una gran polvareda. Cuando se asentó el polvo, se supo la verdad: realmente AACS no había sido comprometido. El sistema seguía siendo sólido. Pero se descubrió algo casi igual de malo: una forma de obtener las claves de título.

El propio muslix64 lo explicó todo[44]. Su motivación fue similar a la de DVD Jon cuando reventó el sistema CSS. Resulta que sus programas de software (en Windows) no podían reproducir contenido en alta definición. Se puso a pensar, y en un par de semanas descubrió una manera de obtener la clave de título *Tk*, que recordemos es la que, tras todos los pasos de verificación, utiliza el sistema para descifrar la película. Durante el proceso, la clave de título se encuentra en algún lugar de la memoria del ordenador. Así pues, ¿por qué no limitarse a reproducir el disco, hacer un volcado de memoria y buscar algo que se parezca a una clave?

Una vez conseguida la clave, lo único que había que hacer era confeccionar un programa que, al introducirle la clave, descifre y reproduzca los contenidos cifrados. Eso es lo que hace su programa BackupHDDVD[45], puenteando todas las demás protecciones que hemos descrito anteriormente. El problema estriba en que hay que darle las claves, ya que el programa no las recupera; pero la ventaja es que cualquiera que sepa hurgar en la memoria de un ordenador puede recuperar la clave de un disco

y colgarla en Internet. Un segundo programa para Blu-ray (BackupBluRay) fue liberado por el mismo autor en enero de 2007[46].

Pronto se dio un paso más en esta escalada criptográfica. Recordemos que la Clave de Título *Tk* está (o están, si hay más de una en el disco) protegida con la Clave Única de Volumen *Kvu*. En algún momento el sistema tendrá que sacar la *Kvu*, y en ese momento estará almacenada en memoria. ¿Podemos repetir el truco y extraer la *Kvu* de la memoria? Pues resulta que sí se puede. Poco tardó muslix64 en modificar sus programas para utilizar directamente la Clave Única de Volumen.

Los esfuerzos de diversos grupos de internautas se agruparon en webs como aacskeys.com y hdkeys.com, donde se llegaron a acumular hasta trescientas claves de películas, la mayoría obtenidas de discos HD DVD. En la actualidad esas webs ya no existen, pero aún pueden encontrarse copias si se sabe dónde buscar. Por ejemplo: [47] para aacskeys.com y [48] para hdkeys.com. En forum.doom9.org, todavía se puede acceder al *thread* sobre claves HD DVD[49], así como a un archivo que las contiene todas[50]. Un archivo similar guarda las claves obtenidas, por procedimientos similares, para los discos en formato Blu-ray[51].

Es importante reseñar que, en todos estos casos, no se trata de un ataque criptoanalítico en términos convencionales. Nadie saltó las protecciones criptográficas, nadie encontró una vulnerabilidad en AES o SHA-1. Tan sólo se trató de aprovechar el eterno talón de Aquiles: cómo usar claves sin que éstas puedan ser leídas y copiadas. En los reproductores de hardware, resulta relativamente sencillo proteger las claves contra lectura, pero en aplicaciones software, como hemos visto, es algo extraordinariamente difícil. Basta con que un solo vendedor de software lo haga mal para que el sistema se venga abajo poco a poco.

En el caso de los reproductores de software HD DVD, que fue donde comenzó el problema, se mencionó en primer lugar el programa PowerDVD; posteriormente se localizó el principal "culpable" de la filtración, o cuando menos, el programa que utilizaron hackers como muslix64 para extraer información sobre las claves. Este programa es el WinDVD, de Intervideo, el mismo programa donde se detectó el truco de "hagamos una película grabando los fotogramas uno a uno". La pobre Corel Corporation tuvo que tragarse un sapo amargo, ya que acababa de adquirir Intervideo apenas dos semanas antes de que muslix64 hiciese público su primer ataque. Seguro que el CEO (consejero delegado) de Corel pensaba en otra cosa cuando anunciaba que "*estamos centrados en presentar software que le inspire a usted, de modo que desee usarlo,*" aunque el propio muslix64 puede suscribir esas palabras[52].

También la AACS-LA tuvo que reconocer la gravedad de los ataques. El 24 de enero de 2007, una nota de prensa reconoció que "*algunas Claves de Título de AACS han aparecido en webs públicas sin autorización,*" al tiempo que le daba un tirón de orejas a los vendedores de WinDVD (sin nombrarlos explícitamente) y anunciaba el

uso de “todos los remedios apropiados,” lo que sin duda significa notificaciones legales DMCA para detener la proliferación de claves. Puede pensarse que lograron un cierto éxito, ya que como resultado ahora no se encuentran accesibles las webs originales donde se ofrecían gratuitamente tanto las claves como los programas de reproducción, pero Internet es un animal con memoria. El mero hecho de que yo, varios años después, haya logrado encontrar copias de toda la información relevante, es una indicación de la futilidad de intentar ponerle puertas al campo.

Las cosas se pondrían peor para la industria. Mucho peor. En febrero, entró a escena un segundo hacker, apodado arnezami, cuyas hazañas harían palidecer a las del propio muslix64. Arnezami se dedicó a extraer toda la información posible de un disco de alta definición. Su idea, al parecer, es ir subiendo por el “árbol de jerarquías” del sistema criptográfico, obteniendo cada vez claves más importantes.

Arnezami comenzó con el Identificador de Volumen (VID, a partir de ahora), ese código que usa el sistema AACS para comprobar si una película es original o ha sido copiada. Estuvo probando suerte con la película *King Kong*, y este es el VID que consiguió encontrar[53]:

```
40 00 09 18 20 06 08 41 00 20 20 20 20 20 00 00
```

Lo más curioso del caso es que, como comprobaron otros expertos, esta cadena numérica no es aleatoria. Muy por el contrario, tiene una estructura. Lo primero que llama la atención es el hecho de que, aunque el VID está escrito en formato hexadecimal, todos sus números son decimales. No es algo significativo, pero resulta llamativo. En segundo lugar, durante el proceso para descubrir la Clave de Volumen de esa película, se comprobó que estaba relacionada con la fecha en la que la película fue creada para ser estampada en los discos. En el caso de King Kong, la fecha era el 18 de septiembre de 2006, y la hora era las 8 y 41 minutos de la mañana.

Para que entiendan lo que ello implica, voy a poner la fecha en formato norteamericano, donde escriben primero el mes y luego el día: 09-18-2006. Ahora, la hora: 08:41. Y ahora fíjense de nuevo en la VID de King Kong. Si se dan cuenta, verán que comienza con *40 00* y acaba con *00 20 20 20 20 00 00*. ¿Y qué hay entre medias? Pues la cadena numérica 09 18 20 06 08 41. ¿Lo captan? 09-18-2006 08-41? ¡El VID no es más que la fecha de fabricación del disco!

Un análisis de otros discos HD DVD reveló que no siempre el VID se construye a partir de una fecha y hora de fabricación, pero las alternativas son asimismo predecibles. El VID de la película *Serenity* contenía la cadena hexadecimal 53 45 52 45 4E 49 54 59, que cuando se pasa a código ASCII nos da la palabra *SERENITY*[54]. De modo similar, el VID de *La Chaqueta Metálica* (*Full Metal Jacket*) no es más que la representación ASCII de *FULLMETALJAC*. Los VID de algunas películas incluían la cadena hexadecimal 57 47 48 44 56 4D, que en ASCII se convierte en la críptica

cadena alfanumérica WGHDVM.

El propio Arnezami calificó este resultado como increíble, y no podemos menos que estar de acuerdo. Si hay algo que reduce la seguridad de un sistema, es el hecho de usar claves o códigos identificativos fácilmente predecibles. El VID fue creado con la intención de evitar la copia no autorizada, y en su arrogancia, la industria de Hollywood utilizó identificadores de volumen altamente predecibles, confiando en que la oscuridad que protege al VID sería suficiente. Por si no fuera suficiente, lo rodean siempre con las mismas cadenas numéricas, lo que para un programa que efectúe búsquedas en memoria es lo mismo que una gigantesca bandera que diga "estamos aquí, dispare cuando quiera".

La ventaja de conocer el VID es que, si además tuviésemos acceso a la Clave de Medio, podríamos descifrar la Clave Única de Volumen *Kvu*. No tendríamos que ir buscándola en un título tras otro, sino que tendríamos una forma de calcularlo sin más que echar mano al VID. Por supuesto, eso es equivalente a afirmar que, si yo fuese millonario, estaría dándome la gran vida. Ese "si" es condicional, no afirmativo, y para mi desgracia resulta que yo no soy millonario, y el paisaje que veo por la ventana no es precisamente el de las playas paradisíacas de la Polinesia Francesa.

Soñar no cuesta nada. Salvo por el detalle de que hackers como muslix64 y arnezami ya habían realizado logros que supuestamente no eran posibles. Las claves de título no deberían ser accesibles, pero podemos acceder a ellas. La clave única de volumen está protegida mediante cifrado, pero podemos conseguir una copia. El identificador de volumen no debería ser visible, pero podemos verlo. ¿Hasta dónde podemos llegar siguiendo ese camino?

El siguiente gran paso sería, por tanto, conseguir la Clave de Medio, pero el proceso de "esnifar" en memoria ya no nos ayuda. El esquema de diferencia de subconjuntos, usado por el sistema para comprobar si un reproductor ha sido bloqueado, hace muy difícil la obtención de la Clave de Medio. Lo increíble del caso es que arnezami lo consiguió. El 11 de febrero de 2007, publicó la Clave de Medio para la película *King Kong*[55]:

```
07 4E 1F C8 8F B9 B7 80 A2 25 CA A2 3B C3 DB 56
```

Se trataba de una gran noticia, pero arnezami apuntaba a lo más alto: quería las claves maestras, las que controlan la seguridad de todo el sistema. Recordemos el sistema de capas que conforman AACS: hay un conjunto de **Claves de Dispositivo**, con las que se podía construir una **Clave de Procesamiento**, la cual controla el MKB (*Media Key Block*), que a su vez construía la **Clave de Medio**, que a su vez produce la **Clave Única de Volumen**, que a su vez descifra la **Clave de Título**, que es la que descifra la película. Hasta ahora hemos visto que se puede acceder a todas salvo a las más importantes: las de Dispositivo y las de Procesamiento. Son las claves maestras,

y su revelación podría echar abajo todo el sistema. Los creadores de las especificaciones AACS lo sabían, e hicieron todo lo posible por mantenerlas fuera del alcance de ojos fisgones.

El propio arnezami comparte con nosotros su experiencia[56]. Sus intentos por obtener las claves de dispositivo fueron inútiles. Pero descubrió que la clave de medio se borraba de la memoria en cuanto ya no era necesaria. Pensando que lo mismo le sucedía a la clave de procesamiento, y puesto que había comprobado que todas las claves importantes se almacenaban en la misma zona de la memoria del ordenador, se dedicó a comprobar todos los cambios que se hacían en dicha zona. De esa forma, con un poco de paciencia, el 11 de febrero de 2007 logró encontrar y capturar la Clave de Procesamiento:

```
09 F9 11 02 9D 74 E3 5B D8 41 56 C5 63 56 88 C0
```

Resulta difícil exagerar la importancia de este descubrimiento. La Clave de Procesamiento es la que, como he mencionado antes, controla el *Media Key Block* y sus listas de revocación. Está por encima del esquema de revocación, así que ella misma no puede ser revocada; de hecho, es ella la que decide qué claves son revocadas. Es lo más próximo al control total que existe. Y era un control en manos del usuario, ya que a mediados de marzo el propio arnezami hizo público un programa llamado aacskeys, que buscaba automáticamente todas las claves necesarias en el disco. Ya no hacía falta hackear el sistema a mano, ni buscar claves en segmentos de memoria. Era un sistema de "pulsar y usar," que no requería conocimientos técnicos por parte del usuario[57].

El 11 de febrero de 2007 fue un día negro para la industria del video digital. El día 15, la AACS-LA tuvo que emitir una nota de prensa en la que reconocía la publicación de la Clave de Procesamiento, pero la fraseología que usaron intentaba quitarle hierro al asunto: "*es una variación de un ataque ya publicado… no representa un impacto adverso para la capacidad del ecosistema de AACS de responder al ataque*".

Lo cierto es que las reglas de juego habían cambiado radicalmente. El cambio de claves era inútil a estas alturas. El descubrimiento de la Clave de Procesamiento no podía arreglarse más que cambiándola, o bien modificando el *Media Key Block*. En cualquier caso, todos los discos de alta definición que habían sido puestos en circulación hasta la fecha podrían ser leídos y copiados impunemente. Cualquier solución solamente serviría para proteger los discos futuros, no los ya publicados.

Finalmente, el día 24 de febrero un hacker llamado ATARI Vampire dio el golpe final al sistema: buscando en la zona de memoria ya sugerida por arnezami, consiguió extraer la Clave de Dispositivo correspondiente al programa WinDVD8[58]:

```
AA 85 6A 1B A8 14 AB 99 FF DE BA 6A EF BE 1C 04
```

Esto cambiaba totalmente las cosas para la industria. No solamente habría que modificar la Clave de Procesamiento, sino que las mismísimas Claves de Dispositivo estaban en peligro. Que solamente se hubiese extraído para el programa WinDVD8 no significaba que los demás reproductores en software no estuviesen en peligro, y cada uno de ellos era una fuente en potencia de claves reveladas y películas extraídas. De hecho, apenas diez días después otro hacker, llamado jx6bpm, publicó la Clave de Dispositivo del programa PowerDVD[59]:

```
47 37 67 60 58 D7 02 94 52 51 4F 0A B1 86 DC 4C CA 8C 57 8F
```

Para empeorar las cosas, una empresa de software llamada SlySoft aprovechó para pescar en río revuelto. Esta compañía está radicada en una isla caribeña, fuera de la jurisdicción norteamericana, y desde allí comercializa diversos programas para… hmmm… realizar copias de seguridad. AnyDVD, uno de sus programas, fue actualizado el 15 de febrero de 2007 para poder realizar copias de discos HD DVD protegidos con AACS. El 5 de marzo ya podía reproducir discos Blu-Ray[60]. Al no ser una empresa licenciada, no podía obtener legalmente ninguna clave, pero según jx6bpm AnyDVD incorporaba la Clave de Dispositivo de PowerDVD. Ya no se trataba de un entretenimiento para grupos de hackers con mucho tiempo libre, sino de un ataque en toda regla contra el modelo de negocio de la industria cinematográfica.

Fue la gota que colmó el vaso de la AACS-LA, que cambió a mediados de abril las claves de ambos reproductores. En principio deberían haberse revocado, pero eso hubiera afectado a todos los usuarios de esos programas impidiéndoles reproducir discos legítimos, así que se optó por un cambio de claves. En este caso, como se trataba de software, resultaba muy fácil de realizar: bastaba con proceder a una actualización online. Las dos empresas responsables, Corel[61] y Cyberlink[62], intentaron hacerlas pasar por actualizaciones rutinarias, pero por supuesto esta medida no cogía de sorpresa a los que sabían el verdadero motivo.

Los responsables de seguridad de la alianza AACS-LA tenían delante un ímprobo trabajo para intentar atajar los daños. Debían cambiar la Clave de Procesamiento en todos los aparatos reproductor y recuperar la confianza de los usuarios. En el intento, protagonizaron uno de los episodios más estrambóticos en la historia de la propiedad intelectual.

La AACS-LA utilizó la ley DMCA para detener, o al menos intentar frenar, la diseminación de información. Las webs que guardaban claves de volumen, y las que proporcionaban copia de los nuevos programas para reproducir y copiar discos, eran retiradas bajo amenaza de iniciar acciones legales contra los responsables. Hasta cierto punto, es algo que resulta razonable, toda vez que estaban intentando proteger

su negocio contra un ataque.

Pero cuando intentaron prohibir la publicación de la Clave de Procesamiento (`09 F9 11 02 9D 74 E3 5B D8 41 56 C5 63 56 88 C0` en notación hexadecimal), los acontecimientos adquirieron un cariz que solamente puedo intentar describir como una mezcla entre Kafka y los Hermanos Marx. Clave o no clave, se trata de un número (en notación hexadecimal, pero número a fin de cuentas). Resulta absurdo pensar en patentar o prohibir un número, pero eso es exactamente lo que la AACS-LA intentó hacer. El 17 de abril, al tiempo que se anunciaban las "actualizaciones de seguridad" de los reproductores WinDVD y PowerDVD, se enviaron notificaciones a diversas páginas web donde se guardaba una copia de la "clave 09F9".

Para los no-norteamericanos, nos resulta absurdo pensar siquiera que alguien se tome en serio una cosa tan boba como prohibir un número, pero en EEUU la ley DMCA establece penas muy serias para cualquier violación de propiedad intelectual, incluida la posesión de "herramientas para sortear ilegalmente medidas de protección". Algunas páginas de gran tráfico, como la propia Wikipedia, cumplieron con el mandato.

A la postre, sin embargo, la sensatez se impuso. Prohibir un número no solamente es cuestionable desde el punto de vista de la ley, es que resulta imposible en la práctica. Una de las formas más originales de diseminarlo fue mediante códigos de colores. Un color viene representado mediante tres parejas de dígitos hexadecimales. Eso significa que la clave puede indicarse mediante cinco barras de colores (15 dígitos), mas el dígito "C0" añadido". Si usted tiene una pantalla en color, podrá ver a continuación la Clave de Procesamiento codificada en esos cinco colores:

09 F9 11
02 9D 74
E3 5B D8
41 56 C5
63 56 88
C0

Esta representación cromática se hizo famosa con el nombre de "bandera del discurso libre", donde el C0 final se incorporó como "+C0," y representaba la idea de que publicar un número debería ser un "crimen cero"[63]. Ahora el absurdo aumentaba, porque habría también que prohibir cinco colores de la paleta.

La rebelión prendió en la Red, y fue amplificada por digg.com, una web colaborativa donde los lectores pueden proponer noticias, y también votarlas. En un principio, los responsables de Digg decidieron, a pesar de lo absurdo de la medida, atenerse a la ley DMCA y retirar los mensajes y comentarios relacionados con la clave 09F9. Sin embargo, los usuarios no hacían más que enviar noticias sobre la

clave, cada vez con mayor frecuencia e insistencia. Finalmente, el creador de Digg Kevin Rose cedió, y el 1 de mayo de 2007 le dio al público lo que quería:

"*Vosotros preferís ver caer a Digg luchando, antes que arrodillarse ante una gran empresa. Os hemos oído, y con carácter inmediato dejaremos de borrar historias y comentarios que contengan el código… Si perdemos, qué demonios, al menos habremos muerto en el intento*" [64].

Lo que la BBC describió como "una revuelta del siglo 21"[65] adoptó tantas formas como cabían en la imaginación de la gente. La clave se diseminó en camisetas[66], canciones[67], obras de arte gráfico[68], tatuajes[69], gorras, tazas y un sinfín de objetos. La historia de la clave 09F9 ha quedado como ejemplo de cómo la lucha por proteger un derecho legítimo puede llegar a extremos de absurdo.

Mientras la rama de relaciones públicas de la AACS-LA intentaba bailar con la más fea, sus colegas del departamento técnico se pusieron manos a la obra e hicieron lo único que parecía resolver el problema: crear y distribuir una nueva Clave de Procesamiento. El resultado conllevó una modificación del *Media Key Block* a una nueva versión denominada MKBv3 (MKBv1 fue la que utilizaba la Clave de Procesamiento 09F9; hasta donde se sabe, nunca hubo una MKBv2). Esta solución no impediría el copiado y diseminación de las películas ya publicadas, pero podía impedir que los hackers extrajesen las Claves Únicas de Volumen de los futuros estrenos. Era el menor de dos males, comparado con no hacer nada. Eso también resolvería la pesadilla de relaciones públicas, ya que la clave 09F9 quedaría obsoleta.

Sin embargo, seguro que el lector avezado habrá llegado ya a la conclusión lógica: ¿qué impedirá que los hackers extraigan la nueva Clave de Procesamiento? Las técnicas de volcado de memoria y búsqueda paciente que tan buenos resultados les ha dado en el pasado seguían vigentes. Ed Felton, desde su blog *Freedom to Tinker*, profetizaba ya futilidad de la medida[70]:

"*Parece inevitable que los atacantes, en cosa de un mes, tendrán éxito al extraer las claves del nuevo software… yo diría que los atacantes extraerán las claves del nuevo software en unas tres semanas*".

Felten se refería a las nuevas claves insertadas en los programas WinDVD y PowerDVD. Aún faltaban cosa de un mes para que la clave de procesamiento 09F9 se hiciese pública. Pero sus palabras resultaron proféticas. En uno de sus artículos, un internauta apodado BtCB incluyó el día 23 de mayo un número hexadecimal y las palabras "¿cuál es la probabilidad de que se trate de la nueva clave de procesamiento?"[71]. Una semana después, arnezami confirmó que la clave publicada por BtCB es realmente la nueva Clave de Procesamiento[72]:

```
45 5F E1 04 22 CA 29 C4 93 3F 95 05 2B 79 2A B2
```

No se sabe la fecha exacta del descubrimiento, ni quién lo realizó, pero el

programa copiador AnyDVD lanzó una actualización el día 20 de mayo para ofrecer "soporte a la nueva versión de AACS"[73].

Esta nueva derrota representaba un inconveniente catastrófico para la AACS-LA. Incluso una medida tan radical como cambiar la Clave de Procesamiento en todos los reproductores software (y los futuros reproductores hardware) del mundo se mostraba inútil frente a una comunidad de personas anónimas que parecen conocer el sistema mejor que sus propios diseñadores. Programas desprotectores como AnyDVD muestran que cualquiera puede extraer el contenido de una película de alta definición; o más cómodo aún, esperar a que otro lo haga y lo cuelgue en una red p2p.

Lo único que podía hacerse era volver a cambiar la Clave de Procesamiento, y cruzar los dedos esperando que los hackers se cansasen del asunto y se dedicasen a otra cosa. Vana ilusión. El nuevo *Media Key Block* versión (MBKv4) fue anunciado el 7 de septiembre de 2007, y apenas un mes después el programa AnyDVD ya ofrecía soporte para algunos discos protegidos con MBKv4[74]. Un mes después, "algunos" se convirtieron en "todos"[75].

Un nuevo cambio modificó de nuevo el MBK a la versión 8, que comenzó a funcionar en abril de 2008 (no sabemos qué pasó con las versiones 5-7); luego hubo más cambios en algún momento de 2009 (a estas alturas, la AACS-LA no se molestaba siquiera en anunciar sus "actualizaciones"), y como resultado apareció la versión MBK9, seguida prontamente por la MBK10. Todas fueron atacadas y sus Claves de Procesamiento fueron recuperadas. Las incluyo aquí todas, empezando por la primera:

```
MKBv1:
09 F9 11 02 9D 74 E3 5B D8 41 56 C5 63 56 88 C0
MKBv3:
45 5F E1 04 22 CA 29 C4 93 3F 95 05 2B 79 2A B2
MKBv4:
F1 90 A1 E8 17 8D 80 64 34 94 39 4F 80 31 D9 C8
MKBv8:
7A 5F 8A 09 F8 33 F7 22 1B D4 1F A6 4C 9C 79 33
MKBv9:
C8 72 94 CE 84 F9 CC EB 59 84 B5 47 EE C1 8D 66
MKBv10:
45 2F 6E 40 3C DF 10 71 4E 41 DF AA 25 7D 31 3F
```

A partir de este punto, la situación degenera en una carrera predecible y hasta cierto punto aburrida. La Clave de Procesamiento deja de renovarse, o al menos los hackers dejan de publicarla en Internet, así que podemos suponer que la AACS-LA se limita a cambiar el MKB por otros procedimientos Comienzan a aparecer nuevas versiones de MKB cada vez con mayor frecuencia, y no siempre consecutivas; y

continúan apareciendo Claves Únicas de Volumen para todos los estrenos de cine.

A estas alturas, el campo de batalla del mercado había cambiado radicalmente. La guerra de los formatos de alta definición acabó con la derrota del sistema HD DVD. Toshiba, su principal valedor, abandonó en febrero de 2008, y el sistema Blu-ray quedaba como vencedor único en el panorama de la alta definición. Hace falta un nuevo paradigma de seguridad, y este es el momento en que aparece un nuevo esquema de seguridad. AACS seguirá siendo usado, pero será complementado por una capa adicional de seguridad.

El nombre de la nueva protección es BD+ y fue desarrollado por la empresa Cryptography Research. Diré simplemente que se trata de lo que los informáticos llaman una "máquina virtual," una especie de sistema operativo dentro de otro sistema operativo. El concepto de máquina virtual es muy útil para poder, por ejemplo, desarrollar o ejecutar programas Windows en un ordenador que funcione bajo Linux. Es una forma de simular un sistema operativo sin tener que instalarlo.

Como recordará el lector, la vulnerabilidad fundamental del esquema AACS consistía en que se podían extraer las claves gracias a programas que realizan volcados de memoria. Eso ya no será posible, ya que BD+ crea algo parecido a un sistema operativo propio. Como un pelotón de comandos que asegura una playa antes de un desembarco, la máquina virtual creada por BD+ examina los alrededores, comprueba la integridad de las claves, ejecuta ciertos programas para asegurarse de que no hay enemigo a la vista y, cuando está satisfecho, da la luz verde a las fuerzas de invasión, en este caso al proceso de descifrado de audio y video. Esto funciona tanto en hardware como en software.

Los primeros títulos protegidos con BD+ salieron al mercado en octubre de 2007. En aquellos momentos, la AACS-LA pugnaba por intentar cerrar la brecha de seguridad, y ya había visto morder el polvo sus MKB v3 y v4. Eran los últimos meses del formato HD DVD, y aunque su caída no fue cosa de un día o de un solo factor, hay que resaltar el hecho: HD DVD no disponía del sistema BD+ y Blu-ray, sí.

La *Blu-ray Disc Association* no tuvo mucho tiempo para disfrutar mientras veía pasar el cadáver de su enemigo. El 19 de marzo de 2008, justamente un día después de la muerte oficial del formato HD DVD[76], AnyDVD tenía ya capacidad completa para eliminar la protección de los discos Blu-ray, incluyendo la capa de BD+[77]. En repetidas ocasiones, el formato DB+ fue modificado para adaptarse al último ataque, pero caía en el siguiente. Y había más fabricantes de software: en 2010 entraron en liza DVDFab[78] y Pavtube[79], con programas capaces de saltarse las protecciones AACS y BD+ Para resumir: una combinación de hackers, piratas e internautas diversos habían logrado obtener todas las claves necesarias para saltarse las protecciones AACS, y también para evitar los controles adicionales de BD+.

Había una protección adicional al final de la línea. Los críticos siempre habían

señalado que, por muy buenas que sean las medidas de protección, al final el contenido de la película es enviado al televisor o monitor. Eso significa que, si elimino el televisor y lo sustituyo por una grabadora, puedo obtener una copia sin cifrar. Es una vulnerabilidad, en efecto, y para evitarla Intel Corporation diseñó un sistema de protección llamado HDCP (*High-bandwidth Digital Content Protection*). Se trata de una capa de cifrado que protege las comunicaciones entre el reproductor y el televisor. Ambos elementos tienen que estar diseñados con ese estándar si queremos disfrutar de contenido en alta definición.

Por supuesto, nada nos impide conectar nuestro reproductor Blu-ray a un televisor o monitor sin HDCP, pero solamente obtendremos calidad estándar similar a la de un DVD. En Europa, cualquier elemento de hardware que pretenda lucir la etiqueta "*HD ready*" deberá incorporar HDCP de forma obligatoria y, por supuesto, pagar la correspondiente cuota[80]. Huyendo de la seguridad mediante oscuridad, las especificaciones HDCP se hicieron públicas[81], y hay incluso código fuente libre para replicarlo[82]. Si usted tiene un dispositivo con conexión HDMI o DVI, sepa que HDCP está ahí dentro, haciendo guardia en la garita.

HDCP tiene dos funciones: autenticación y cifrado. Como de costumbre, el primer paso es asegurarse de que ambas partes, a las que llamaremos desde ahora E (emisor, por ejemplo un reproductor Blu-ray) y R (receptor, por ejemplo un televisor) están autorizadas para intercambiar información; en caso afirmativo, el segundo paso consistirá en acordar una clave de cifrado. Los algoritmos son relativamente sencillos, así que vamos a examinarlos aquí.

En lo alto de la jerarquía de claves de HDCP tenemos la **Clave Maestra**, fuertemente protegida por la autoridad central. A partir de esa clave maestra se creará una clave privada y una pública para cada aparato, sea emisor o receptor. La clave privada es en realidad un "paquete" de 40 claves, de 56 bits de longitud cada una, y que colectivamente reciben el nombre de **Claves Privadas de Dispositivo**. A partir de ahí, cada aparato toma un bit de cada una de esas claves, y con ellos construye una clave pública que recibe el nombre de **Vector de Selección de Clave** (KSV). La KSV tiene la particularidad de que, de sus 40 bits, veinte son ceros y otros veinte son unos. Además de eso, cada aparato tiene su propia clave privada, también de 40 bits. A partir de ahora, llamaremos *U* a la clave privada y *V* a la clave pública, y le añadiremos una letra adicional para identificar al emisor *e* y al receptor *r*.

El mecanismo de autenticación comienza en el emisor, que tiene claves privada y pública *Ue*, *Ve*, respectivamente. Lo primero que hace el emisor es escoger un número aleatorio, digamos *N*, y se lo envía al receptor junto con su clave pública *Ve*. El receptor toma la clave pública que acaba de recibir y la multiplica por su propia clave privada, obteniendo la cantidad *K' = Ve*Ur*. Puesto que cada clave tiene 40 bits, el producto *Ve*Ur* debería tener 1600 bits de longitud. En nuestro caso no es así,

porque se desea que el producto tenga 56 bits, así que se utiliza una operación llamada “producto vectorial en el anillo Z/2^56 Z”.

En cualquier caso, cuando el receptor calcula *K’*, no se lo envía al emisor. En su lugar, efectúa una operación hash, que para nuestros propósitos es una función criptográficamente segura en la que introducimos dos números (*Ve, N*) y obtenemos un tercer número de 16 bits que llamaremos *R’*. Matemáticamente lo podemos escribir como *R’= h(Ve, N)*. Ahora el receptor envía al emisor el paquete de dos números formado por *R’* y *Vr*. Cuando el emisor recibe dicho paquete, calcula *K=Vr*Ue* y efectúa la operación *R=h(K,N)*. Si *R=R’* todo ha ido bien, y *K=K’* es la clave que acaban de compartir.

Para que quede más claro:

```
EMISOR              RECEPTOR
(Ue,Ve)             (Ur,Vr)
----- (Ve, N) ------------>
               K’ = Ve*Ur
               R’ = h(K’,N)
<---------- (Vr, R’) -----
K = Vr*Ue
R = h(K,N)
¿R=R’?   SI --> Aceptar
         NO --> Rechazar
```

El hecho de que *R = R’* indica dos cosas. En primer lugar, que ambas partes han usado el mismo número aleatorio *N*. Ese número en sí no significa nada, pero indica que el proceso se ha llevado a cabo ahora. Sin él, un atacante podría grabar la comunicación que E y R mantuvieron la semana pasada. En segundo lugar, nos hemos asegurado de que *K’* es igual a *K*, y esa será la clave compartida. A primera vista no parece evidente que ambas sean iguales, pero eso es porque me he callado un pequeño detalle: las claves de ambos dispositivos se calcularon de tal forma que siempre se cumpla que *Ve*Ur=Vr*Ue*. Si el emisor y el receptor fueran personas, ni siquiera bajo tortura podrían confesar por qué es así, ya que ellos mismo no lo sabrían. Basta con que los creadores originales de las claves lo sepan, y con eso basta. De ese modo, no solamente emisor y receptor han comprobado que quien está al otro lado es un interlocutor autorizada, sino que al mismo tiempo han acordado la clave *K* con la que cifrarán la información que a partir de ahora se intercambien.

El algoritmo de cifrado es una combinación de tres elementos. En primer lugar, tenemos un conjunto de cuatro Registros de Cambio por Retroalimentación Lineal (LFSR). Ya vimos el funcionamiento de los LFSR en el apartado de los DVD, así que no lo repetiremos aquí. Lo importante es que estos LFSR generan un flujo pseudoaleatorio, es decir, una cadena de bits que no son aleatorios pero lo parecen.

Esa cadena de bits actuarán como clave en un algoritmo de cifra en bloque. Finalmente, el resultado es mezclado en una tercera fase para, finalmente, producir un flujo de datos cifrado.

Es una lástima que tanto trabajo al final no sirva para nada. Porque, como quizá habrán adivinado a estas alturas, del dicho teórico al hecho práctico hay mucho trecho. El esquema HDPC adolece de una grave vulnerabilidad, descubierta en 2001 por un grupo de criptoanalistas norteamericanos[83]. Lo más curioso del caso es que no entraron en absoluto en los detalles técnicos del proceso de cifrado, nada de jugar con los LFSR o con la función de compresión. En lugar de eso, se centraron en el proceso de intercambio de claves y se dieron cuenta de que es lineal. Y eso es malo. Les explicaré por qué.

La idea básica en cualquier autenticación emisor-receptor es, recordarán, que los pares de claves cumplan la condición de que *Ue*Vr=Ur*Ve=K*, donde *K* es la clave compartida. Para entender el problema, vamos a imaginar que tengo dos aparatos emisores, a los que llamaré X e Y. Yo digo que puedo usarlos para "construir" otro aparato Z cuya clave sea la suma de las dos anteriores, esto es: Uz=Ux+Uy, Vz=Vx+Vy.

¿Qué pasará si intento comunicar Z con un receptor R? Veámoslo. Cada uno de los dos aparatos (Z, R) calculará sus claves respectivas *K* y *K'* de la siguiente forma:

```
K' = Vz*Ur
   = (Vx*Ur) + (Vy*Ur)
   = K1 + K2
K = Vr*Uz
  = (Vr*Ux) + (Vr*Uy)
  = K3 + K4
```

Ahora bien, como X y R tienen claves válidas, resulta que *Vx*Ur =Vr*Ux*, es decir, *K1=K3*. Por el mismo motivo, al tener Y y R claves válidas, tenemos que *K2=K4*. Conclusión: *K=K'*, lo que significa que cualquier combinación lineal de claves válidas será también una clave válida. Un atacante puede sacar partido inteligente de este hecho, ya que solamente necesita cuarenta pares de claves (pública-privada) para reconstruir la Clave Maestra, y a partir de ahí el atacante puede hacer lo que se le antoje: general claves falsas, descifrar archivos, clonar reproductores.

¿Y de dónde sacará un atacante esos cuarenta pares de claves? Los autores sugieren técnicas como ingeniería inversa, pero hay una forma más fácil: comprándolas legalmente a la entidad que otorga las licencias.

No fue el único ataque de estas características. Un mes antes, un criptógrafo holandés llamado Neils Ferguson anunció que había conseguido romper el algoritmo HDPC, consiguiendo un nivel de control similar. Desafortunadamente, las

consecuencias legales derivadas de la ley DMCA le impidieron publicar sus resultados[84].

Los ataques de 2001 no parecen haber sido aprovechados por ningún hacker. De hecho, da la impresión de que fuesen ignorados. Quizá sea porque HDPC era un sistema de protección secundario, comparado con los pesos pesados del barrio BD+ y AACS. Pero cuando esos pesos pesados cayeron, HDPC se convirtió en una especie de última línea de defensa, y fue entonces cuando se ganó la atención del respetable. El propio Neils Ferguson lo profetizó en 2001:

[La DMCA] me impide publicar mi artículo ahora, pero algún día, alguien, en algún lugar, duplicará mis resultados. Esa persona podrá decidirse a publicar la clave maestra HDPC en Internet. En lugar de arreglar HDPC ahora antes de ser desplegado a gran escala, la industria se enfrentará a los gastos de insertar HDPC en cada dispositivo, sólo para que luego resulte inútil. La DMCA acabará costándole dinero a la industria. Adivinen quién acabará pagando al final.

Indudablemente, el pagano acabará siendo el consumidor. Durante los diez años que ha estado utilizándose el sistema HDPC, la entidad licenciadora ingresó incontables millones de euros gracias a una situación de monopolio *de facto*. Nadie que quiera vender un dispositivo de alta definición podrá dejar de pasar por caja.

Esa situación cambió radicalmente en 2010. El 13 de septiembre de ese año, apareció una página en pastebin.com con el título "*¿Es auténtica la clave HDPC filtrada?*" La web ya no existe, pero existen copias[85]. En ella, se incluye nada menos que la Clave Maestra, el "anillo para controlarlos a todos". Si esa clave fuese la auténtica, ahora cualquiera podría crear pares de clave pública/privada para cualquier dispositivo.

Por supuesto, el hecho de que alguien publique una ristra de números no significa que realmente sea la Clave Maestra. Pero en este punto la historia se repite. En el año 1917, los ingleses consiguieron descifrar el Telegrama Zimmermann, que contenía una información capaz de alterar el curso de la guerra. Surgió entonces la duda sobre si el telegrama era auténtico, o si se trataba de una hábil falsificación llevada a cabo por Inglaterra para arrastrar a los Estados Unidos a la guerra. Arthur Zimmermann, ministro alemán de Asuntos Exteriores, disipó las dudas de los incrédulos al reconocer en una rueda de prensa que el telegrama era cierto.

En este caso, el papel de Zimmerman fue representado por Tom Waldrop, portavoz de Intel, quien dos días después de la revelación confirmó que la clave era auténtica[86]. Quizá fuese un intento por parte de Intel de paliar los daños, ya que cualquiera podría tomar la Clave Maestra filtrada y probarla. Un criptógrafo podría verse frenado por las consecuencias legales, pero en la Red hay mucha gente con menos escrúpulos. La identidad de la persona que filtró la Clave Maestra permanece desconocida a día de hoy.

La Clave Maestra es una matriz de 40*40=1600 elementos, y cada uno de ellos es un número de 56 bits (o, lo que es lo mismo, 16 caracteres hexadecimales). Como es bastante voluminosa, puede usted consultarla en la referencia[85b], que también incluye las instrucciones de uso. Lo primero que hacemos es escoger un vector de selección de clave (KSV), que es como se llama aquí a la clave pública. Ese vector va a tener veinte unos y veinte ceros. A continuación, nos vamos la matriz que forma la Clave Maestra, y escoger las filas que nos indican los ceros del KSV. Para entenderlo con un ejemplo, supongamos que este es nuestro KSV:

```
10100010101110001001101011100001110101 01
```

Tenemos que leer comenzando por el bit más bajo, es decir, de derecha a izquierda. Encontramos unos en las posiciones 1, 3, 5, 7, 8, 9… Eso significa que tenemos que tomar las filas 1, 3, 5, 7, 8, 9… de la matriz formada por la Clave Maestra, y sumarlas módulo 2^56 (eso significa que, si la suma nos da un número mayor que 2^56, le restamos 2^56 tantas veces como sea necesario para obtener un número menor que 2^56). El resultado es la clave privada. Así de sencillo. Bueno, salvo por un detalle. Lo que hemos construido así es la clave privada para un emisor, lo que en las especificaciones se llama una fuente. Para construir la clave privada de un sumidero (un receptor, como un monitor de TV), hacemos lo mismo, pero tomando la matriz traspuesta.

Hasta cierto punto puede parecer que romper HDCP no es relevante, ya que los ataques contra AACS y BD+ pusieron a disposición del público reproductores software libres, copiadores y películas libres de cifrado. Con todo, conlleva interesantes aplicaciones para el usuario. HDCP se utiliza también para proteger la señal de los dispositivos llamados “repetidores,” entre los que se incluyen los aparatos que reciben señales de video en descarga directa (*streaming*). Con el uso de la Clave Maestra filtrada, cualquier empresa puede en principio construir un dispositivo adaptador que, aplicado a la salida del reproductor (por ejemplo, un video Blu-ray o una PS3) permite visualizar el contenido en cualquier televisor, convertido así en un receptor “Full HD”.

La *Digital Content Protection LLC* (DCP), entidad que otorga las licencias HDCP, tomó medidas al respecto. Para no cometer los mismos errores de relaciones públicas que vimos en el caso AACS, se abstuvieron de perseguir a los difusores de la Clave Maestra. En su lugar, remodelaron el sistema de seguridad. La fase de autenticación está ahora basado en un sistema de criptografía de clave pública, con claves RSA de 1024. El protocolo SHA-1 fue escogido para las operaciones de firma digital. La generación de números aleatorios, parte importante en el proceso, se confió al algoritmo AES, que también sería utilizado para el cifrado de los contenidos. Como ven, copiaron a AACS en lo bueno: la elección de buenos

algoritmos. Se acabó eso de utilizar criptografía hecha en casa.

Un hecho notable es que las contramedidas de protección del nuevo HDCP (versión 2.0 y posteriores) no fueron hechas precipitadamente, como suele ocurrir cuando alguien rompe el sistema. Las especificaciones 2.0 tienen fecha de octubre de 2008, dos años antes de que la Clave Maestra fuese filtrada y se hiciese necesaria una sustitución. Mi hipótesis es que los técnicos de la DCP tomaron buena nota del ataque de 2001, y a lo largo de varios años fueron diseñando un sistema más robusto, que entre otras cosas podía incluir opciones para proteger transmisiones inalámbricas (entre dos televisores ubicados en habitaciones distintas, por ejemplo, o las provenientes de un router que retransmita contenidos de televisión), así como comunicaciones por Internet.

A pesar de ello, existe un grave fallo de seguridad. Un grupo de investigadores de la Universidad Ruhr-Bochum alemana, dirigidos por Tim Güneysen, anunciaron en noviembre de 2011 que habían abierto brecha en la protección HDCP 1.3 mediante un sistema distinto. En lugar de utilizar la Clave Maestra, interpusieron un conjunto de chips especiales FPGA entre un reproductor Blu-ray y un monitor. Interceptando y modificando los mensajes que se intercambiaban, consiguieron descifrar las comunicaciones, en este caso una película[87].

La principal aplicación del estudio de Güneysen era la protección de comunicaciones en el campo militar y otros entornos de alta seguridad. Pero resulta que el ataque puede ampliarse a los reproductores HDCP de todo tipo, incluidos los que cumplen las nuevas especificaciones 2.1 (sucesoras de las 2.0). El problema está en la compatibilidad. Si las especificaciones 2.1 se impusiesen tal cual, ningún dispositivo que utilizase las versiones anteriores podría funcionar. Nuestro nuevo reproductor Blu-ray con HDCP 2.1 no podría verse en un televisor de alta definición con HDCP 1.4. Por ese motivo, los nuevos sistemas han de ser compatibles con los algunos, lo que suele llamarse “compatibilidad hacia atrás”. Mientras exista esa compatibilidad los dispositivos antiguos y los nuevos podrán conectarse sin problemas, pero por el mismo motivo el sistema seguirá siendo vulnerable a los fallos que tumbaron la versión HDCP 1.4.

Las perspectivas de supervivencia de HDCP 2.0 (ahora 2.1) son, por el momento buenas. El sistema parece robusto, y lo será más cuando se resuelva el agujero de seguridad que representa la compatibilidad hacia atrás. Pero, visto lo visto sobre los demás sistemas de seguridad que han intentado proteger los contenidos de alta definición hasta el día de hoy, no me atrevería a asegurar nada.

4) LA RESPUESTA: CÓMO PROTEGERNOS

Los usuarios de productos audiovisuales en muchos países del mundo se enfrentan al mismo problema: por un lado, tienen ciertos derechos sobre los productos que adquieren; por otro, ejercer dichos derechos pasa por utilizar herramientas que están prohibidas por la ley. En pocos sitios es más evidente el problema como en Estados Unidos, donde un comprador legalmente no puede hacer una copia de seguridad de su propio DVD. En la actualidad, el gobierno federal de EEUU está considerando la posibilidad de relajar la draconiana ley DMCA para permitir la copia en casos legales como copia privada o "fair use," derecho de cita, crítica y comentario, usos educativos y documentales, aplicaciones no comerciales, copia de seguridad, etc[88].

La legislación española actual, sobre el papel, parece prohibir todo tipo de copia de material audiovisual. El artículo 270.3 del Código Penal español castiga con hasta dos años de cárcel a quien *fabrique, importe, ponga en circulación o tenga cualquier medio específicamente destinado a facilitar la supresión no autorizada o la neutralización de cualquier dispositivo técnico que se haya utilizado para proteger programas de ordenador o cualquiera de las otras obras, interpretaciones o ejecuciones en los términos previstos en el apartado 1 de este artículo*. Este artículo, aprobado en medio de una gran polémica entre internautas y gestores de derechos de propiedad intelectual, sanciona todo tipo de conductas, desde la fabricación de DVDs pirata a la mera posesión de un programa "ripper" para extraer música o video de un disco.

Su ámbito era tan extenso que la Fiscalía General del Estado tuvo que aclararla en su Circular 1/2006[89], que vino motivada por la necesidad de establecer criterios unificados de interpretación y actuación, así como de establecer límites a la persecución penal de los delitos a que se refiere el Código Penal en lo relativo a propiedad intelectual e industrial. Es evidente que no se puede encarcelar a diez millones de personas por tener copias de una película o un programa de ordenador. Y, por supuesto, había que aclarar una contradicción que los detractores del canon por copia privada habían denunciado muchas veces: no tiene ningún sentido que me dejen hacer una copia privada, y al mismo tiempo me impidan usar los programas informáticos necesarios para hacerla:

"*Cabe plantearse el problema de legitimidad de la conducta de la persona que habiendo adquirido un original de un CD o un DVD, protegido por dispositivo técnico para evitar su reproducción esté en posesión de un medio que sea apto para eliminar o neutralizar la protección, si lo hace en el ejercicio de una copia privada… teniendo en cuenta que el derecho de copia privada da lugar a la remuneración equitativa contemplada en el artículo 25 de la Ley de Propiedad Intelectual.*"

La mera existencia de esos programas refleja según la Fiscalía "*la inutilidad de las barreras de protección que [los titulares de derechos] colocan*," algo que hemos visto a lo largo de estas páginas. La Circular 1/2006 concluye, con buen criterio, que la mera tenencia de tales medios técnicos no se considerará ilícito penal, salvo prueba en contrario; y que sólo serán sancionables si la elusión de las medidas técnicas anticopia se realiza con fines comerciales. Nadie irá a la cárcel en España por instalar una copia de DVD Decrypter para uso particular[90]. Caso muy distinto sería el de un pirata profesional que utilizase dispositivos de desprotección para, una vez obtenida la película o desprotegido el juego, proceder a su venta; en cuyo caso se podría considerar un agravante, puesto que su uso sería sancionable como acto preparatorio para actividades lesivas de los derechos de propiedad intelectual ajenos con ánimo de lucro.

Un área todavía gris corresponde a los vendedores de productos informáticos que proporcionan a sus clientes dispositivos hardware para saltarse las protecciones de los fabricantes. Entre esos productos se encuentran cartuchos adaptadores para consolas tanto portátiles (tipo Nintendo) como de sobremesa (Xbox360, PlayStation). En ese caso, la decisión del juez depende de las circunstancias del caso.

Por ejemplo, en junio de 2011 el Juzgado de lo Penal nº 1 de Avilés absolvió a los responsables de una tienda de informática que importaban y vendían cartuchos para consolas Nintendo DS. Entre otras cosas, el juez entendió que los cartuchos difícilmente podían desproteger obras audiovisuales o programas de ordenador, puesto que carecían de software instalado. También se consideró el hecho de que los cartuchos pueden emplearse en otros usos legales, como guardar y reproducir música o fotografías propias.

En términos similares se pronunció la Audiencia Provincial de Valencia en un auto de marzo de 2008. En este caso el sistema en litigio era un chip que desprotegía consolas de videojuego, lo que hacía posible utilizar juegos pirata. Los jueces entendieron que:

"*Los chips que se instalan o se pueden instalar en las videoconsolas de autos, pueden servir, desde luego, como dispositivo tendente a desprotegerlas para permitir utilizar juegos no originales, pero también, para permitir la ejecución de juegos originales de otras zonas y para convertir la consola en un ordenador personal apto para realizar múltiples tareas absolutamente lícitas, como pueda ser el manejo de fotografías, ejecutar juegos de libre distribución no diseñados para consola, escuchar música, etc. No se cumpliría, por tanto, el requisito de la exclusiva o específica destinación a la supresión o neutralización de dispositivos de protección de las consolas, y en este sentido el razonamiento de la instructora no resulta desacertado para este Tribunal*".

En consecuencia, confirmaron el dictamen del Juzgado de Instrucción nº 8 de

Valencia que sobreseyó el caso. Ello no obstante, las circunstancias de cada caso en particular imponen un análisis particularizado. Fue el caso, por ejemplo, de la tienda "Juega 2" de Jerez de la Frontera. Denunciado su dueño por vender chips para desproteger videoconsolas, el tribunal estableció acertadamente que

"*obviamente pueden suscitarse problemas de prueba en la concurrencia del requisito de que los medios que pueden suprimir o neutralizar los dispositivos técnicos de protección (generalmente algún programa informático), estén específicamente destinados a esa finalidad, lo que habrá de deducirse, de la propia naturaleza o funcionalidad del medio en cuestión y de las circunstancias de su intervención*".

En este caso, el juez condenó al acusado al entender que en este caso no concurría la eximente de que los chips podían ser usados para otros fines, ya que:

"*La defensa ha señalado que dichos programas no solo permiten la utilización de las video consolas para ver juegos no autorizados sino también para que las mismas video consolas hagan funciones de ordenador personal con software o programación libre. Sin embargo tal hipótesis es poco probable al afectar a un establecimiento que se dirige al área específica de las video consolas y casi no comercializa otros productos distintos y más habituales a otras tiendas de informática, donde se venden ordenadores personales, portátiles, distintos programas etc … la única explicación lógica es que el acusado se dedicaba a la instalación de dichos mecanismos o a su venta al público*".

Basten estos ejemplos para resaltar que los tribunales españoles han emitido dictámenes tanto absolutorios como condenatorios en diversas ocasiones. En todos los casos, existe el componente lucrativo, exigencia previa necesaria para aplicar el Código Penal. En cuanto al uso privado de programas de ordenador para desprotección de material audiovisual, sencillamente no encuentro sentencias judiciales al respecto. Quizá sea porque, como dijo la Audiencia Provincial de Madrid en sentencia de junio de 2012:

"*Aunque en el art. 270.3 no se exige la concurrencia expresa de ánimo de lucro, este debe considerarse implícito en todos los supuestos típicos de delitos contra la propiedad intelectual como diferenciadores de los ilícitos civiles*".

Lo que, en los casos de uso privado, evidentemente no sucede. Por lo general, además, las condenas en las que el artículo 270.3 se aplica suelen ser aquellas en que también se ha considerado violado el artículo 270.1, que trata de las violaciones de propiedad intelectual con ánimo de lucro y daño a terceros.

Por tanto, descanse usted tranquilo, señor ciudadano de a pie. Si usted utiliza programas de desprotección de ordenador, se ha comprado un DVD pirata, o se ha bajado música o películas de Internet, no tiene nada que temer desde el punto de vista penal. En cuanto a ilícitos civiles, técnicamente sí pueden perseguirle por ese motivo,

pero la cuantía del bien en litigio sería tan pequeña que dudo siquiera que un juez lo admitiese a trámite, salvo que usted tuviese cien mil películas en su ordenador. En tales casos, además, no se admiten pruebas habituales en procesos penales, como los registros de ordenadores o interceptación de comunicaciones. Es matar moscas a cañonazos, y los jueces no se dedican a esas cosas.

Eso sí, amigo, si piensa usted vivir de ello, sea vendiendo chips de desprotección o decodificadores de TV pirateados, le recomiendo que comience consultando a un buen abogado. Que una cosa es el uso personal y otra muy distinta pretender vivir a costa del trabajo ajeno. Eso incluye al listillo que desprotege los canales de TV de pago para compartirlos por la antena comunitaria: por muy loable y generoso que parezca su ofrecimiento, está usted violando la ley. Avisado queda, como digo siempre.

TARJETAS DE CRÉDITO

El mundo de las tarjetas de crédito es seguro. En términos relativos, claro. Pregunten a cualquier banquero (o bancario, como prefieren que los llamemos ahora), y le contará una historia de horror tras otra; suponiendo que decida contarle la verdad, claro. Hay fraude, robos, suplantación de identidad, de todo. En general, la profesión bancaria ha aceptado como hecho inevitable la existencia de cierto grado de fraude en el uso de tarjetas de crédito. Eso no significa, por supuesto, que se crucen de brazos. Al contrario, incorporan medidas de seguridad de todo tipo para evitar fraudes o hurtos. Algunos de ellos utilizan criptografía, y esto es lo que aquí nos interesa.

En las siguientes páginas, nos detendremos brevemente en algunos de los errores cometidos por la banca en su búsqueda de la perfección técnica. Algunos no tienen repercusión práctica, pero aun así sorprende la imaginación que muestran algunos atacantes. En este punto, debo aclarar que con “atacantes” no me refiero al ladrón hacker de guante blanco, o al de pasamontañas negro. Muchas veces, los errores son detectados por matemáticos, criptógrafos o informáticos. Es importante corregirlos, porque el próximo en darse cuenta de que existe un agujero de seguridad puede no ser tan honrado.

En este campo, Reino Unido ha constituido un excelente campo de pruebas para los sistemas de tarjetas bancarias. Su sector financiero es uno de los puntales económicos del país, y precisamente en una de sus mejores universidades se han descubierto algunos de los más eficaces ataques contra la seguridad de las tarjetas. Por ese motivo, este tema me ha quedado bastante británico. Pero no tema, porque al final haremos un amplio repaso a la situación en España.

Sea el país en que nos encontremos, siempre es bueno aprender, sobre todo si nos jugamos la cartera. Si le parece bien, comenzaremos con un pequeño ataque oscuro y poco conocido, pero que personalmente me encanta porque muestra hasta qué punto puede llegar la imaginación de la gente. Un pequeño sistema, diseñado inicialmente para hacerle más fácil la vida al usuario, puede convertirse en una vía de acceso en manos de un atacante inteligente. Prepárese a ser decimalizado, estimado lector.

1) EL ATAQUE DE DECIMALIZACIÓN

En febrero de 2003, los investigadores Mike Bond y Piotr Zielinski, del Laboratorio de Informática de la Universidad de Cambridge, publicaron un informe sobre una técnica a la que denominaron “ataque de decimalización,” que aprovechaba una vulnerabilidad en el sistema de cajeros automáticos. Se trata de un pequeño detalle, de esos que nadie se plantea, pero que puede proporcionar una vía de ataque fácil[1].

Los cajeros automáticos modernos descienden del modelo IBM 3624, introducidos por vez primera en los años ochenta. De ellos se han heredado muchas características de los sistemas actuales, como el método de uso de los códigos de identificación personal (PIN). En principio, podríamos pensar que el PIN es transmitido a alguna base de datos para comprobar si es válido, y en caso afirmativo darnos acceso. Pero eso requiere acceso continuo a la red, y si las líneas no funcionan no hay acceso al servidor central y no se podría entregar el dinero al cliente.

La solución pasa por que el propio cajero automático efectúe las comprobaciones. El cajero tiene un módulo de seguridad de hardware donde se almacena una clave *K*. Cuando usted introduce su tarjeta, el sistema lee el número de 16 cifras que aparece grabado en la superficie, llamémosle *N*. Lo que hace el cajero es cifrar *N* mediante el algoritmo DES, usando con *K* como clave. El resultado se convierte a un número de cuatro dígitos, y ese es sencillamente el PIN original.

¿Por qué cuatro dígitos? Porque son fáciles de recordar. John Shepherd-Barrow, a quien se atribuye la invención del cajero automático, afirmó en una entrevista a la BBC que se le ocurrió usar un número como identificación personal, en principio de seis dígitos; sin embargo, cuando lo comprobó con su esposa, ésta era incapaz de recordar más de cuatro[2]. De ese modo nació la moda de los números PIN de cuatro dígitos.

Volviendo a nuestro cajero automático, el número identificador se calcula como PIN=DES(*K*,*N*). Si esa cantidad coincide con el número de cuatro dígitos que ha introducido el cliente, entonces se da luz verde a la transacción. De otro modo, habrá que intentarlo de nuevo. Para evitar que un tenaz ladrón de tarjetas se pase la noche en el cajero intentando los diez mil números PIN posibles, el cajero se traga la tarjeta tras el tercer intento fallido. Puede que usted tenga un mal día y falle las tres veces, pero es mejor perder unos minutos hablando con el director de la sucursal que arriesgarse a que un amigo de lo ajeno dedique su mucho tiempo libre a desvalijarle.

Como ya sabrán ustedes, los números PIN contienen dígitos decimales, entre 0 y 9. Parece algo lógico: nosotros, los humanos, leemos el número 8995 y lo traducimos mentalmente como “ocho millares, nueve centenas, nueve decenas y cinco unidades”. O lo que es lo mismo:

```
8995 = 8*10*10*10
     + 9*10*10
     + 9*10
     + 5
```

Para ser más preciso, es lo que hacen los niños en la escuela. Al llegar a la edad adulta, estamos tan acostumbrados a la notación decimal que la consideramos algo natural.

Sin embargo, al poner en marcha el algoritmo de cifrado DES lo que se obtiene es una cadena de 64 bits o lo que es lo mismo, un número de cuatro dígitos hexadecimales. En el sistema hexadecimal, los dígitos van de 0 a 9 y de A a F, donde se hace la siguiente conversión: A=10, B=11, C=12, D=13, E=14, F=15. Este sistema es mucho más cómodo para los ordenadores, ya que utiliza base 16, que es a su vez un múltiplo de 2. En este sistema, el número hexadecimal 8995 tendría este significado:

```
8995 = 8*16*16*16 + 9*16*16
     + 9*16 + 5
     = 32768 + 2304 + 144 + 5
     = 35221
```

Eso hace más difícil el uso del PIN para los humanos, ya que tendríamos que recordar un “número” formado por cuatro dígitos hexadecimales, o bien un PIN de cinco dígitos decimales que no fuese superior a 65535, que es el mayor número que podemos representar con cuatro dígitos hexadecimales.

Los diseñadores del sistema de cajeros automáticos cayeron en la cuenta de que el usuario medio quiere que le pongan las cosas fáciles, así que idearon un procedimiento para simplificar el problema. En vez de obligar al usuario a “traducir” de decimal a hexadecimal, hagamos que el cajero haga la traducción inversa.

La solución es la conocida como “tabla de decimalización,” que no es más que un nombre rimbombante para convertir un número hexadecimal en otro decimal, y que el resultado siga teniendo cuatro dígitos. La tabla de decimalización habitual efectúa la siguiente conversión

```
Entrada:   0123456789ABCDEF
Salida:    0123456789012345
```

De esa forma tan sencilla, el número 3F7A se convierte en 3570. Como verá, se limitan a transformar los dígitos (A B C D E F) en (0 1 2 3 4 5) respectivamente. En principio, puede utilizarse cualquier tabla de decimalización, pero los diseñadores se limitaron a echar mano de la más sencilla. Ahora bien, y este es el detalle, puesto que la tabla de decimalización no forma parte del sistema de seguridad, nadie pensó en

almacenarla en el módulo seguro del cajero o en protegerla de alguna otra forma. Eso lo convierte en un elemento que se puede modificar impunemente.

Los investigadores de Cambridge lo notaron y aprovecharon para reformular el viejo juego *Master Mind*. Recordará usted ese famoso juego de mesa. El "defensor" escoge una combinación de cuatro fichas de color. El "atacante" intenta adivinar esa combinación mediante un sistema de ensayo y error. Tras cada intento, el defensor le indicará si ha acertado el color de alguna ficha, y si la ficha que ha acertado se encuentra en la posición adecuada. El atacante gana si consigue reproducir el patrón de fichas del defensor antes de un número fijado de intentos.

Ahora prepárese usted, porque vamos a jugar al *Master Mind*, versión cajero automático. Vamos a partir del supuesto de que podemos escoger la tabla de decimalización que queramos, y de momento no tendremos problema con el número de PINs que podemos probar. Supongamos, por fijar conceptos, que el PIN auténtico es 3F7A, que para nosotros se convertiría en 3570. Nosotros, atacantes, no lo sabemos, así que vamos a probar con 0000.

Supongamos que tomamos esta tabla para empezar:

```
Entrada:    0123456789ABCDEF
Salida:     0010000000001000
```

Esta tabla convierte todos los dígitos en cero, salvo si aparece un 2, en cuyo caso lo transforma en el número uno; también lo hace si aparece un C, que con la anterior tabla se convertía en 2. Al calcular el PIN hexadecimal 3F7A, el sistema lo convierte en el número "decimalizado" 0000. A continuación, lo compara con el que hemos introducido nosotros. Evidentemente 0000 es igual a 0000, así que nuestro PIN ha sido aceptado, y así nos lo comunica el cajero.

Ya sabemos que nuestro PIN 0000 no vale, lo que significa que no hay ningún dos en el resultado decimalizado. Esta tabla de decimalización no es más que una forma de preguntar al sistema "¿aparece el número 2 en el PIN?" Si el sistema me da acceso, es señal de que el número dos no aparece.

Ahora cambiemos la tabla de decimalización a:

```
Entrada:    0123456789ABCDEF
Salida:     0001000000000100
```

En esta ocasión, el único dígito que se transformará en uno es el 3 o el D (que con la antigua tabla también se convertía en 3). Vuelta a lo mismo: el atacante introduce 0000 como PIN de prueba. Si el sistema lo rechaza, es porque hay al menos un 3 en el PIN auténtico. Efectivamente, hay un 3 en el primer dígito, así que el resultado decimalizado es 1000, distinto de 0000. De este modo, al cabo de un máximo de 16 cambios en la tabla de decimalización ya sabemos de qué dígitos se compone.

Tenemos cuatro casos posibles:

—El PIN contiene un solo dígito, digamos el 2. En ese caso lo tenemos a huevo: la única posibilidad es el 2222

—El PIN contiene dos dígitos, digamos el 2 y el 5. Las posibilidades se reducen a catorce: 2225, 2252, 2522, 5222; 2255, 2525, 2552, 5225, 5252, 5522; 5552, 5525, 5255, 2555.

—El PIN contiene tres dígitos. Ahora tenemos 36 posibilidades (hoy estoy muy vago, así que se lo dejo como ejercicio)

—El PIN contiene cuatro dígitos. El número de posibilidades es de 24 (lo siento, sigo en mi fase de vagancia).

Fíjense cómo, manipulando un máximo de 16 veces una tabla de decimalización, hemos pasamos de diez mil posibles números PIN (incluyendo el cuádruple cero) a un máximo de 36. No está mal, teniendo en cuenta que acabamos de empezar. Podemos probarlas todas, o jugar con algunas tablas de decimalización más. Y el ataque puede mejorarse. En lugar de limitarnos a usar las tablas de decimalización “a piñón fijo” como hemos hecho aquí, podemos mejorar el ataque escogiendo una tabla en función de los resultados obtenidos en los pasos anteriores, lo que se conoce con el nombre de **ataque adaptativo**

2) LOS PECADOS DE LOS BANCOS

La historia del ataque de decimalización comenzó enmarcada en un proceso judicial en Reino Unido relacionado con los llamados “reintegros fantasmas”. En ocasiones, desaparece dinero de una cuenta corriente, así sin más. El titular afirma que tiene la tarjeta bajo control, no ha dicho el PIN a nadie ni lo ha escrito en lugar alguno, pero aun así el extracto bancario muestra un reintegro hecho con su tarjeta y con su PIN. En estos casos, el banco asume que la responsabilidad no es suya sino del cliente, por lo que se niega a reembolsarle el dinero.

Este descargo de responsabilidad parte del supuesto de que el sistema es seguro. No hay agujeros, ni vulnerabilidades, de modo que la única conclusión del banco es que el cliente es el responsable. Este fue el caso de Anil y Vanitha Singh. Estos ciudadanos sudafricanos sufrieron en marzo de 2000 nada menos que 190 reintegros fantasmas, efectuados en diversos cajeros automáticos de Londres. No fue el primer caso de reintegro fantasma en llegar a los tribunales, pero la cuantía global era inusitada: casi 70 000 euros al cambio actual.

Los Singh negaron haber hecho tales reintegros, y de hecho se encontraban en Sudáfrica cuando tuvieron lugar los reintegros. Diners Club South Africa no estaba interesado en esas menudencias, y al final presentó una demanda para obligarles a pagar. La lógica (por llamarla de alguna forma) esgrimida fue: el sistema es infalible, no es culpa nuestra, así que es culpa de ustedes.

Por supuesto, el sistema de tarjetas bancarias no es perfecto, y nadie mejor que un experto como Ross Anderson para demostrarlo. Anderson es criptoanalista británico y un peso pesado en su campo. No hay más que ver la cantidad de trabajos que tachonan su página web[3] para darse cuenta de que no es un ignorante en la materia. Se da la circunstancia de que el ataque de decimalización fue el resultado de un trabajo conjunto entre Mike Bond y Ross Anderson, su supervisor, así que no es de extrañar que ambos fuesen llamados como testigos expertos en el caso, que se estaba dirimiendo en el Tribunal Superior de Sudáfrica. Un tercer experto, Jolyon Clulow, había descubierto el ataque de decimalización de modo independiente, y también se unió al caso, pero como testigo en el bando opuesto. Años después, Clulow acabaría uniéndose al grupo de Anderson como investigador.

A los responsables de Diners Club no les interesaba que los fallos técnicos de los sistemas que usan se hiciesen públicos, así que a mediados de febrero intentaron que un tribunal de Londres emitiera una orden restrictiva, según la cual se obligaría a Bond y a Anderson a mantener confidencial toda la información que aportasen, incluyendo su reciente ataque, que de todos modos ya había sido hecho público[4]. La orden no les afectaba solamente a ellos, sino que exigía confidencialidad a otros investigadores de seguridad, incluidos expertos del sistema bancario de Sudáfrica[5].

Anderson, que en cuestiones de privacidad no tiene pelos en la lengua, respondió de forma contundente. Afirmó que las vulnerabilidades en cuestión son de importancia científica significativa, y que una restricción afectaría a su trabajo profesional[6]. También impediría que otras víctimas de reintegros fantasma pudiesen ejercer su derecho a defensa. Anderson declaró al respecto:

"Durante los últimos dos años, ha habido un creciente número de reintegros fantasma. Recibo e-mails con cada vez más frecuencia, por parte de personas de todo el mundo cuyos bancos les cobran reintegros de cajeros automáticos que ellos no han hecho. Los bancos de muchos países se limitan a afirmar que sus sistemas son seguros, así que el cliente es el responsable. Parece ahora que algunas de esas vulnerabilidades han sido descubiertas por los malos. Nuestros tribunales deberían hacer que los bancos arreglen sus sistemas, en vez de dejarles mentir sobre seguridad y cargar los costes al cliente" [5b]].

La orden de restricción se mantuvo. La información revelada durante el juicio y relativa a nuevas vulnerabilidades se mantuvo secreta en el territorio de Inglaterra y Gales, pero la información conocida hasta entonces no resultó afectada.

A pesar de la intervención de Anderson y Bond, en agosto de 2004 el Tribunal Superior de Durban falló a favor de Diners Club. El juez decidió que:

"*Diners Club debe protegerse contra el uso no autorizado de la tarjeta, porque tan pronto como se la da al cliente, Diners Club ya no la controla. Más aún, los clientes no tienen obligación de aceptar esta cláusula [según la cual, los titulares de la tarjeta son responsables de todos los reintegros que requieran el uso del PIN], ni se les obliga a aceptar los números PIN emitidos por Diners Club… Cuando aceptaron las tarjetas y los PIN, los clientes deberían haberse informado de la responsabilidad subyacente*"[7].

En un mundo ideal, o cuando menos sin demasiadas imperfecciones, los jueces reconocerían que la posibilidad de fallos en un sistema de seguridad debería ser usada en beneficio del cliente, que a fin de cuentas es la parte más débil. Los tribunales de Francfort, Alemania, nos dieron un caso extremo de ello. En febrero de 1997, una dentista jubilada de 72 años sufrió el robo de su tarjeta EC (Eurocheque), a lo que siguieron rápidamente seis retiradas fantasma por un importe de 4543 marcos (unos 2300 euros). El banco intentó hacer recaer las culpas sobre el cliente, al afirmar que era imposible que los robos se hubiesen podido realizar sin el número PIN; la cliente, por su parte, insistió en que había guardado el PIN con todo cuidado.

En aquella época, las transacciones estaban cifradas por medio del DES (Data Encryption Standard), un algoritmo de cifra con clave de 56 bits. Descifrar un mensaje protegido con DES hubiese requerido probar 2^56 claves, una cantidad enorme en términos absolutos. Sin embargo, ese mismo año EFF (*Electronic Frontier Foundation*) construyó una máquina apodada DES Cracker, que puede obtener la

clave correcta en unos 4.5 días de media[8]. El coste de construcción (un cuarto de millón de dólares) convertía al DES Cracker en una herramienta al alcance potencial de grupos criminales. Ante la posibilidad, siquiera teórica, de que alguien pudiera utilizar una tarjeta sin conocer el PIN, el tribunal falló en contra del banco y le obligó a devolver a su cliente las cantidades en litigio más un 4% en concepto de intereses[9].

A todos nos gusta pensar que, cuando la verdad se establece de modo claro, la justicia toma nota y la aplica. Es lo que sucedió en el caso de la dentista alemana. Como contraste, el caso Singh resulta especialmente significativo porque muestra un enfoque legal perverso. Los Singh estaban en Sudáfrica cuando los reintegros tuvieron lugar en Londres. Los expertos testificaron explicando los múltiples fallos de seguridad. A pesar de ello, como habían firmado un documento aceptando la responsabilidad, se deduce que no hay más que hablar, caso cerrado. No fue ni con mucho el primer caso de un reintegro fantasma en el que el banco pretendió descargar las responsabilidades sobre sus clientes, ni tampoco sería el último.

John Munden era un agente de policía de Cambridge con una hoja de servicios impecable y casi veinte años de servicio a sus espaldas. Tras una felices vacaciones en Grecia en 1992, volvió a su casa para descubrir que su cuenta corriente había sido vaciada. Al ir a pedir explicaciones, su banquero, que había notado los movimientos de su cuenta, le preguntó cómo le había ido en sus vacaciones en… Omagh, Irlanda. Al parecer, alguien desde allí ordenó seis reintegros por un total de 460 libras esterlinas.

Cuando Munden replicó que había estado en Grecia, no en Irlanda, la versión oficial del banco cambió radicalmente: fue Munden quien retiró el dinero en su banco local justo antes de irse de vacaciones. Cuando Munden insistió en su queja, el banco contraatacó furiosamente: llevó al agente a los tribunales y ganó. La lógica aplastante del banco era la habitual: nuestros sistemas son seguros, por tanto la culpa necesariamente es del cliente[10]. Munden se vino abajo: perdió peso, contrajo una úlcera duodenal y su mujer intentó incluso suicidarse. Además de ello, fue suspendido de su cargo, y solamente el apoyo de sus vecinos y la intervención de un nuevo jefe de policía consiguieron evitar su expulsión[11].

La salvación para Munden vino precisamente de la mano de uno de sus vecinos: Ross Anderson (también de Cambridge, no lo olviden), quien intervino en el caso de forma contundente. Según sus palabras, “*la descripción de los sistemas [de seguridad] del banco, según se vio en el juicio, recordaban más a Laurel y Hardy que al [estándar] ISO 9000*”. Resulta que la sucursal involucrada (del banco Halifax) no había investigado la transacción en disputa; no solamente esa, sino que tenía casi doscientas “transacciones en disputa” pendientes de investigar. El banco afirmó que tenía cámaras de seguridad para grabar a quien realizaba retiradas de dinero en sus

cajeros, pero dichas cámaras casualmente no funcionaron cuando se realizaron los reintegros en cuestión. Los cajeros automáticos realizaron las operaciones de cifrado mediante software, un procedimiento mucho más vulnerable que el de hardware. En palabras del propio Anderson:

"*Mi reacción [a la condena de Munden] ha sido retirar mi dinero de Halifax y cerrar mi cuenta. Aparte de que tienen unos sistemas penosos, la idea de que quejarme por un error informático puede llevarme a prisión está por encima de mi límite de tolerancia*"[12].

John Munden apeló ante un tribunal superior. El banco Halifax presentó un grueso documento procedente de su auditor (KPMG), según el cual el banco era seguro. Cuando la defensa pidió y consiguió acceso a dicho documento, la acusación se negó, y el banco que presumía de seguro se negó a permitir que otros examinasen su seguridad. Nueve meses de constantes negativas por parte del banco agotaron la paciencia del juez, quien acabó dictaminando que la acusación no podría usar el documento del experto como prueba a su favor puesto que a la defensa no se le permitía el mismo acceso:

"*Cuando un caso involucra ordenadores o equipos similares, entonces, como cuestión de justicia común, la defensa debe tener acceso a las pruebas y ver si hay algo que haga falible a los ordenadores*".

El banco no estaba dispuesto a que escrutasen la seguridad de sus sistemas, así que no tuvo más remedio que dar la callada por respuesta, con lo que perdía cualquier posibilidad de presentar evidencia informática. El juez dictaminó a favor de la defensa. John Munden fue absuelto y reivindicado.

En general, el problema legal es peliagudo por ambos bandos. En el caso de los bancos, tienen que hacer frente a múltiples fraudes, a los que hay que añadir los problemas derivados de la mala gestión de las tarjetas y de los PIN asociados por parte de los clientes. Ciertamente, hay muchos casos de fraude por parte del cliente, o sencillamente de mal uso. El problema es que, por otro lado, los defraudadores profesionales son muy hábiles y el sistema tiene multitud de fallos. Una red mundial de tarjetas de crédito, donde intervienen actores tan diversos como bancos, emisores de tarjetas, usuarios, empresas, terminales de punto de venta, etc, forzosamente ha de ser complicado de establecer, mantener y asegurar.

Lo que no resulta justificable en modo alguno es que, por defecto, la responsabilidad de un mal uso recaiga en el cliente. Eso solamente podría ser admisible en el caso de un sistema perfecto, en cuyo caso podríamos aplicar el principio usado por Sherlock Holmes: si eliminamos todo lo imposible, entonces lo que queda, por improbable que sea, debe ser cierto. El sistema de tarjetas dista mucho de ser imposible de subvertir. Así pues, la responsabilidad del cliente debe ser limitada, y siempre ha de considerarse la posibilidad de que haya habido una

violación de seguridad. Los atacantes pueden mirar por la espalda del cliente para ver cómo teclean el PIN, o bien poner una cámara en miniatura para captar los movimientos de los dedos. Pueden alterar el cajero para insertar un dispositivo que graba el contenido de la banda magnética. Esos son, por cierto, dos de los procedimientos más habituales para clonar tarjetas bancarias.

Mi propia esposa fue víctima de un fraude con tarjeta en España. Alguien usó su número de tarjeta de crédito en 2007 para realizar diversas compras por Internet por un total de 80 euros. Me alegra poder decir que la entidad emisora de la tarjeta (el banco ING Direct) se comportó correctamente y le reembolsó el dinero. En relación a los autores, al parecer los cargos se hicieron desde la misma ciudad que mi esposa había visitado seis meses antes, y en la que utilizó la tarjeta.

En general, que una víctima encuentre una solución satisfactoria a su problema depende del país, la época y las circunstancias. Nuestro amigo Ross Anderson escribió a comienzos de los años noventa un artículo cuyo título es esclarecedor: "*por qué fallan los criptosistemas*"[13]. Podemos allí ver ejemplos estremecedores acaecidos en el Reino Unido en los años ochenta y noventa:

—En 1985, una adolescente fue condenada por robarle cuarenta libras a su padre. La chica se declaró culpable por consejo de sus abogados. Más tarde se descubrió que el robo no fue tal, sino un error contable del banco.

—En 1988, un sargento de la policía de Sheffield fue acusado de robo y suspendido de su cargo porque una tarjeta que había confiscado a un sospechoso sufrió una retirada fantasma. Tuvo la fortuna de que sus compañeros encontraron a una mujer cuyo testimonió logró exonerarle.

—En 1992, un ama de casa de Hastings sufrió repetidos robos en su cuenta. Los sistemas de seguridad del banco no notaron nada raro, y se negaron a creerle. El caso solamente se aclaró cuando un empleado del banco tuvo una crisis de conciencia y confesó: él había hecho una segunda tarjeta para su propio uso. Le ayudó el hecho de que las retiradas de fondos en cajero no aparecían en la lista de movimientos que se enviaba al cliente.

—Escocia, 1992. Un ingeniero de mantenimiento introdujo un miniordenador en un cajero automático para capturar números de cuenta y PIN. Con esa información consiguió falsificar tarjetas y saquear cuentas. Mientras tanto, el propio banco actuaba a la defensiva frente a las víctimas.

—En una ocasión, un banco emitió tarjetas especiales a los cajeros (humanos) para los casos en que se quedasen temporalmente sin fondos. Lo especial de las tarjetas es que podían extraer dinero de la cuenta corriente de cualquier cliente del banco.

—En 1987, el personal de un banco descubrió que las operaciones de sus cajeros no estaban cifradas ni autenticadas, de forma que podían reproducir una orden de

pago una y otra vez, cosa que hicieron en repetidas ocasiones. Los cómplices, en el exterior, recogían el dinero que salía del cajero hasta dejarlo vacío.

—En otro banco, unos ladrones descubrieron un fallo de programación. Cuando se introducía una tarjeta telefónica en el cajero, éste reaccionaba como si fuese la tarjeta de crédito introducida justo antes. No había más que observar a un cliente, apuntar su PIN, esperar a que se fuese y vaciar su cuenta.

—Cierto modelo de cajero automático tenía oculta una funcionalidad curiosa: al introducir una secuencia concreta de catorce dígitos, escupía diez billetes. Esta opción se introdujo para poder probar el sistema. Un banco tuvo la genial ocurrencia de imprimir el número en un manual. No sabría yo decir qué es más extraordinario, que el banco hiciese tal estupidez… o que los defraudadores tardasen tres años en sacarle partido.

—Un error de programación hizo que una pequeña entidad bancaria emitiese el mismo número PIN para las tarjetas de todos sus clientes. En otro caso, un programador trucó el sistema para que solamente se creasen tres PIN distintos, en este caso con fines delictivos.

—En la década de los ochenta, no era raro que un cajero automático estuviese desconectado de la red. Eso provocó una oleada de fraudes en Italia y el Reino Unido. El ladrón abría una cuenta, obtenía una tarjeta con su PIN, hacía docenas de copias de la tarjeta y sus cómplices las usaban para retirar efectivo de forma simultánea en tantos cajeros como pudiesen.

—En algunos casos, el banco subcontrataba la seguridad de sus cajeros a empresas externas, y para ello les proporcionaba las claves criptográficas que contenían.

¿Sorprendidos? Pues aún nos queda lo mejor. Prepárense, que vienen curvas.

3) EL ESLABÓN MÁS DÉBIL

Como hemos visto, el PIN de una tarjeta se verifica dentro del cajero, en un proceso que pasa por usar el algoritmo de cifrado DES y utilizar una clave que se encuentra almacenada en forma segura. Eso significa que debe haber una forma segura de crear la clave. El procedimiento habitual en los noventa era el siguiente. El banco envía a la sucursal a dos empleados, cada uno de los cuales lleva la mitad de la clave, escrita en papel. Ambas mitades se introducen en el teclado, y como resultado se genera la clave *K* del cajero. A continuación, se utiliza esa misma clave de papel para cifrar la clave *K* y enviarla por red al ordenador central del banco.

Las cosas se fueron complicando cuando se crearon las redes de cajeros y los clientes de un banco tuvieron la posibilidad de sacar dinero del cajero gestionado por otro banco de la misma red. Posteriormente, incluso las tarjetas de una red pudieron ser usadas en otra red (con comisiones, por supuesto), y luego llegaron tarjetas ligadas a una entidad emisora que no tiene que ser un banco o caja.

Para dotar de flexibilidad extra al sistema, es preciso establecer un medio de verificación seguro. Ya no basta con crear una clave *K* compartida entre el cajero y el ordenador central del banco. En su lugar, la autorización para extraer el dinero o hacer un cargo se hace en un **Centro de Verificación (CV)**, que puede ser el del propio banco, el de VISA o cualquier otro que corresponda. No siempre es posible una comunicación directa entre el cajero y el CV, así que el proceso se lleva a cabo mediante un conjunto de **puntos de conmutación**, que actuarán como eslabones del sistema.

El proceso va a ser el siguiente. El cliente mete la tarjeta en el cajero e inserta el PIN. El cajero toma ese PIN, le da formato y lo cifra con su clave *K1*. El resultado, llamado Bloque PIN Cifrado (EPB) es enviado al primer punto de conmutación, que comparte con el cajero esa clave *K1*. Este punto de conmutación descifra el mensaje, obtiene el PIN, lo reformatea si es necesario, lo cifra con otra clave *K2* y lo envía al segundo punto de conmutación. El proceso se repite hasta que finalmente llega al centro de verificación. En cada uno de esos conmutadores, el proceso de descifrado y recifrado se efectúa en un módulo criptográfico hardware seguro, y durante el transporte el PIN siempre viaja cifrado.

Se trata de una cadena donde todos los elementos han de ser fuertes para que la cadena también lo sea. Pero en 2006 Omer Berkman y Odelia Moshe Ostrovsky, del Instituto Académico de Tel Aviv (Israel) descubrieron que las cosas no son tan sencillas como parecen: ese “formateo” que se hace a los bloques PIN esconde una vulnerabilidad. El artículo que escribieron tiene por nombre “*la insoportable levedad del crackeo de PINs,*” lo que de entrada ya da una idea sobre lo poco que les impresionó el sistema de seguridad que estaban atacando.

Para entenderlo, sepa el lector que la información que se transmite por la red de puntos de conmutación no siempre se limita al PIN. Existen cuatro estándares para empaquetar la información, conocidas con el nombre de ISO-0, ISO-1, ISO-2 e ISO-3. El primero acompaña el PIN con el número de cuenta; el segundo utiliza un paquete de datos aleatorios; el tercero no usa ninguno de los dos y el cuarto los utiliza ambos. Cuando un punto de conmutación recibe el paquete de datos en uno de esos estándares, puede re-formatearlo en otro estándar diferente.

Lo primero que Berkman y Ostrovsky notaron es que el formato ISO-0 no sirve como código autenticador de mensajes. En ese formato, el bloque de datos es una suma del PIN y el número de cuenta, lo que significa que no puede verificar la autenticidad de ninguno de los dos. El ISO-1 no es mucho mejor, y no debería ni existir. En cuanto al formato ISO-2, también resulta tremendamente vulnerable: puesto que solamente incluye un número PIN, basta con enviar 10 000 números PIN por la red y anotar los 10 000 valores que salen del punto de conmutación. De hecho, se trata de un formato tan vulnerable que ni siquiera está aprobado oficialmente para transacciones online. El trabajo de los investigadores israelíes demuestra que una cadena de puntos de verificación acabará por degradar la seguridad del sistema hasta hacerlo tan malo como el formato ISO-2. Una cadena tan vulnerable como el eslabón más débil.

Los puntos de conmutación no suelen estar controlados por las entidades emisoras de tarjetas, y por tanto no pueden protegerlos adecuadamente. Escoger un formato ISO adecuado sería una solución, pero el proceso de adopción (y lo más importante, la prohibición de usar los otros formatos) en todos los puntos de todas las redes sería un proceso que bien pudiera durar décadas.

Los autores ponen el dedo en la llaga al sugerir, mejor dicho, al afirmar a las claras que los ataques que describen pueden dar explicación a muchas retiradas fantasma. Las consecuencias pueden ser devastadoras: “*los ataques [descritos] son tan sencillos y prácticos que los emisores de tarjetas tendrían que aceptar responsabilidades no sólo por los casos [de fraude] futuros sino también con carácter retroactivo. Los ataques pueden aplicarse a una escala tan grande (en algunas de las variantes, se pueden descubrir hasta 18 000 000 números PIN) que la responsabilidad puede llegar a ser enorme*[14]”.

4) LLEGAN LAS TARJETAS CON CHIP

Cuando los parches a un sistema no son suficientes para garantizar su seguridad, urge un cambio de paradigma. Eso es lo que se propuso la industria bancaria para reducir de raíz el fraude de tarjetas de crédito. El uso de tarjetas en una variedad cada vez mayor de servicios (sea compra por Internet o pago en un restaurante) hacía deseable un sistema de protocolos más flexible, más fiable, y ya puestos más seguros. Gracias a la caída de precio y el aumento en potencia de los chips informáticos, había llegado el momento en que un pequeño chip en una tarjeta podría sustituir a la anticuada banda magnética. Sería como un miniordenador capaz de gestionar las etapas de autenticación, verificación y autorización; además de ello, copiar o clonar información de una banda magnética es fácil, pero hacerlo a partir de un chip de hardware resulta mucho más difícil.

Las empresas Europay, MasterCard y VISA unieron fuerzas y crearon el estándar EMV, regulado por medio de la entidad EMVCo que fundaron con este fin en 1999[15]. El lector no debe sacar la conclusión de que EMV es un conjunto de protocolos fijo e inamovible. Más bien es una especie de caja de herramientas, de donde cada banco o entidad emisora escoge los protocolos de firma digital, autenticación, cifrado, etc que desea usar. El lector interesado puede encontrar las especificaciones de seguridad y gestión de claves en[16].

En general, podemos dividir el protocolo de seguridad EMV en tres fases: autenticación, verificación y autorización:

—Autenticación (de la tarjeta). Garantiza al terminal (cajero automático o terminal punto de venta) cuál es el banco emisor, y también que la tarjeta no ha sido alterada.

—Verificación (del cliente). Asegura al terminal que el PIN introducido por el usuario es el que corresponde a la tarjeta.

—Autorización (de la transacción). Asegura al terminal que el banco emisor de la tarjeta autoriza la transacción.

Los algoritmos específicos de seguridad son bastante seguros. Para el cifrado simétrico, DES fue descartado por la corta longitud de su clave, y en su lugar se utilizó una variante llamada TripleDES. También es posible utilizar el algoritmo AES, mucho más seguro y rápido. Los algoritmos asimétricos están basados en RSA, y la función de hash utilizada es la SHA-1. Como el lector ya habrá descubierto leyendo este libro, buenos algoritmos de cifrado no garantizan un protocolo seguro; con todo, es indudablemente un buen comienzo.

La adopción del sistema EMV ha sido más o menos rápida dependiendo del país. Según datos de EMVCo correspondientes a mediados de 2011, tres cuartas partes de todos los terminales, y casi la mitad de las tarjetas, del mundo entero están adaptadas

al nuevo sistema. La región líder en su adopción ha sido Europa, con más del 80% de tarjetas con chip respecto al total en su región. Extrañamente (o no, como veremos luego), los Estados Unidos, tradicionales usuarios masivos de tarjetas bancarias de todo tipo, han sido mucho más lentos en adoptar este estándar, con apenas un 38% de penetración en tarjetas, aunque el 80% de sus terminales las aceptan[17].

También en España tardaron en llegar las tarjetas con chip. Recuerdo haber asistido a una conferencia sobre seguridad electrónica en 2002, en la que dos responsables del BBVA afirmaban que ya disponían de esa tecnología y que el banco comenzaría a distribuirlas a sus clientes en breve; no fue hasta finales de 2009 cuando el BBVA lanzó una campaña publicitaria en la que se enorgullecían de ser "*el primer gran banco que inicia el proceso de migración al estándar de seguridad EMV*"[18]. Realmente no fueron los primeros, pero la verdad es que la adopción del estándar EMV fue muy lento en España, con apenas el 10% del total de tarjetas en circulación a marzo de 2009[19].

¿Por qué esta disparidad? ¿Qué impulsa a españoles y norteamericanos a rechazar un sistema tan seguro, recibido por los británicos con los brazos abiertos? El motivo se encuentra en la legislación bancaria. Las leyes norteamericanas protegen fuertemente al cliente en caso de disputa con su banco por un asunto de fraude. En lugar de asumir que el cliente es culpable o responsable del modo que sea (por ejemplo, por negligencia en la custodia de la tarjeta o del PIN), es el banco quien tiene que demostrarlo. En caso contrario, el banco sufre la pérdida y el cliente queda protegido. En España sucede algo similar, como veremos más adelante.

En un país como el Reino Unido, con leyes que protejan al banco, éste solamente tiene que adoptar un sistema de seguridad adecuado para, de ese modo, poder culpar "por defecto" al cliente ante cualquier disputa. Ya saben: el sistema funciona bien, no hay fallos, por consiguiente el cliente es el culpable y debe pagar, fin del caso. Es una aplicación perversa del "principio de Sherlock Holmes". Es la misma lógica (por llamarla de alguna forma) que siguieron durante los años previos a la aparición de la tarjeta de chip, y que hemos analizado en las secciones anteriores de este capítulo. El aliciente principal para dotarse del sistema de seguridad basado en chip no es reducir el fraude (aunque bienvenida sea esa reducción), sino pasarle la patata caliente al cliente. Ni más ni menos.

Por el contrario, en un país que disponga de una legislación garantista con los derechos del consumidor, como España o Estados Unidos, la entidad emisoras de tarjetas no puede encogerse de hombros y echar la culpa al cliente de forma automática por un fraude derivado de la inseguridad del sistema. Si hay disputa, es el banco quien debe probar que éste ha cometido fraude o que es responsable por negligencia, y el cliente no tiene responsabilidad (o la tiene en forma limitada). En tales condiciones, adoptar un sistema más seguro no le sirve para desviar la

responsabilidad hacia el cliente, así que ¿para qué molestarse? No existe ese aliciente de “pasemos la patata caliente al cliente” para hacerlo.

Por supuesto, un sistema más seguro es bueno *per se* para el emisor, ya que reduce las pérdidas derivadas de fraude, mala publicidad y pérdida de confianza, pero esas consideraciones deben contraponerse con el alto coste de cambiar a un nuevo sistema. En 2006, el diario Daily Mail evaluó el coste de la adopción del sistema EMV en el Reino Unido en más de 1300 millones de euros[20].

A tenor de la lentitud con la que este sistema ha sido adoptado en España, parece que el aliciente principal haya sido, sencillamente, mantenerse a la par de los demás países. En Estados Unidos, ni siquiera se han molestado en intentar adoptarlo hasta ahora. Hubo que esperar hasta comienzos de 2012 para que VISA[21] y MasterCard[22], dos de las fundadoras de EMVCo, anunciasen la implantación del estándar en EEUU.

En los casos español y norteamericano, la legislación protege al cliente, y de ese modo el emisor de tarjetas carece de uno de los dos principales alicientes para adoptar el sistema EMV (echar las culpas al cliente). La situación era muy distinta en el Reino Unido. Como hemos visto, los bancos allí culpaban al usuario en cuanto tenían ocasión, y nada mejor para ello que poder asegurar que sus sistemas son fiables. Las primeras pruebas del sistema EMV, conocido allí como *Chip and PIN*, tuvieron lugar en mayo de 2003; en octubre de ese mismo año se anunció su adopción a nivel nacional, y en enero de 2005 se obligó a todos los propietarios y administradores de terminales punto de venta (TPV) a cambiar al nuevo sistema. Nadie fue obligado a punta de pistola, pero a partir del 1 de enero de 2005, la responsabilidad por fraude recaería sobre quien no hubiese tomado medidas de seguridad para evitarlo, como por ejemplo un vendedor que usara un TPV no actualizado.

Nadie quería que la patata caliente de la responsabilidad legal acabase en sus manos, y como consecuencia en febrero de 2006 cualquier usuario de tarjetas con chip estaba ya obligado a usar su PIN; solamente se permitieron excepciones en casos de personas con discapacidad. Este era un punto muy importante. Antes de eso, una persona tenía que firmar el recibo, y esa firma era la prueba de la transacción. Era responsabilidad del banco comprobar que la firma era correcta. Sin embargo, con el nuevo sistema la única prueba sería el PIN, y como se daba por supuesto que el cliente no lo compartiría con nadie la introducción del PIN correcto se consideraría prueba por defecto de que el cliente, o bien había ordenado la transacción, o bien había permitido que terceros utilizasen el PIN. En cualquier caso, el cliente paga. La invasión del “Chip and PIN” había concluido con éxito.

Y sin embargo, la última nota de prensa que aparece en la web de información sobre el nuevo sistema (www.chipandpin.co.uk) hace alusión a un programa de la

BBC en el que se menciona un ataque contra tarjetas de crédito revelado por un equipo de la Universidad de Cambridge[23]. ¿Le suena familiar, lector? Por si acaso, le refrescaré la memoria con dos palabras: Ross Anderson. En efecto, el azote de la mala criptografía cargó contra el nuevo sistema, y demostró (seguro que a estas alturas no se sorprende usted) que no es tan seguro como se creía.

La adopción del sistema EMV tuvo un éxito parcial. Si bien se constató una disminución de algunos delitos cometidos con tarjetas de crédito, como hurtos o fraudes presenciales, los fraudes no presenciales (compras por teléfono o por Internet) se dispararon, y a mediados de 2009 representaban nada menos que el 50% de todo el fraude con tarjetas de crédito en el Reino Unido. Las soluciones que sirven para pagos presenciales pueden no ser igualmente válidas en el mundo digital actual. Como respuesta, a comienzos de 2009 se cambió la ley británica para proteger al consumidor: sería el banco quien tendría que demostrar la culpabilidad del cliente, en lugar de aceptarlo a priori como hecho probado[24].

Entramos así en una nueva campaña para probar la seguridad del sistema EMV. No será, por supuesto, una búsqueda exhaustiva, ni se darán todos los detalles técnicos aquí. Nos limitaremos a ver algunos de los casos más conocidos y la repercusión que pueden tener para la industria de tarjetas.

La primera vulnerabilidad es muy conocida, y está ligada con el hecho de que no se puede cambiar todo un sistema de la noche a la mañana. Hay un período de transición en el que los nuevos chip conviven con las antiguas bandas magnéticas. Saque su cartera y échele un vistazo a su tarjeta. Apuesto a que la mayoría de las transacciones que ha realizado usted en los últimos meses se basaron exclusivamente en el PIN y el chip (por eso ya no le piden el DNI para efectuar algún pago). Y, a pesar de ello, ahí está. La negra banda magnética sigue en el reverso de la tarjeta, lista para ser activada cuando por algún motivo no puede utilizarse el sistema Chip+PIN.

En mayo de 2006, la empresa petrolífera Shell anunció que suspendía el uso de tarjetas de Chip+PIN en las más de 600 gasolineras que poseía en el Reino Unido. El motivo fue que centenares de clientes sufrieron hurtos en sus cuentas tras haber utilizado sus tarjetas en dichas gasolineras, en un montante que inicialmente se estimó en más de un millón de libras. El modus operandi de los ladrones fue el siguiente: disfrazados de técnicos de mantenimiento, alteraron el funcionamiento de los lectores de tarjetas, insertando dispositivos que capturaban los PIN y los detalles de las bandas magnéticas. Con esa información, clonaron tarjetas que posteriormente utilizaron para retirar grandes sumas de dinero en cajeros automáticos[25]. Shell no reintrodujo la opción de pago con chip hasta septiembre.

No hay indicación de que la información dentro del chip sufriese copia alguna. Los defraudadores aprovecharon la menor seguridad de la banda magnética, el eslabón más débil, para copiar su información y utilizarla de modo fraudulento. Eso

se supo más tarde, pero durante varios meses cundió el rumor de que el sistema Chip+PIN británico, presentado con gran pompa como la solución contra el fraude de tarjetas de crédito, era vulnerable. Eso provocó una pérdida de confianza en el nuevo chip, algo injusto puesto que el sistema seguía siendo seguro. Pero resultó un buen recordatorio de la vulnerabilidad asociada a un doble sistema de autenticación: si uno resiste y el otro falla, la seguridad desaparece.

Hubo otros casos de fraude, aunque de menor escala, en los que la presencia de la banda magnética saboteaba la seguridad del sistema Chip+PIN. Uno de los más conocidos fue el de Jane Badger. Su caso guarda ciertas similitudes con el caso Munden de los años noventa. Como Muden, Jane Badger era agente de policía. En 2006 informó a su banco de una transacción fraudulenta por importe de 772 libras (casi mil euros en la actualidad), que ella afirmó no haber realizado. El Banco (Egg) no solamente no aceptó su palabra sino que la acusó de fraude. Su casa fue registrada, ella fue detenida y suspendida de empleo. Y, al igual que Munden, ella tuvo la fortuna de contar con la ayuda de un criptógrafo llamado Ross Anderson (sí, él de nuevo), quien testificó en su favor[26].

Para probar el fraude, Egg mostró una copia impresa con un código de transacción 05, que significaba "*lectura de Tarjeta con Circuito Integrado - datos CVV fiables*". Eso parecía indicar que los datos de la tarjeta procedían del chip, un lugar en el que no pueden ser duplicados, y por tanto la transacción se realizó con la tarjeta original, no con una tarjeta clonada que contuviese la información en banda magnética. La duda consistía en los "datos CVV". Todas las tarjetas bancarias tienen impreso un número de tres dígitos denominado CVV (*Card Verification Value*) para asegurar que los datos guardados en la banda magnética son válidos y fueron generados por la entidad legítima.

Pero no tiene mucho sentido comprobar el CVV cuando se está utilizando el sistema Chip+PIN. Así que la siguiente pregunta es esencial: la expresión "lectura de Tarjeta con Circuito Integrado - datos CVV fiables" ¿qué significa exactamente? ¿Se leyó el chip y además se comprobaron los datos CVV, o se hizo solamente una de las dos operaciones? En el segundo caso, eso solamente dice que se utilizó, una de dos: o bien el chip, o bien la banda magnética, pero no se indica cuál fue el caso. Parece un detalle menor, pero es una pregunta básica para averiguar qué sistema fue utilizado realmente, el de chip (seguro) o el de banda magnética (inseguro). La defensa solicitó la presentación de los archivos originales de la transacción y demás material relacionado. Egg se negó, y perdió el caso. Jane Badger fue absuelta en enero de 2008.

Es posible que lo que digo a continuación no sea más que una casualidad, pero lo diré por complitud: apenas unos días después de la absolución de Badger, el banco Egg (adquirido por Citicorp en mayo de 2007) anunció la retirada de 161 000 tarjetas

de crédito. Los motivos aducidos fueron un excesivo perfil de riesgo por parte de dichos clientes, si bien algunos clientes adujeron motivos de maximización de beneficios[27]. En la actualidad se especializa en seguros y cuentas de ahorro; su negocio de tarjetas de crédito fue vendido a Barclays Bank en 2011.

Uno de los problemas con los que se encontraban las víctimas de fraude es que era el propio banco quien contaba con las pruebas que podían exonerarlas. Hacía falta llegar a juicio, y cuando el banco era obligado a entregar los documentos (en papel o electrónicos) relativos a las transacciones en disputa, éste se negaba. De ese modo se ganaron casos como los de Munden y Badger: no es que consiguiesen demostrar su inocencia, sino que ganaron por incomparecencia del banco. Oficialmente, la inseguridad de los sistemas bancarios sigue sin ser demostrada, así que a dichos bancos les interesa más no presentar las pruebas que demuestra su vulnerabilidad y pagar. Mejor eso que reconocer la verdad ante millones de clientes.

A veces el banco se sale con la suya. Como ejemplo tenemos el caso de Alain Job, quien en febrero de 2006 afirmó haber sufrido varias "retiradas fantasma" de efectivo desde dos cajeros automáticos, por un total de 2100 libras (unos 2600 euros). Tras quejarse al banco y afirmar que él no había realizado esas transacciones, la respuesta del banco fue la habitual: negarlo todo y culpar al cliente. Jobs presentó una queja ante la oficina del Defensor Financiero, y al fallar en su contra[28] denunció al banco (Halifax) ante los tribunales en febrero de 2007. Durante el juicio se desveló que el banco había destruido información vital para la resolución del caso: los recibos en papel, los "logs" electrónicos, e incluso la propia tarjeta del señor Job.

El juez rechazó la argumentación de Job (que él había guardado la tarjeta y el PIN de forma segura en todo momento), y a pesar de la destrucción de las pruebas no consideró que el banco tuviese que probar la validez de la transacción. Al contrario, decidió que en ausencia de pruebas el tribunal debía aceptar la validez de la operación de retirada "por lo que valga," y falló a favor del banco en junio de 2009[29]. Alain Job no solamente perdió el caso sino que tuvo que pagar más de 18 000 euros al banco en concepto de costas del juicio.

La situación empeoró para los clientes británicos en 2007. Enfrentados de nuevo a la existencia de un sistema de tarjetas inseguro, los bancos y emisores de tarjetas forzaron cambios. Hasta entonces, los casos de fraude eran denunciados por los clientes ante las autoridades policiales, pero desde abril de 2007, los clientes solamente podían denunciar los fraudes ante el propio banco, quien determinaría si había que dar parte o no a la policía[30]. La asociación británica APACS (*Association for Payment Clearing Services*), que actúa como foro para los temas relativos a los sistemas de pago y suele representar a bancos y emisores de tarjetas, afirmó que el cambio redundaba en ventajas para el usuario; pero es evidente que, si los bancos controlan las denuncias por fraude en tarjeta están en una posición extremadamente

ventajosa respecto al cliente. Si éste no está de acuerdo, puede acudir al Defensor Financiero (*Financial Ombudsman*), una oficina teóricamente independiente cuyo presupuesto proviene de la industria de finanzas, y que puede tardar más de un año en tomar una decisión.

Peor aún, cuando los bancos decidieran denunciar el fraude, pasarían la información a las fuerzas regionales de policía, o bien a la Unidad Dedicada para Delitos de Cheque y Plástico (DCPCU). El problema es que esta unidad fue creada en 2002 por la propia industria bancaria, y está patrocinada por ella (a través de la APACS), lo que lo hace, digámoslo suavemente, muy influenciable en el cumplimiento de su labor. Por muy profesional y dedicado que sea su trabajo policial, el hecho de estar pagados por la industria bancaria recuerda al lobo que guarda el redil. Pronto tuvieron que enfrentarse a un fraude real, a gran escala y con un grado de sofisticación sin precedentes.

5) EL ATAQUE DEL DOCTOR MALIGNO

Como vimos en el caso Shell, la táctica de manipular el terminal para extraer información de la tarjeta es un punto de ataque en el que los protocolos criptográficos resultan inútiles. No fue un hecho aislado. Durante 2006, se detectaron diversos casos de clonado en los que tarjetas de ciudadanos alemanes de vacaciones eran copiadas en Italia y Hungría[31].

Un grupo de investigadores consiguió controlar los circuitos de un lector de tarjetas de forma tan perfecta ¡que incluso se permitieron reprogramarlo para jugar al Tetris![32]. Suena cómico, pero el poder de manipular un terminal asusta. Usted podría pensar que está pagando con tarjeta la cuenta del restaurante, sin saber que el camarero está utilizando esa transacción para pasar la información a una segunda tarjeta (fraudulenta) y realizar compras sin que el usuario legítimo se entere hasta que sea demasiado tarde. Esa compra fraudulenta no podría ser disputada, porque para la tienda fue el cliente quien utilizó su tarjeta y su PIN de forma correcta[33].

En febrero de 2008, un artículo firmado por los investigadores de Cambridge Saar Drimer, Stephen J. Murdoch y (sorpresa, sorpresa) Ross Anderson mostraba precisamente cómo dos de los teclados más utilizados para introducir los PIN de las tarjetas eran vulnerables, y podían ser fácilmente manipulados para extraer información que permitiera clonar la tarjeta[34]. El ataque era, al menos en potencia, particularmente grave en el Reino Unido por dos motivos: el primero, que las tarjetas con Chip+PIN habían sido ya diseminadas y usadas de forma masiva; el segundo, que bastaban pocas horas para sacar una tarjeta clonada del país y utilizarla en otro, digamos en Irlanda o en Holanda.

A pesar de que el ataque salió descrito en un programa de la BBC, y de que Drimer, Murdoch y Anderson recibieron el premio al artículo más práctico en el Simposio de la IEEE sobre Seguridad y Privacidad en 2008, no parece que la industria de tarjetas de crédito se lo tomase muy en serio. Un comunicado de VISA respondió del modo habitual en la mayoría de estos casos: "*no tenemos pruebas, a partir del artículo académico de Cambridge, de nada que no supiésemos ya, o de nada que presente una amenaza a la seguridad de las tarjetas en el mundo real*"[35]. La empresa Ingenico, fabricante de los lectores de tarjeta analizados fue incluso más allá:

"*El método identificado en el artículo de la Universidad de Cambridge requiere conocimientos especializados y tiene dificultades técnicas inherentes. Este método, por tanto, no es reproducible a gran escala, y no tiene en cuenta el proceso de vigilancia de fraude utilizado por la industria*"[36].

Por su parte, APACS, que coordinó la introducción del sistema Chip+PIN en el Reino Unido, evaluó como muy bajo el riesgo del nuevo ataque porque, en su

opinión, requeriría mucho esfuerzo, sería fácil de detectar y no resultaría económicamente viable para grupos criminales[37]. Según se desprende de estas declaraciones, los descubrimientos hechos por algunos académicos aquí y allá no son más que brillantes ejercicios teóricos para publicar artículos especializados, pero sin aplicaciones en el mundo real. No hay un Doctor Maligno oculto en una base secreta de alta tecnología, manipulando lectores de tarjetas a millares para obtener grandes beneficios.

Para sorpresa de todos ¡eso era exactamente lo que estaba sucediendo! En octubre de 2008 se descubrió que centenares de lectoras de tarjetas habían sido manipuladas para extraer información. Lo novedoso del caso consistió en que la manipulación no fue efectuada en los países de uso, sino en China, donde las máquinas lectoras se construían. Siguiendo un guión de película, los atacantes entraron en la fábrica de máquinas lectoras, y realizaban allí mismo las manipulaciones, o bien consiguieron acceso poco después. Las lectoras viajaron luego a Europa, donde fueron instaladas en centenares de tiendas y supermercados de al menos cinco países: Reino Unido, Irlanda, Bélgica, Holanda y Dinamarca. Cuando un cliente insertaba su tarjeta, un módulo manipulado leía los datos y, mediante una conexión telefónica GSM, enviaba la información a un número de Lahore (Pakistán).

La sofisticación de este ataque iba más allá de los detalles técnicos. Los defraudadores utilizaban solamente parte de la información de las tarjetas de crédito leídas. A veces las lectoras enviaban datos sobre una de cada diez tarjetas usadas; en otros casos, se interesaban solamente por las tarjetas VISA Platino. Las lectoras llamaban a casa a intervalos irregulares, diariamente o una vez por semana. Cuando la información robada era recibida los ladrones fabricaban tarjetas clonadas, pero esperaban al menos dos meses para usarlas, con lo que complicaban las labores de seguimiento de la policía y los bancos. Una vez pasado un tiempo prudencial, procedían a desvalijar al cliente legítimo mediante compras en otros países, transacciones online (por teléfono o Internet), y en ocasiones, retiradas de efectivo en cajero.

La banda llegó incluso al extremo de realizar los robos “a medida” según el tipo de tienda donde estaba instalada la máquina lectora de tarjetas. Si se trataba de una tienda pequeña, las cantidades que defraudaban eran pequeñas; por el contrario, las tiendas grandes de electrónica eran las candidatas ideales para un saqueo a gran escala. Por ese motivo, las autoridades tardaron varios meses en detectar el fraude. Se cree que comenzó hacia enero-febrero de 2008, lo que no permite aclarar si el grupo obtuvo la idea del artículo de Anderson de febrero, o si llegaron a la conclusión ellos solos.

Lo cierto es que se trató de un ataque impresionante: a pesar de que ya en marzo MasterCard detectó patrones extraños en el uso de tarjetas de crédito (lo que sugería

un posible fraude), las autoridades tardaron seis meses en descubrir el mecanismo usado por los ladrones. Una vez comprobado el método de ataque, resultó fácil descubrir cuáles de las lectoras de tarjeta había sido manipulados, ya que los dispositivos adicionales insertados por los ladrones (el módulo GSM y el grabador de datos) hacía que pesasen cien gramos más que las genuinas.

Cuando se hizo público el fraude, las cifras preliminares del fraude arrojaban una cantidad de cien millones de dólares. La preocupación iba más allá del mero montante económico, ya de por sí muy grande. El jefe del *National Counterintelligence Executive* (una agencia norteamericana dedicada a la lucha contra el espionaje industrial) llegó a equiparar la hazaña con algo "*que, hace algunos años, sólo un servicio de inteligencia exterior podría haber hecho*"[38].

Las agencia de inteligencia occidentales llegaron a preocuparse seriamente por la posibilidad de que la cadena de fraude se estuviese utilizando para proporcionar fondos a al Qaeda. En noviembre de 2008, el director de la CIA Michael Hayden indicó que Osama bin Laden probablemente se escondía en el noroeste de Pakistán, información que se confirmó en mayo de 2011, cuando fuerzas especiales norteamericanas dieron muerte a bin Laden en la ciudad pakistaní de Abbotabbar. No hay pruebas de que todo o parte del dinero llegase a manos de al Qaeda, pero en cualquier caso el fraude de 2008 queda como uno de los más importantes de la historia tanto por su volumen como por su sofisticación.

El problema de asegurar la integridad de los terminales lectores de tarjetas es grave y sigue sin solución. Las propias entidades de tarjetas de crédito lo reconocen sin tapujos[39]. En un caso más reciente, la policía de Toronto (Canadá) descubrió una nueva táctica de hurto. Los defraudadores entran en una tienda, y mientras algunos de ellos distraen a los clientes y empleados, otro cambia el terminal legítimo por uno trucado, algo especialmente fácil de lograr si son del tipo inalámbrico. Al final del día, deshacen el cambio y se van a su guarida con un terminal que ha leído las bandas magnéticas y los PIN de todas las tarjetas que se han usado en él[40].

Una de las pistas que permite descubrir el terminal falso es que la tarjeta se introduce en su totalidad, no solamente la parte del chip; eso es necesario para que pueda leer y copiar la información de la banda magnética. Si eso le sucede a usted, desconfíe. Es posible que el terminal sea legítimo y lo haga para poder verificar la información de la banda magnética, pero ¿por qué arriesgarse a estas alturas? No le importe ponerse en modo paranoico. Al fin y al cabo, es su dinero.

6) CONTRAMEDIDAS

Lo cierto era que la introducción del sistema Chip+PIN (que concluyó a comienzos de 2006) no había servido para frenar las cifras de fraude en el Reino Unido. Cierto, el fraude mediante tarjetas había descendido en 2006 un 15% con respecto a 2004, pero la caída fue muy desigual según el tipo de fraude: si bien la falsificación de tarjetas disminuyó un 30%, el fraude en transacciones sin presencia (teléfono, Internet y ventas por correo) se habían disparado un 40% y suponía casi la mitad del fraude total.

Puede argumentarse que el fraude online subió porque el volumen de transacciones online también había aumentado, y los ladrones van donde está el dinero. Aun así, el fraude en transacciones no presenciales siguió aumentando. En 2008 el fraude no presencial era ya el doble que el de 2004, y lo más preocupante, el fraude por falsificación y clonado de tarjetas, tras un mínimo en 2006, volvió a crecer, alcanzando una cifra récord en 2008[41].

Las autoridades achacaron el aumento del fraude con tarjetas clonadas al hecho de que los criminales robaban los datos de las tarjetas y las usaban en otros países donde todavía se utilizaba el sistema de banda magnética. A tenor de casos como los anteriormente reseñados, era evidente que también se perpetraban dichos ataques en el Reino Unido. En cualquier caso, resultaba una evidente manifestación de un principio básico en seguridad: si fortaleces una ventana, los ladrones intentarán entrar por otra. Las cifras correspondientes al fraude telefónico y por Internet, por su parte, reflejaban que el sistema Chip+PIN, diseñado para transacciones presenciales, no estaba resultando eficaz en el entorno del comercio electrónico online. Estaba claro que había que adoptar más medidas.

En un intento por reducir el fraude con tarjetas Chip+PIN, la industria bancaria introdujo un nuevo elemento de seguridad. Se trató de un nuevo número CVV (llamado iCVV), un código electrónico insertado en el chip de la tarjeta y que siempre tiene el valor 999. De ese modo, si alguien copia la información del chip a la banda magnética e intenta usar ésta en una tarjeta clonada, el lector podrá adivinar fácilmente que se trata de una transacción fraudulenta, ya que aparecerá el 999 en lugar del CVV correspondiente, y ninguna banda magnética puede usar tal número. Esta medida estaba diseñada para evitar fraudes como los descritos por el grupo de Anderson en febrero de 2008, aunque a la vista del “caso Doctor Maligno” descrito anteriormente no parece que fuese aplicado con gran velocidad en el sistema bancario. La adopción de la mejora iCVV fue lenta, lo que no debe extrañarnos ya que cualquier cambio en un sistema tan extenso ha de ser necesariamente lento y costoso.

Cualquiera que fuese el éxito del código iCVV, resulta inútil en las transacciones

online, donde no hay lectores de chips. Decir de viva voz el PIN por teléfono es una práctica muy arriesgada puesto que cualquiera puede oírnos; la situación en Internet no es mucho mejor, con tanto troyano y capturador de contraseñas. Se hizo urgente mejorar el proceso de autenticación.

En España, la mayoría de los bancos usan una tarjeta de coordenadas, que se envía por correo o se entrega en mano en una sucursal. Cuando el cliente desea hacer una transacción con su banco online, la web le solicita un número de su tarjeta de coordenadas, uno distinto para cada transacción. De ese modo, para obtener las claves del usuario habría que capturar toda la tarjeta de coordenadas. Es un sistema válido para un entorno limitado a un banco o tienda online, pero si queremos que el cliente pueda usar su tarjeta en cualquier tienda o web en Internet, hace falta un procedimiento más general.

Una solución es el llamado Programa de Autenticación de Chip (CAP). Desarrollado por MasterCard, es una especificación dentro del esquema EMV para que las tarjetas de crédito puedan usarse en transacciones sin presencia física; VISA lo usa también, aunque le cambió el nombre a Autenticación de Contraseña Dinámica (DPA). Cualquiera que sea el nombre, se basa en un pequeño lector de tarjetas que se entrega al cliente. Cuando éste desea realizar un pago online, lo primero que hace es insertar su tarjeta en el lector. La gran diferencia es que dicho lector no está conectado a nada y no transmite información alguna, sino que se limita a mostrar números en una pequeña pantalla.

Digamos que el cliente desea comprar un libro por Internet. Cuando solicita el pago mediante tarjeta equipada con CAP, lo que hace la tienda es presentar una cadena numérica, lo que se denomina un “reto” (*challenge*). El nombre es descriptivo, porque actúa como una pregunta que solamente el usuario legítimo podrá contestar. Para producir la respuesta, el cliente inserta la tarjeta en el lector CAP, la activa mediante el PIN y teclea el reto. El lector calcula la respuesta y el cliente la introducirá en la web, que a su vez pasa los datos al emisor de la tarjeta para su verificación. Si la tarjeta es legítima, el emisor autorizará la transacción.

Lo que acabo de describir es el llamado “modo respuesta”. Existen otros dos modos: firma e identificación. En el modo de firma, el cliente introduce en el lector información adicional como la cantidad a pagar y el número de cuenta; en el modo de identificación, el lector se limita a generar una contraseña de uso único, que puede usarse para acceder a la web del banco.

En principio, el sistema CAP es flexible y muy útil. Como han podido ver, la tarjeta no filtra ninguna información al exterior, y en ningún momento existe la posibilidad de capturar y transmitir el PIN. Es un sistema cuyo uso no está limitado a una tienda o web bancaria concreta, sino que en principio puede utilizarse para cualquier transacción online, incluso por teléfono.

Por supuesto, tampoco este sistema es perfecto. El mayor problema potencial proviene de la "experiencia de usuario". Si el CAP es incómodo, los usuarios tenderán a no usarlo, o a hacerlo de forma inadecuada. Un posible peligro consiste en que, para facilitar el paso de los datos de reto/respuesta, algún banco decida facilitar la tarea al usuario realizando una implementación en software que envíe directamente la información del lector CAP al ordenador, lo que abriría la puerta a las típicas vulnerabilidades derivadas de virus, troyanos, capturadores de contraseñas, sistemas operativos inseguros, etc. Una segunda vía de ataque sería la manipulación de los lectores CAP, cosa que tras el ataque "Doctor Maligno" de 2008 se demostró que no solamente es factible sino también muy rentable.

No hace falta irse tan lejos. En marzo de 2009, nuestros conocidos de Cambridge Saar Drimer, Steven J. Murdoch y Ross Anderson publicaron un artículo cuyo título podemos traducir como "*Optimizados para fallar: lectores de tarjetas para banca online*"[42] La principal queja de los autores es que el sistema CAP está optimizado para reducir tanto las molestias a los usuarios (haciéndoles teclear lo menos posible) como los costes para el banco, y eso afecta a la seguridad:

"*El fallecido Roger Needham dijo una vez que la optimización es el proceso de tomar algo que funciona y convertirlo en algo que casi funciona pero es más barato… el sistema [CAP] ha sido optimizado hasta la muerte*".

Algunas de las vulnerabilidades del artículo son, reconozcámoslo, poco evidentes, y no se refieren al protocolo técnico en sí. A pesar de ello, detectaron al menos un fallo en el sistema, consistente en que un campo numérico tiene dos usos distintos, según el modo que estemos utilizando. Como resultado, si un cliente efectúa una transacción con valor cero, la información puede utilizarse en modo respuesta. No es algo que suceda todos los días, pero imagine al malo de turno convenciendo al usuario legítimo para que haga una prueba del sistema. "Vamos, intente una transferencia, ponga cero libras y así podrá comprobar si todo va bien". El usuario legítimo, tranquilizado por el hecho de que no está arriesgando un solo céntimo, accede a la propuesta, pero no sabe que al mismo tiempo su "amigo" está aprovechando para establecer una comunicación en modo respuesta con fines fraudulentos. Es un pequeño detalle que, por supuesto, no conoce el usuario medio, una puerta abierta al fraude cuyo fin inicial era optimizar el sistema y cuya víctima será la misma de siempre: el cliente.

Durante el año 2009 la industria británica de tarjetas vio con alivio cómo las medidas puestas en funcionamiento (el programa de autenticación CAP, el iCVV y mejores herramientas de detección de fraude) habían reducido el fraude en tarjetas bancarias casi un 30% respecto a 2008. El fraude mediante tarjetas clonadas había caído a la mitad, y el correspondiente a tarjetas robadas o perdidas era poco más que la mitad del existente en 2005. Resultaba asimismo revelador el hecho de que el

fraude en tarjetas de crédito británicas realizado fuera de sus fronteras había caído asimismo a la mitad. Las cifras de fraude habían caído al nivel de 2005 en términos absolutos[43]. Todo parecía indicar que las vías tradicionales de fraude contra los sistemas Chip+PIN estaban cerrándose; y también algunas de las nuevas, porque pronto se encontró un fallo que permitía un ataque en ciertas condiciones.

Para entender este nuevo fallo, recordemos que el protocolo EMV consta de tres fases: autenticación de la tarjeta, verificación del cliente y autorización de la transacción. Puesto que el terminal punto de venta puede estar fuera de línea, se descartaron las soluciones en las que el terminal necesita estar en comunicación con el emisor de la tarjeta, y por eso las tres fases pueden hacerse a base de intercambios electrónicos entre la tarjeta y el terminal exclusivamente.

La primera fase permite al terminal (el lector de la tarjeta) verificar si la tarjeta ha sido o no alterada, y comprobar cuál es el banco emisor de la tarjeta. Para ello, la tarjeta incorpora una firma digital RSA. Si el resultado de la autenticación es positivo, se negocia la segunda fase: el cliente introduce su PIN en el terminal, el cual pregunta a la tarjeta si es el PIN correcto. Según el resultado de la comprobación, el proceso continúa o es interrumpido en este punto.

En la tercera fase, el terminal pregunta al emisor de la tarjeta si autoriza la transacción. Si éste dice que sí, el proceso continúa, y al final de todo se genera un certificado de transacción del que se hacen dos copias, una para el cliente y otra para el banco emisor. Ahora bien, es posible que el terminal esté fuera de línea y no pueda comunicarse con el emisor. En ese caso puede autorizarse la transacción de modo "offline," en cuyo caso la tarjeta enviará al terminal el equivalente electrónico de un "bueno, vale, apúntamelo," y cuando se restablezcan las comunicaciones el terminal procederá a cobrarse del banco. La decisión de proceder sin saber si la transacción ha sido autorizada la toma la propia tarjeta, que ha sido programada por el emisor. Es evidente que permitir la transacción fuera de línea disminuye la seguridad del sistema, porque nos pone en manos de defraudadores, pero permite al usuario legítimo completar la transacción en condiciones de desconexión.

Cuando se comenzó a utilizar EMV, la autenticación se efectuaba mediante una variante conocida como SDA (autenticación estática de datos), en la cual la firma digital era estática, esto es, nunca cambiaba. Eso permite un ataque de repetición, en el que la tarjeta legítima es leída y copiada en una segunda tarjeta. Como la firma digital no cambia y tiene un valor constante sea cual sea la transacción, el clonado no es detectado y la fase primera (autenticación) se cumple con éxito. La segunda fase (verificación) es aún más sencilla. El malo de turno se limita a programar su tarjeta clonada de forma que siempre siga sí cada vez que alguien le pregunte "¿es este el PIN correcto?" Eso le granjeó a este tipo de tarjeta clonada el apodo de "tarjeta sí" (*yes card*). Queda por cumplir la tercera fase, para lo cual no hay más que buscar un

terminal que no tenga conexión con la red bancaria. Para cuando el banco pueda comprobar si la transacción es válida o no, ésta ya se ha llevado a cabo horas antes.

Afortunadamente, la flexibilidad del estándar EMV permitió solventar este problema. La solución se llama DDA (autenticación dinámica de datos), y consiste en dotar a la tarjeta de un sistema criptográfico de clave pública. Para negociar la autenticación, el terminal enviará a la tarjeta un número aleatorio llamado *nonce*. La tarjeta deberá firmar digitalmente ese número con su clave privada y devolverlo al terminal. El terminal, por su parte, verificará la firma con la correspondiente clave pública que la tarjeta le acaba de enviar. Como el *nonce* es distinto para cada transacción, la firma también lo será, lo que impide poder usar la firma digital de una transacción anterior. Y, puesto que las claves son únicas, una tarjeta que haya sido clonada o robada puede ser detectada con mucha mayor rapidez. Una segunda modificación, llamada CDA (autenticación combinada de datos), efectúa una operación de firma digital similar en la tercera fase, la de autenticación de la transacción.

Lo triste del asunto es que, aunque las tarjetas con chip podían haber llevado la autenticación DDA activada dese el principio, no todos los países la implantaron por defecto. Una tarjeta con DDA debe llevar un cripto-procesador, y cuesta más caro de producir que uno con SDA. El coste no es muy elevado, digamos un euro por tarjeta, y los terminales y cajeros automáticos no tendrían que ser modificados, ya que DDA es parte del paquete de protocolos incluidos en el sistema EMV. El coste extra de equipar con autenticación DDA todas las tarjetas emitidas en un país representa una fracción del ahorro que se lograría con la reducción de costes por fraude que ello conllevaría.

Algunos países como Suiza, Alemania y Francia se decidieron desde el principio por DDA o bien forzaron su adopción temprana, sacrificando mayores costes iniciales a cambio de una mayor seguridad para sus clientes a largo plazo. Otros, como el Reino Unido, prefirieron apostar por los menores costes del SDA, ahorrando con ello en el chocolate del loro, y por cruzar los dedos con la esperanza de que el fraude no se resintiese por ello. A tenor de las cifras de fraude, la apuesta, que algunos expertos llamaron “pragmática”[44] no les salió bien. Eso sí, la práctica de “el cliente es culpable” les permitió salirse con la suya y no costear el fraude que podían haber evitado, aunque queda por evaluar el coste adicional en mala publicidad y caída de la confianza de sus clientes.

En cualquier caso, llegó un momento en que incluso los bancos ingleses se dieron cuenta de lo poco inteligente que resultaba mantenerse anclado en la opción insegura de un sistema seguro, y en 2009 comenzaron la migración hacia el DDA. Para comienzos del año 2010, dos tercios de todas las tarjetas EMV de Europa tenían ya el nuevo sistema DDA. En Asia oriental, donde EMV aún no se había impuesto frente a

otro tipo de tarjetas, la adopción de la autenticación DDA fue menor, 25-30%, cifra similar a la de África. El continente americano era el más retrasado en este respecto, con un 4-6% de uso de la DDA[45]. Estas cifras tan bajas se explican si recordamos que Estados Unidos ha sido un país muy retrasado en la adopción del estándar EMV.

Pero incluso las grandes empresas de crédito norteamericanas (que a fin de cuentas habían creado EMV) reconocieron la vulnerabilidad de la autenticación SDA y la necesidad de acabar con ella para reducir el fraude. En junio de 2009, VISA y MasterCard decidieron que todas las tarjetas EMV emitidas a partir del 1 de enero de 2011 incorporarían el sistema DDA.

No parece haber datos sobre el tipo de autenticación utilizado en España (o, al menos, yo reconozco humildemente no haberla encontrado). Sin embargo, la lenta implantación del sistema EMV en España hace suponer que el impacto de la vulnerabilidad derivado de la autenticación estática de datos (SDA) fue pequeña. En junio de 2009, cuando VISA y MasterCard dieron su aviso contra el uso de SDA, menos del 15% de las tarjetas que circulaban en España tenían chip[46]. No hay cifras acerca del grado de incorporación del sistema DDA en esas tarjetas. En 2005, Caixa Penedés ya anunciaba la adopción de soluciones EMV con DDA[47]. En el extremo opuesto, la empresa de seguridad Gemalto presentó las nuevas tarjetas EMV+DDA de Diners Club España… en febrero de 2012[48].

7) EL ATAQUE “FISH AND CHIPS”

Al comenzar el año 2010, la cuarta parte de las tarjetas españolas eran del tipo EMV. Este bajo porcentaje nos situaba muy por detrás de la mayoría de países europeos. Lo sorprendente, de hecho, era que estuviésemos en un nivel tan alto. La banca española fiaba más en las soluciones tradicionales basadas en identificación mediante DNI, y consideraba que los altos costes de un sistema diseñado para solucionar los problemas de fraude que se daban en otros países no se verían justificados.

Eso pudo ser cierto en su momento. A finales de 2001, las casi setenta millones de tarjetas bancarias en circulación en España movían 108 000 millones de euros, y el fraude por falsificación o robo se estimó en sesenta millones de euros, apenas un 0,06% del total[49]. Las cantidad fue ascendiendo en valor absoluto, y para 2006 se había triplicado hasta un total de 183 millones de euros, una cifra solamente superada por Reino Unido (649 millones) y Francia (262 millones). Un buen porcentaje de los fraudes parecen deberse a tarjetas emitidas en el extranjero y utilizadas en España, donde el uso de tarjeta con banda magnética era la norma habitual. Entre 2006 y 2009, la cantidad global defraudada en nuestro país aumentó otro 17%.

A la vista de estas cifras, incluso los altos costos derivados de la conversión al estándar EMV se convierten en un mal necesario si se quieren reducir las cifras de fraude. Pronto se vieron resultados, ya que para 2011, el fraude con tarjetas había disminuido un 8% respeto a 2009[50]. La reducción en términos absolutos era pequeña, menos de veinte millones de euros, y hay muchos otros factores que impiden atribuir al nuevo sistema el mérito en la lucha contra el fraude; pero es verosímil concluir que, de no haberse realizado el cambio, las cifras de fraude serían en este momento mucho mayores. En este sentido, el esquema EMV evitó males mayores, lo que ya es motivo suficiente para justificar los costes de su implantación.

Hubo una segunda causa que impulsó la transición, tanto en España como en otros países europeos. Se trata de algo llamado Zona Única de Pagos en Euros, o SEPA (*Single Euro Payments Area*), un intento de la Unión Europea para establecer un sistema común de medios de pago. Durante 2009, los países del SEPA debían transcribir a sus legislaciones nacionales la Directiva 2007/64/CE sobre servicios de pago[51]. España hizo lo propio el 14 de noviembre de 2009 con la publicación en el Boletín Oficial del Estado de la Ley 16/2009 de Servicios de Pago[52]. Básicamente, mantenía el carácter garantista respecto a los derechos de los clientes. En caso de que el cliente niegue haber realizado una operación, el banco deberá demostrar de modo fehaciente lo contrario o bien devolver todo el dinero. En cualquier caso (salvo fraude manifiesto o negligencia grave), se estableció un límite de 150 euros para las pérdidas en caso de sustracción o extravío de la tarjeta.

Esto estaba en consonancia con una especie de aceptación tácita, según la cual la banca española admite y asume que la seguridad de sus sistemas no es total (algo que choca con las pretenciosas afirmaciones de sus homólogos británicos). Cuando el BBVA, adelantándose un mes al BOE, anunció a bombo y platillo sus nuevas tarjetas con sistema EMV, hizo hincapié en el concepto de "mayor seguridad," lo que implícitamente lleva el mensaje de que la seguridad es un proceso, no un producto[18].

En Reino Unido, donde la implantación del sistema Chip+PIN era del 100%, las perspectivas en torno a la seguridad eran optimistas. El sistema de protocolos EMV, aunque falible, era lo bastante flexible como para responder a vulnerabilidades como las que hemos mencionado en páginas anteriores. La migración hacia el sistema de autenticación dinámica de datos DDA, que había comenzado en 2009, continuaría de forma rápida y masiva durante todo 2010, con lo que la industria bancaria británica se las prometía muy felices.

En el apartado negativo (visto desde los ojos de la industria bancaria, por supuesto), la regla de "en caso de disputa, el cliente es culpable" se quebró finalmente. Resulta increíble, pero durante años la política del sistema bancario en cuestiones de fraude estaba regulada no por leyes o decretos, sino por un código bancario (*Banking Code*) de autorregulación, un conjunto de prácticas que los bancos aceptaban voluntariamente, sin imposición legal alguna.

Eso cambió el 1 de noviembre de 2009, cuando la adaptación a la Zona única de Pagos en Euros forzó la entrada en vigor de la Reglamentación de Servicios de Pago (*Payment Services Regulations*). Entre otros cambios, revocó la asignación automática de la culpa, y la consiguiente obligación de pagar, al cliente en casos de fraude. Desde entonces, las transacciones que el cliente estime como no autorizadas deberán serle reembolsadas[53]. El banco puede entablar acciones legales si lo estima oportuno, pero será él quien deba probar que el cliente miente. Es decir, la carga de la prueba recae ahora sobre el banco, y la presunción de inocencia salvo prueba en contrario se asigna "por defecto" al cliente[54]. Y no bastará con mostrar un recibo con las palabras "verificado por PIN" impresas.

En su ventaja, Reino Unido contaba con amplia experiencia en el sistema de tarjetas EMV. Así pues, una de cal y una de arena para la banca inglesa en el 2010, con la ventaja de que la de cal estaba más que colmada. Eso sí, suponiendo que los amigos de la Universidad de Cambridge se estuviesen callados y no encontrasen más fallos de seguridad. A estas alturas no les sorprenderá si les digo que en febrero de 2010 presentaron el estudio más demoledor sobre la seguridad del sistema. En este caso no se trató de tarjetas con banda magnética clonada sino de verdaderos agujeros de seguridad en el protocolo EMV. Su título lo dice todo: "*Chip y PIN está roto*"[55]. Veamos en qué consiste exactamente, y juzgue usted si exageraban o no.

En los sistemas complejos el fallo, como el diablo, suele estar en los detalles, y este caso no fue una excepción. El detalle se oculta en la fase dos, la de verificación del cliente. Antes dije que el cliente inserta el PIN y la tarjeta lo contrasta con el que debería ser. Así es el método estándar, pero no es el único posible. En realidad, el protocolo EMV permite que el terminal escoja el método de verificación más adecuado para cada ocasión. No es lo mismo pagar una cena en un restaurante que un Bugatti Veyron en el concesionario, de forma que a veces habrá que usar el PIN, otras veces la firma manuscrita, o bien ambas. También hay que regular qué pasa si el proceso de autenticación falla. ¿Qué hacemos: interrumpir la transacción, comenzar otra nueva o avisar a la policía?

Cada terminal establece sus condiciones de verificación por medio de un archivo incorporado llamado CVM (Método de Verificación del Cliente). Esto no sólo resulta cómodo para establecer los límites de riesgo del cajero o la empresa que tiene el terminal con el lector de la tarjeta, sino que permite ofrecerlo como servicio a los clientes. Por ejemplo, un cliente con problemas de visión o de memoria puede tener problemas con el PIN, de forma que hay tarjetas en las que la verificación se realiza mediante firma manuscrita. O puede permitir que, si falla la verificación mediante PIN, se intente por segunda vez mediante una firma manuscrita. Cada entidad o banco puede establecer las condiciones que desee para cada tarjeta de cada cliente en particular, y así el sistema gana mucho en flexibilidad.

Veamos un ejemplo del funcionamiento de la fase dos. Hoy es mi cumpleaños, así que he invitado a mi familia a un buen asador. Tras una excelente velada con chuletones y doradas a la sal, pido la cuenta. El camarero toma el lector de tarjetas portátil, me lo acerca a la mesa y yo introduzco mi tarjeta. El CVM de su lector dice que admite todos los tipos de verificación (PIN o firma manuscrita), pero como la cuenta ha excedido de cierto límite exige la introducción del PIN. Lo tecleo en el terminal, y éste lo envía a la tarjeta. Ahora la tarjeta lo compara con el que tiene guardado en el chip.

Pueden pasar dos cosas. Si el PIN es el correcto, la tarjeta devuelve al terminal el código *0x9000*, que es la forma que tiene de decir “de acuerdo, adelante”. Pero puedo haberme equivocado al introducir el PIN, en cuyo caso la tarjeta responde con el código *0x63Cv*, donde *v* es el número de intentos que me permite antes de cerrar el proceso y llamar a la policía. En este caso estoy algo achispado, y me salta el primer aviso. Felizmente, mi segundo intento tiene éxito. El camarero me da el recibo, donde aparecerá probablemente la frase “verificado con PIN,” me invita a un chupito de cortesía y me desea buenas noches. Y feliz cumpleaños, por supuesto.

¿Dónde está aquí el problema? Pues en el hecho de que ese código intercambiado entre el terminal y la tarjeta no está autenticado, lo que significa que podemos alterarlo impunemente. Para ello, insertamos lo que en lenguaje criptográfico se llama

un “hombre interpuesto” (*man-in-the-middle*), una especie de metomentodo que se ha introducido dentro del sistema para subvertirlo. Se supone que su intrusión debería ser detectada, pero como veremos, resulta que la detección falla.

Nuestro metomentodo viene equipado con un lector de tarjetas, un PC y una tarjeta falsa, así como la capacidad de robar (o tomar prestada sin que su dueño se entere) una tarjeta legítima. El proceso comienza cuando se inserte la tarjeta falsa en el terminal de venta. Lo primero que hacemos es teclear un PIN cualquiera. El terminal interroga a la tarjeta falsa sobre la validez del número, y ésta pasa la solicitud al PC. El ordenador, a su vez, pregunta a la tarjeta robada si autoriza la transacción. No importa cuál sea la respuesta de dicha tarjeta, solamente el hecho de que se le ha preguntado.

A continuación, el PC envía a la tarjeta falsa el código *0x9000*, indicativo de “sí, este es el PIN correcto” Ese código llega hasta la tarjeta falsa, que le dice al terminal “sí este es el PIN correcto”. En otras palabras, hemos logrado interceptar la comunicación entre el terminal y la tarjeta, de forma que cuando aquél diga “¿es este el PIN correcto?,” siempre oiga la respuesta “sí, lo es”. Ya está. La verificación está hecha, a pesar de que la tarjeta legítima es robada y no sabemos su PIN. Ahora, a comprar a manos llenas, que el titular de la tarjeta paga.

Pero espere un poco, me dirá usted, no puede ser tan fácil. ¿Es que no hay mecanismos para detectar esta jugarreta? Lo cierto es que sí los hay, pero son hábilmente sorteados. Como en la película *La Gran Evasión*, los guardias son engañados para ver lo que los prisioneros quieren que vean. En primer lugar, la tarjeta legítima robada. Uno de los mecanismos que lleva es un dispositivo que cuenta cuántas veces se ha introducido el PIN. De ese modo puede, entre otras cosas, contabilizar cuántas veces ha intentado el cliente entrar el número correcto. Pero como la verificación no está autenticada, el PC del metomentodo puede engañar a la tarjeta legítima y hacerle creer que el proceso de verificación no necesita PIN. Si fuese humana, la tarjeta se encogería de hombros y murmuraría algo como “de acuerdo, terminal, tú mismo,” y lo autorizaría. Puesto que la petición del PC indica “verificación sin PIN,” no se activa el contador que indica el número de veces que se ha utilizado el PIN. En lo que a la tarjeta concierne, la verificación con PIN nunca ha tenido lugar, y si le preguntasen solamente podría responder algo como “no me consta haber autorizado transacciones con PIN, así que por eliminación debe de haber sido con firma manuscrita”.

En segundo lugar, el banco. Supuestamente, tras la fase de verificación, el terminal y el banco intercambiarán información. Entonces debería detectarse el ataque, porque el terminal ha comenzado una verificación con PIN pero la tarjeta ha llevado a cabo una verificación sin PIN. ¡Pues tampoco se detecta! El motivo es que, cuando una verificación ha fallado, el terminal indica rápidamente al banco que ha

habido un fallo y le da indicación de cuál ha sido este fallo: el PIN se introdujo mal más veces de lo autorizado, o bien se solicitó y no hubo respuesta; pero cuando la verificación ha tenido éxito ¡el banco no tiene forma de saber qué método se ha utilizado!

Como en una comedia de enredo, nuestro atacante metomentodo ha logrado sembrar la confusión generando un diálogo de besugos. La tarjeta legítima cree haber autorizado una transacción mediante firma manuscrita; el terminal cree haber autorizado una transacción mediante PIN; y el banco no tiene idea de qué tipo de verificación se ha hecho, ni le importa lo más mínimo. En cuanto al dueño de la tienda donde el terminal estaba instalado, sólo recordará (si es que lo recuerda) que alguien insertó un PIN.

Imagínense el jaleo cuando llegue el inevitable cara a cara. La transacción ha sido autorizada, y según la tarjeta no se ha producido una verificación con PIN, así que el banco concluirá que se ha usado la banda magnética y la firma. Si el titular de la tarjeta legítima no denunció a tiempo su desaparición, el banco puede hacerle responsable; y si lo hizo, puede culpar al dueño de la tienda. El enredo está servido, y mientras tanto el metomentodo se ha quitado de en medio, listo para probar fortuna con otra tarjeta robada.

Los investigadores de Cambridge se arriesgaron mucho al afirmar lapidariamente que “Chip y PIN está roto”. Su razonamiento, con el que por supuesto podemos estar de acuerdo o no, es que:

“*Nosotros medimos el éxito de Chip+PIN por sus dos objetivos fundamentales: primero, evitar la falsificación de tarjetas que incorporen el chip, y segundo evitar que se puedan usar tarjetas perdidas y robadas sin el PIN. Como pueden usarse tarjetas robadas sin necesidad de saber el PIN, según nuestra definición Chip+PIN está roto*”.

Este ataque, al que yo llamo “fish and chips” (no me pregunten por que, soy así de raro), es mucho peor que el que vimos en el apartado anterior por dos motivos. En primer lugar, ahora el ataque funciona incluso si el terminal se encuentra online; y en segundo, la solución dada para resolver el ataque anterior (pasar a autenticación DDA) no resuelve el problema actual. Existen algunas soluciones, pero son parches y de eficacia no clara.

Cuando los autores escribieron su artículo, lo hicieron circular entre responsables de la industria y las entidades reguladoras durante tres meses antes de darle publicidad. No recibieron ninguna respuesta[56]. Puesto que la industria no estaba por la labor, decidieron hacer una prueba en vivo y en directo, en condiciones reales tales que nadie pudiese ponerlo en duda o utilizar la típica excusa de “es un ataque teórico, no tiene consecuencias prácticas”. Para ello, organizaron una prueba ante un equipo de reporteros de la BBC, quienes proporcionaron las tarjetas “robadas,” dos de débito

y dos de crédito, todas de bancos distintos. Saar Drimel, uno de los firmantes del artículo, escondió todo el "equipo de metomentodo" en una mochila. Un cable salía de ésta, atravesaba la manga del "defraudador" y se conectaba mediante un cable con la tarjeta fraudulenta que éste llevaba en la mano.

La prueba se hizo en la propia cafetería de la Universidad de Cambridge. Drimel insertó la tarjeta falsa en el terminal, tecleó el PIN 0000, y la transacción fue autorizada sin problemas. Funcionó, y el recibo indicaba claramente "verificado por PIN". Ahora Saar Drimel era el ilegítimo propietario de una pequeña botella de agua con cargo a una tarjeta supuestamente protegida por el sistema Chip+PIN. Por supuesto, la Universidad de Cambridge autorizó el intento de fraude, y suponemos que el señor Drimel reembolsó los seis euros de la compra al legítimo propietario de la tarjeta.

El reportaje se emitió el 11 de febrero de 2010 en el programa *Newsnight* del canal BBC2[57]. Los bancos cuyas tarjetas de crédito habían sido trucadas publicaron comunicados en los que indicaban que el problema era común al sistema EMV y no tenía nada que ver con las prácticas de un banco en particular; lo cual era rigurosamente cierto. En cuanto a la *UK Cards Association* (representante de la banca en temas de sistemas de pago), conocida antes como APACS, si bien reconocía la existencia del ataque como poco más que una variante de un ataque anterior, negaba que ese fuese el final de Chip+PIN y afirmaba que "*ni la industria bancaria ni la policía tienen evidencia alguna de que algún criminal tenga la capacidad de desplegar ataques tan sofisticados*"[58].

Al parecer, el término "sofisticado" tiene un significado muy particular para ellos. Como prueba de lo que digo, permítanme compartir con ustedes un pequeño cotilleo. Cuando Ross Anderson publicó una entrada en su blog explicando el ataque[59], apareció un comentario negativo de un tal *Scrutineer*; su opinión era que el artículo dejaba mucho que desear, la técnica subyacente no era gran cosa, no había pruebas concluyentes[60]. Evidentemente, no sabía con quién se la estaba jugando. Ross Anderson tardó poco en descubrir que la IP usada por *Scrutineer* correspondía a una dirección de APACS[61]. Anderson no quiso dar muchos detalles sobre la identidad de *Scrutineer*, pero si tiene usted curiosidad, pásese por[62]. Ah, casi se me olvida decirlo: APACS es un nombre antiguo de la actual… *UK Cards Association*. Sutileza en acción[63].

8) FELIZ NAVIDAD

El año 2010 tocaba a su fin, y la industria bancaria se recuperaba de los varapalos recibidos. Podían sacar pecho, ya que a despecho de las afirmaciones de Ross Anderson y su equipo sobre que “Chip y PIN está roto,” las cifras de fraude por tarjetas seguían bajando. El fraude mediante tarjetas clonadas o robadas había caído otro 40%, y soluciones como *SecureCode* de MasterCard o *Verified by VISA* estaban reduciendo el impacto del fraude en las compras online.

Pero la “patrulla X” de la Universidad de Cambridge estaba allí para fastidiarles el día. Quizá les sorprenda la fijación que tiene el grupo de Ross Anderson con las tarjetas EMV. Que yo sepa, no es nada personal, sencillamente, es su trabajo. No se pueden establecer sistemas informáticos seguros si no se analizan y destripan a conciencia, de igual modo que no podemos luchar contra el SIDA sin conocer su mecanismo de actuación. El grupo de Cambridge continuó trabajando para mostrar que las vulnerabilidades que habían descubierto eran algo más que un ejercicio académico. Existían y eran potencialmente peligrosas. Ellos mismos desarrollaron una solución: si un metomentodo atacante puede insertarse en el sistema para robar información, otro metomentodo “amigo” podría ayudar a protegerlo. Sería una especie de verificador de confianza, presta a advertir de cualquier acción no autorizada.

La teoría estaba ya desarrollada, pero había que comprobar la validez práctica de la solución. El encargado de la tarea fue un estudiante de doctorado llamado Omar Choudary, nacido en Madrid y titulado por la Universidad Politécnica de Rumanía. Su tesis doctoral trata precisamente de cómo se puede crear, con medios técnicos modestos, un “interceptor,” es decir, un dispositivo que pudiese proteger al cliente contra los ataques que ya se conocían. El dispositivo, al que llamaron *Smart Card Detective*, sería portátil, asequible, fácil de manejar y de dominio público.

La información del proyecto *Smart Card Detective* (SMD) es accesible desde Internet. Se puede descargar todo el software en código libre[64], adquirir el hardware por poco más de quinientos euros[65], consultar documentación adicional[66] y hasta la tesis doctoral de Choudary[67]; quien, por cierto, afirma que el dispositivo puede construirse por poco más de cien euros y que podría hacerse una versión industrial por unos veinte.

Una de las opciones en el SMD permite efectuar el ataque “fish ´n chips”. Esto no es parte de la solución de seguridad, ni es algo que se debiese incorporar a un SMD vendido comercialmente. De hecho, el autor tuvo mucho cuidado de no incluir esta opción en la documentación pública por motivos de seguridad. Se trata de una prueba de concepto, una forma de demostrar que dicho ataque es factible con equipo electrónico sencillo y barato. Choudary logró comprobar la validez del ataque en la

cafetería de la Universidad de Cambridge, tras haber pedido permiso. También comprobó que funcionaba en los pequeños lectores del tipo Programa de Autenticación de Chip (CAP).

A continuación, hizo la prueba en algunas tiendas de la ciudad, y sólo tras lograr el éxito advirtió del hecho a los tenderos. Uno de ellos, sorprendentemente, dijo que ese tipo de ataques ocurrían con frecuencia, aproximadamente una vez por semana durante las navidades. Esto era un detalle preocupante, porque una de dos, o alguno de los investigadores del grupo de Ross Anderson se dedica a complementar la beca con compras gratuitas, o bien las bandas criminales ya se saben el truco y lo aplican de forma habitual.

En general, el dispositivo de Choudary muestra que es posible utilizar el SMD como mecanismo de seguridad adicional, lo que puede ser útil en transacciones electrónicas cuantiosas. Tal vez sea más engorroso para llevar de forma habitual, pero sospecho que sólo es cuestión de mejorar la tecnología, bajar el precio por unidad y concienciar a los clientes de su conveniencia. En cuanto a la industria bancaria, su utilidad para encontrar vulnerabilidades no tiene precio.

Omar Choudary presentó su tesis en junio de 2010. En octubre fue entrevistado por un equipo de Canal+ (Francia). El 2 de diciembre, una segunda entrevista fue televisada por el programa *Rip Off Britain*, del canal BBC1. Finalmente, la industria bancaria británica decidió tomar medidas al respecto, aunque no precisamente para proteger a sus clientes. Según su punto de vista, una cosa es publicar ataques teóricos o probar un ensayo práctico en condiciones controladas ante un equipo de televisión, y otra muy distinta regalar los secretos del ramo a cualquiera que tenga una conexión a Internet.

El 1 de diciembre, la UK *Cards Association* (UKCA) envió una carta a la Universidad de Cambridge. Afirmando que se habían sobrepasado "*las fronteras de lo que constituye una revelación responsable*," la UKCA sostenía que la tesis de Choudary, hecha pública, no solamente facilitaba la tarea a los amigos de lo ajeno sino que también "*podría minar la confianza del público en [el sistema de tarjetas de pago]*". No está claro qué les preocupaba más, si la pérdida económica o quedar como unos idiotas ante sus clientes. En consecuencia, solicitaban que los datos de la investigación se retirasen del dominio público[68].

En este punto, y como profesor universitario, debo declarar con tristeza que, si esto hubiese sucedido en España, todos los responsables académicos, desde el director de tesis hasta el rector, habrían competido en rapidez y diligencia a la hora de retirar la información, pedir disculpas por la torpeza y culpar al pobre investigador. Pero estamos hablando del Reino Unido y de la Universidad de Cambridge. El día de nochebuena, Ross Anderson respondió con una contundente carta[69]. Comenzó dejando claro que ciertas cosas, sencillamente, no son aceptables en el Reino Unido:

"*Parece creer usted que podemos censurar la tesis de un estudiante, legal y en dominio público, simplemente porque intereses poderosos lo encuentran conveniente. Esto muestra una concepción profundamente errónea de lo que son las universidades y de cómo trabajamos. Cambridge es la Universidad de Erasmo, de Newton y de Darwin; censurar escritos que ofenden a los poderosos es ofensivo para nuestros más profundos valores. Así pues, aunque la decisión de poner la tesis online fue de Omar, no tenemos más opción que respaldarlo. Eso sería así ¡incluso si no estuviésemos de acuerdo con el material! En consecuencia, he autorizado que la tesis sea publicada como Informe Técnico del Laboratorio de Informática. Eso hará más fácil que la gente lo encuentre y cite, y hará que su presencia en nuestra web sea permanente*".

A continuación, dejó las cosas bien claras. Ni la tesis de Omar contenía información novedosa, ni incluyó la información sobre el ataque "fish ´n chips" en el código fuente de su dispositivo, ni estaban minando la confianza del público. En este punto, Anderson fue abrasivo en su respuesta:

"*Lo que dará confianza al sistema de pagos es la prueba de que los bancos son francos y honestos al admitir sus debilidades, y diligentes al poner los remedios necesarios. Su carta muestra que, por el contrario, sus bancos miembros hacen lamentables esfuerzos por denigrar el trabajo de los que se encuentran fuera de su club, y de hecho los censuran*".

Mi parte favorita es el párrafo final, cargada de fina ironía británica: "*Me alegro de constatar en su carta que el ataque ya no funciona, y de que la industria haya conseguido tratar por fin con este problema de seguridad, aunque se tomaron su tiempo tras la revelación original allá por 2009*".

No sé qué cara pondrían en la UKCA al leer la carta de Ross Anderson, pero me encantaría que alguien hubiese sacado una foto. Lo cierto es que este ataque a la integridad de una de las más prestigiosas instituciones académicas del país les costó cara. Los principales medios de comunicación recogieron la noticia[70]. El intento de censura contra el trabajo de Choudary fue visto como la constatación de que los emisores de tarjetas tenían más empeño en ocultar los trapos sucios que en lavarlos. Ni que decir tiene que las vulnerabilidades del sistema EMV se hicieron más conocidas que nunca. Sólo hubo consuelo para un banco. Por lo visto, Barclays contactó con el grupo de Cambridge no para amenazarlos, sino para pedir una solución. La obtuvo, y gracias a eso las tarjetas emitidas por Barclays ya son seguras contra el ataque "fish ´n chips".

La UKCA prosiguió con sus esfuerzos de censura, y en abril de 2011 envió una segunda carta, más amenazadora que la primera. En lugar de solicitar amablemente la retirada de materiales que pudieran ayudar a los criminales, hizo hincapié es que esa conducta rozaba lo ilegal, ya que podría considerarse ayuda a la comisión de un

delito. En consecuencia, reiteraba la necesidad de que la Universidad de Cambridge se adhiriese a una política de “revelación responsable,” so pena de ver dañada su reputación[71].

La respuesta de Anderson se hizo esperar hasta el 2 de diciembre, pero valió la pena. Al leerla, la primera palabra que me viene a la mente es “enfurecido”. Harto de tanta amenaza, pasó al contraataque, enumerando algunas de las veces en las que la UKCA declaró enfáticamente que los sistemas de seguridad de la banca eran inmejorables. A continuación, acusó a la propia UKCA de actitud delictiva al “*efectuar declaraciones falsas con resultado de ganancias propias y ajenas*”. Terminaba exigiendo una disculpa a su grupo y a las víctimas de fraude de tarjetas que fueron acusadas de fraude[72]. No consta que esas disculpas se hayan llevado a efecto. En mi opinión, tardarán en volver a meterse con él.

9) EPÍLOGO (O CASI)

En la actualidad, Europa concentra la mitad de las tarjetas EMV del mundo, con una implantación que llega al 85% del total de tarjetas bancarias en su área[73]. Uno de los motivos para su éxito ha sido la creación de la Zona Única de Pagos en Euros, o SEPA, donde EMV se ha convertido en un estándar *de facto*. España ha recuperado el terreno perdido a comienzos del siglo, y a comienzos de 2012 el 90% de las tarjetas y la totalidad de los cajeros automáticos había migrado al sistema del chip y el PIN[46].

Otras regiones están todavía en fase de transición. En la zona China/India/Japón, probablemente la más poblada y económicamente activa del mundo, sólo el 28% de las tarjetas son del tipo EMV. Entre los dos mundos se cuenta la zona de América (salvo EEUU), con un 48%. En cuanto a Estados Unidos, donde EMVCo no proporciona cifras, la implantación es más lenta. Los ciudadanos norteamericanos no parecen necesitar las nuevas tarjetas con chip. Sin embargo, también allí las principales entidades de tarjetas EMV tienen sus planes de implantación. Después de todo, ya tienen el sistema funcionando en Europa, así que aplicarlo en otros países les resultará a la larga más económico, por no hablar de más seguro.

Definitivamente, parece que el chip y el PIN han llegado para quedarse. Eso incrementará las presiones para que las especificaciones EMV sean actualizadas, cerrando las actuales vulnerabilidades y asegurando los futuros sistemas de pago online que puedan desarrollarse. La EMVCo está trabajando para actualizar los protocolos criptográficos de las tarjetas EMV. Los algoritmos de criptografía asimétrica RSA usados en la actualidad son fiables pero lentos y exigentes desde el punto de vista computacional, y por eso se está considerando la posibilidad de sustituirlos en el futuro por el sistema de Criptografía de Curva Elíptica, mucho más eficiente[74].

EMV es el sistema de seguridad para tarjetas bancarias más seguro de la historia. Lo que no significa que pueda dormirse en los laureles. Su uso masivo en cada vez más tipos de operaciones de pago hará que se ejerza una enorme presión sobre sus puntos débiles por parte de los grupos del crimen organizado. Ahora es allí donde está el dinero.

10) LA RESPUESTA: CÓMO PROTEGERNOS

El hecho de que el Reino Unido sea una rica fuente de experiencias sobre la seguridad de las tarjetas EMV en particular, y del negocio de tarjetas de pago en general, no debe hacernos olvidar la situación local. Evidentemente, nos interesa lo que sucede en casa. Así pues, ¿cómo nos afecta en la práctica la seguridad imperfecta de los sistemas de tarjetas? ¿En qué se diferencian la legislación y la jurisprudencia española de la inglesa?

La doctrina de los tribunales españoles en relación con las clonaciones, retiradas no autorizadas y uso del PIN tras el robo de tarjetas ha descansado en algo asumido por todos: el sistema no es infalible. Existe un riesgo derivado de la emisión y utilización de tarjetas, sea por robo, duplicación o cualquier otro procedimiento, y eso se considera un hecho, al margen de los medios de seguridad que el emisor de la tarjeta ponga para intentar evitarlo. De ello deriva un perjuicio para las partes involucradas: emisor, titular y dueño del establecimiento. Sin negar la posible culpa del cliente en casos de fraude o negligencia grave, se tiende hacer recaer el daño sobre el emisor de la tarjeta por los siguientes motivos:

—Porque es quien se lleva los mayores beneficios (comisiones, recargos, intereses), así que se considera justo que asuma los riesgos.

—Porque fomentan el uso de productos más arriesgado que otros ya existentes.

—Porque el riesgo no debe recaer en la parte más débil (el titular de la tarjeta)

(Audiencia Provincial de Castellón, 12/2/2000; Audiencia Provincial de Tarragona, 27/12/2004). Como consecuencia, los tribunales aplicaron la doctrina de "quien más se beneficia, más responsabilidad tiene" en diversas ocasiones. En un caso de uso ilegítimo de una tarjeta por haber sido capturada por un cajero automático manipulado, el tribunal afirmó que:

"*es evidente que dicha utilización se corresponde con un fallo del sistema que permite a terceras personas manipular los cajeros automáticos, quebrando su seguridad hasta el extremo de que el mismo emite mensajes incorrectos que inducen a confusión a los usuarios, y si bien es cierto que esta situación se produce por la intervención fraudulenta de terceras personas, las responsabilidades que de estos eventos dimanen frente a los clientes son del banco emisor de la tarjeta, quien necesariamente deberá responder, en su integridad de las consecuencias dañosas producidas*" (Audiencia Provincial de Madrid, 7/12/2000)

Por lo general, el mayor número de quejas correspondía a retiradas de efectivo y compras en tiendas con una tarjeta robada. Los clientes quedaban desconcertados al darse cuenta de que un ladrón pudiese saquearles sin conocer el PIN, y no entendían que el banco se negase a asumir su responsabilidad. El emisor de la tarjeta, por su parte, tendía a pensar que las posibles vulnerabilidades de su sistema no eran causa

probable del fraude, y en aplicación del principio de House (“el cliente es idiota”) estimaban que la hipótesis más probable era que el cliente hubiese apuntado su PIN en la tarjeta. Por otra parte, cuando el cliente no comunicaba al banco “con la debida diligencia” la sustracción de la tarjeta, resultaba más difícil tomar medidas correctivas, de modo que el banco se lo cobraba al cliente. Hasta qué punto el cliente es responsable en esos casos se fue dirimiendo en diversas sentencias.

Lo más importante para la víctima es que los tribunales admitan la posibilidad de que se pudiese obtener el número PIN en forma fraudulenta, ya que en ese caso no sería de aplicación automática la asignación de la responsabilidad al cliente. En un recurso contra un fallo que le obligaba a devolver el dinero sustraído por desconocidos tras el robo de una tarjeta, Diners Club SA alegaba que el PIN no había sido custodiado adecuadamente por el cliente, ya que de otro modo los autores de la sustracción no hubieran podido obtener el dinero. Por su parte, el cliente negó llevar el número junto con la tarjeta, añadiendo con algo de sorna que “*como Letrado se consideraba capaz de memorizar un número de 4 dígitos*”. El razonamiento del banco descansaba en la lógica según la cual solamente el emisor de la tarjeta y el cliente conocían el PIN. El tribunal lo entendió de otro modo:

“*la tarjeta no es tan segura sino que la banda magnética resulta fácil de examinar para deducir el número por personas expertas, auténticos profesionales, como es público y notorio… la conclusión de todo ello es que no puede imputarse el hecho a negligencia del actor*” (Audiencia Provincial de Madrid, Sección 9, 8/4/1999)

Este criterio fue compartido por la Audiencia Provincial de Tarragona en 2004. En un recurso, Cajamadrid afirmaba que el hecho de que existiese una tarjeta duplicada no había quedado suficientemente acreditado, e insistía en que las compras debieron haberse hecho con la tarjeta original del cliente, junto con su PIN. Pero en lugar de argumentarlo con expertos, Cajamadrid presentó el testimonio de un empleado de la sucursal del cliente que… no, prefiero no contárselo con mis palabras. Que hablen los jueces de la Audiencia:

“*aunque fuese cierto que dicho número [PIN] no se deduce de la banda magnética, debería la entidad de crédito haber aportado el correspondiente peritaje técnico de dicho funcionamiento. El Sr. Jose Francisco, que es el X de la oficina de la recurrente [Cajamadrid] y no un técnico informático de seguridad —como lo demuestra al no conocer la técnica del sistema—, reconoce que tarjetas duplicadas han sido utilizadas en autopistas, aunque no en comercios, y que podían ser utilizadas en cajeros, pero siempre con el PIN, que es imposible que sea deducido de la banda [magnética]; lo mismo sucede con la declaración del subdirector que tuvo que preguntar «a la central» para saber si el número PIN es deducible de la banda magnética; puesto que a pesar de las medidas de seguridad, lo cierto es que los*

«hackers» o salteadores pueden entrar en los sistemas informáticos, como también es público y notorio, y conseguir los códigos necesarios; es decir, no se considera probada ni la falta de diligencia de la recurrida ni la infalibilidad del sistema respecto al sistema de número secreto o PIN".

No creo tener que añadir que Cajamadrid perdió el recurso y fue condenada en costas.

Otros tribunales aceptaron como "hecho público y notorio" la falibilidad del sistema, incluso si en el caso particular juzgado no se hubiese demostrado. Bastaba con admitir la posibilidad de acción fraudulenta para poder, cuando menos, levantar el sambenito de "el cliente tenía el PIN, así que él tiene que ser el culpable," y diversos tribunales así lo manifestaron así a lo largo de la década del dos mil. He aquí algunos ejemplos:

"*Con respecto a las extracciones de dinero, es cierto que se precisa para ello de la utilización del código PIN, pero lo que no consta es que tal número no pueda ser conocido a través de la manipulación o lectura de la banda magnética por personas que conozcan el funcionamiento y desciframiento de tales elementos, y así era al demandado al que correspondía probar que sólo el titular de la tarjeta puede acceder a su número secreto*". (Audiencia Provincial de Madrid, 6/10/2004)

"*la actualidad nos ilustra de que este tipo de operaciones [clonación de tarjetas] se realizan mediante la instalación en cajeros de dispositivos de lectura de datos tanto del lector de la banda magnética como del teclado (para el PIN) que son enviados al receptor a disposición de quien emplaza este tipo de dispositivos que inmediatamente crean una tarjeta con los datos suministrados de la banda magnética utilizada regularmente en el cajero en el que se instaló el dispositivo, al tiempo que conocen el PIN de la misma, encontrándose en disposición de realizar extracciones fraudulentas de la cuenta de cargo de la misma, sin que el titular de la tarjeta tenga conocimiento de ese 'clonado' de su tarjeta*". (AP Córdoba, 24/7/2006)

"*No puede presumirse la existencia de negligencia grave en la custodia de la clave secreta o de la tarjeta en todos aquellos casos en que las operaciones no reconocidas por el titular hayan sido validadas electrónicamente con el número de identificación personal (PIN), pues consideramos que la obligación general de custodia del titular de la tarjeta que proclama el contrato, no puede conllevar la presunción anteriormente enunciada, ya que la realidad social actual nos muestra que es cada vez más frecuente el conocimiento del PIN por métodos diferentes y más sofisticados que su simple sustracción o pérdida*". (AP Torrevieja, 4/12/2006)

"*La Sala discrepa con la alegación del recurrente [el banco], ya que parece éste entender que únicamente cuando se trata de sofisticados artificios encaminados a clonar o falsificar tarjetas se puede entender que el titular de la tarjeta carece de negligencia en la utilización fraudulenta de la misma… cuando realmente no es la*

sofisticación del método utilizado, sino la efectividad del mismo lo que determina tal circunstancia" (AP Madrid, 3/10/2007)

Una de las manifestaciones más contundentes de lo que esto significa para la asignación de responsabilidad fue expresada en 2010 en los siguientes términos:

"*… la experiencia diaria confirma cómo son utilizadas fraudulentamente tarjetas que han sido sustraídas o extraviadas, sin que pueda garantizarse una seguridad absoluta en la utilización de tales instrumentos, doctrina que trae como consecuencia que corresponda a la entidad financiera la carga de acreditar que el sistema utilizado es completamente seguro e infalible y que el acceso a sus servicios sólo puede verificarse con el marcado de un número PIN, número que es en sí mismo indescifrable, o dicho de otro modo, que la única forma que tiene el tercero de acceder a tales servicios es visionando previamente el PIN en cualquier forma, y, tal circunstancia no ha sido probada …por tanto, a la entidad bancaria le corresponde asumir los riesgos que conlleva la tarjeta en sí porque ella se lleva los beneficios: comisiones de uso, mantenimiento, recargos, intereses…*" (AP Islas Baleares, 11/2/2010).

En una ocasión, los jueces se dejaron llevar de un espíritu de protección más allá de lo habitual, al entender que "*la nueva utilización ordinaria de la tarjeta entraña un riesgo evidente, que es inherente a tal medio de pago y en definitiva es el banco quien comercializa un producto cuya fiabilidad y seguridad se pone en entredicho ya que no se aplican sistemas biométricos de identificación (implantación de un sistema de identificación a través de la huella dactilar, a través del iris ocular, reconocimiento de voz…)*" (AP Barcelona, 24/10/2006).

En general, puede concluirse que los tribunales son conscientes de la existencia de fallos en los sistemas de tarjetas, y en consecuencia no procede culpar al cliente de forma automática. Por supuesto, siempre hay que contemplar el caso de fraude o mera estupidez. En cierta ocasión (no daré detalles), el titular denunció reintegros fraudulentos en su tarjeta de crédito. Sin embargo, dejó pasar tres meses hasta efectuar la denuncia, y además estuvo pagando las cuotas durante casi dos años. La Audiencia Provincial entendió que no había habido diligencia a la hora de denunciar el presunto robo.

En cuanto al siguiente caso, juzgue el lector por sí mismo. Caixanova fue absuelta de responsabilidad tras una sustracción de una tarjeta bancaria y un DNI seguida de diversos cargos en cuenta por un total de 785 euros. El cliente recurrió ante la Audiencia Provincial de Pontevedra, la cual se apoyó en el criterio de que "*se entiende que concurre negligencia grave cuando el dato del PIN está de tal modo unido a la tarjeta que el robo o extravío de ésta conlleva información del PIN*". En su resolución, el tribunal afirmó que "*el número secreto no figura en la banda magnética de las tarjetas, por lo que no es posible su averiguación mediante medios*

electrónicos" Pero en este caso, sorprendentemente, sí que se podía averiguar. Resulta que el PIN de la tarjeta del cliente ¡era su fecha de nacimiento! Ni que decir tiene que la Audiencia desestimó el recurso.

Cuando el intervalo que pasa entre el robo de la tarjeta y el saqueo de la cuenta es lo bastante grande, la responsabilidad tiende a recaer sobre el cliente. Esto se debe a la idea de que dejar pasar semanas o incluso meses entre el robo y la denuncia (o la comunicación al banco) es síntoma claro de dejación y poca diligencia, y ese razonamiento tiene su lógica. Lo habitual es que, si el cliente comunica con diligencia el robo al banco y a las autoridades, los tribunales suelen fallar en su favor, incluso si ha pasado más tiempo del reglamentado. En tales caso, el hecho de haber actuado de forma correcta y diligente tiene preferencia sobre el contenido exacto del contrato. Los bancos que insertaban cláusulas del tipo "*el titular será responsable sin limitación alguna del uso de la tarjeta antes de la notificación de la pérdida o sustracción*" vieron una y otra vez cómo los tribunales anulaban en la práctica estas cláusulas, hasta que fueron declaradas abusivas por el Tribunal Supremo en diciembre de 2009.

Sin embargo, cuando el intervalo que transcurre entre el robo de la tarjeta y el saqueo de la cuenta se reduce a una hora, o incluso menos, los bancos tiene una oportunidad de acusar al cliente de falta de diligencia. Alegan entonces que no es posible averiguar el PIN de una tarjeta en tan poco tiempo, por lo que la única conclusión lógica es que el cliente ha apuntado el número PIN en la tarjeta, lo que les eximiría de responsabilidad. De hecho una banda bien organizada puede clonar y obtener el PIN en muy poco tiempo. Como hemos visto en páginas anteriores, los ataques que se conocen son rápidos y están al alcance de grupos criminales. No se trata de un esfuerzo de días en el sótano de una casa abandonada.

Una retirada de fondos apenas minutos después del cometido el robo de la tarjeta sonaba sospechoso para algunos jueces. Las Audiencias Provinciales de Salamanca (2004), Madrid (2008) y Asturias (2011) dieron la razón a los bancos, puesto que no podían asimilar que el período transcurrido (8-15 minutos) era suficiente para que un ladrón hiciese su trabajo:

"*tal número estaba con la tarjeta, sin que en buena lógica (lo 'normal'), en tan escaso tiempo, pueda alcanzarse solución distinta, no existiendo medios técnicos distintos para acceder a tal número, al exigir al acceso a tal clave en tan corto lapso temporal unos conocimientos, medios técnicos y dedicación que no podían razonablemente concurrir en el presente caso*" (AP Salamanca, 20/4/2004)

"*… en contra de lo apuntado por la apelante, el número secreto no figura en la banda magnética de las tarjetas, por lo que no es posible su averiguación mediante medios electrónicos… a ello se suma la gran proximidad en el tiempo de esas extracciones con el momento de la sustracción, lo que no es verosímil sin haber*

existido un conocimiento previo de las claves o número secreto para su uso" (AP Madrid, 20/11/2008).

En el caso de Asturias, al menos había un elemento de duda razonable a favor del banco. A Sara, la víctima, le habían robado varias tarjetas, pero los ladrones solamente lograron retirar dinero con dos de ellas. Las otras no pudieron ser usadas por falta de PIN correcto. Si los ladrones hubieran tenido a mano elementos para clonar tarjetas y obtener sus PIN, es lógico suponer que las habrían usado todas. La Audiencia también valoró la afirmación de Sara de que "había dejado el bolso apoyado en el respaldo de su asiento" en un local público como indicio de negligencia.

En cualquier caso, aun suponiendo que una retirada de fondos fraudulenta en pocos minutos es poco probable, sí es una posibilidad que ha de tenerse en cuenta. Eso es lo que admitió la Audiencia Provincial de Madrid (otra sala distinta) en 2007:

"*que los ladrones tardaran muy poco tiempo en hacer uso de la tarjeta desde la sustracción, no es motivo para suponer que necesariamente el número PIN debía encontrarse en la cartera…*".

Por pura coincidencia, justo tras la aprobación de la Ley 16/2009 de Servicios de Pago, la Sala de lo Civil del Tribunal Supremo falló en contra de un cierto número de cláusulas bancarias abusivas[75]. Lo más interesante es que en este caso se aplicó la nueva Ley 16/2009, a pesar de que ésta solamente llevaba en vigor tres días. Algunas de las cláusulas en disputa eximían a los bancos de responsabilidad en caso de pérdida o sustracción, haciendo responsable al cliente de forma ilimitada salvo que éste comunicase el problema al banco de forma inmediata. Más grave todavía, invertían la carga de la prueba: en caso de que se utilizase el PIN, se consideraría "grave negligencia" salvo caso de fuerza mayor.

El Tribunal Supremo, tras declarar desproporcionadas las cláusulas que se limitan a la exoneración de responsabilidad por parte del banco antes de la notificación de sustracción o extravío, entró en el tema de la responsabilidad por el uso del PIN, y determinó que no es proporcionado limitar la responsabilidad bancaria a los casos en que el cliente reveló el número ante coacción o fuerza mayor. El siguiente párrafo es especialmente significativo en la parte que aquí nos interesa (el subrayado es mío):

"*Cierto que con la utilización del chip electrónico en lugar de la tarjeta con banda magnética, y el necesario marcaje o tecleo del número secreto por el titular, cabrá reducir (en las operaciones con presencia física; otro tema lo constituyen las realizadas a distancia, como sucede con internet) las utilizaciones indebidas, pero respecto del caso que se examina no cabe desconocer la posibilidad de captaciones subrepticias, con independencia de otras manipulaciones varias a causa de las deficiencias del sistema de tarjetas, que no permiten sentar una cláusula que exonere de responsabilidad…*".

Como pueden ver, los jueces del Supremo admiten que el sistema no es totalmente seguro, y que por tanto no se puede aplicar la doctrina británica de “mi sistema es seguro, por tanto yo no soy responsable”. En consecuencia, el TS estimó que (ahora el subrayado es de la propia sentencia):

2. La exclusión de responsabilidad en todo caso para la entidad bancaria por las utilizaciones de tarjeta o de libreta —consistentes en extracciones en efectivo u otras operaciones con cargo a la cuenta bancaria—, con anterioridad a la comunicación de la sustracción o extravío (o evento similar) es desproporcionada, y abusiva.

3. Es igualmente abusivo excluir de responsabilidad a la entidad bancaria en todo caso de uso del número de identificación personal limitando aquélla a los supuestos de fuerza mayor o coacción.

Si usted siente en estos momentos la necesidad de comprobar el contrato de sus tarjetas bancarias, hágalo. Yo lo hice. Siempre es mejor prevenir que curar.

Pero basta de asustar. Imagino que a estas alturas ya habrá recibido el mensaje: hay peligro. Eso no significa que deba usted tirar a la basura sus tarjetas de crédito y débito. Después de todo, también le pueden robar sus billetes, y no por eso vamos a dejar de usar el dinero en efectivo. Se trata, sencillamente, de tomar algunas precauciones básicas que le evitarán muchos problemas. Puede que algunas ya las haya leído en otra parte, y en general son de sentido común.

Comencemos por el elemento más vital: el PIN. Para conservarlo, nada mejor que confiarlo a la memoria. Allí debe almacenarse y allí debe estar la única copia. Nunca lo apunte, en ninguna forma o lugar. Si tiene un mal día y lo olvida, siempre habrá tiempo de volver al banco y pedir un PIN nuevo. Nunca lo comparta con nadie: ni con su mejor amigo, ni con su director de banco, con nadie. No es algo que nadie más que usted necesite saber. Ni su banco debe saberlo, y si ellos se lo piden, niéguese.

La tarjeta debe estar siempre localizable. Esto resulta especialmente importante si se encuentra de viaje. Si no la necesita, déjesela en casa. Ahora que las tarjetas tienen chip, no se suele verificar la firma manuscrita, pero de todos modos no deje de firmar la tarjeta al reverso. Una idea interesante es acompañar la firma con un “pedir DNI”. Eso le proporcionará una capa adicional de seguridad, porque el vendedor tendrá que verificar la identidad de usted. Si no lo hace, tanto peor para él.

A la hora de pagar, tenga la precaución de vigilar adónde se llevan la tarjeta. El camarero o el dueño de la tienda podrá pasarla por un lector de bandas magnéticas con facilidad y sin que usted se de cuenta, pero de todos modos esté ojo avizor. En lo posible, utilice el sistema Chip+PIN, y asegúrese de que nadie le vea teclear los números. No se preocupe si alguien se ofende por ello, usted tiene derecho a proteger sus secretos.

Cuando el uso del chip no es posible, el sistema recurre al método de lectura por banda magnética, como ya hemos visto. Se trata de una práctica legítima pero, como

hemos visto, más arriesgada, así que le recomiendo evitar el uso de la banda magnética. Si le dan una lectora portátil donde la tarjeta entre completamente en la ranura, ¡alerta! Es posible que los malos la hayan sustituido por una lectora trucada, sin que los empleados del establecimiento se hayan dado cuenta. Si eso sucede, avíselos y por ningún motivo introduzca su PIN en el teclado. Hay personas que incluso arrancan o inutilizan físicamente la banda magnética. No es algo que le recomiende por diversos motivos: puede que necesite usted usarla, y a lo mejor su banco no le pone buena cara.

Usar la tarjeta en el cajero para luego pagar en efectivo en las tiendas es una opción más segura que pagar con tarjeta. Los cajeros están mejor vigilados, y sabemos qué banco está detrás para dar la cara en caso de problemas. Aun así, tenga cuidado y examine el cajero. ¿Tiene marcas o señales de haber sido forzado? ¿Hay algún objeto insertado en la ranura de la tarjeta? ¿El cajero automático retiene la tarjeta durante un intervalo de tiempo inusualmente largo? En caso de duda, pruebe con otro. Guarde siempre los recibos y comprobantes, incluso en caso de transacción no completada (y especialmente en ese caso). Por supuesto, asegúrese de que nadie esté fisgoneando por encima del hombro. Tape el teclado cuando introduzca su PIN. No se fíe de la ayuda de desconocidos, por muy buena cara que pongan. Y no se deje el recibo en el cajero: aunque todo vaya bien, un ladrón puede utilizar esa información para fines fraudulentos… por no hablar de que así sabe cuanto dinero lleva usted ahora en la cartera y si vale la pena atracarle.

Resulta buena política el comprobar el extracto y últimos movimientos de su cuenta de forma periódica, en busca de transacciones extrañas que no recuerde usted haber autorizado. En caso de duda, hable con el banco emisor de su tarjeta, y si es necesario anule la tarjeta. La molestia de pedir otra y tener que esperar a que llegue es un pequeño precio a pagar a cambio de evitar fraudes con su tarjeta.

Una precaución que le recomiendo especialmente: guarde el número de teléfono del servicio de atención al cliente de su tarjeta en lugar bien visible, por ejemplo en su lista de contactos. Como ya ha leído en este libro, de la sustracción a la clonación y uso a veces solamente pasan unos minutos, así que en caso de robo de tarjeta llame y anúlela inmediatamente. Antes de que el amigo de lo ajeno haya doblado la esquina. No pierda ni un momento. Identifíquese, indique que su tarjeta ha sido robada y pida su anulación con carácter inmediato. Luego habrá tiempo para la denuncia en comisaría, donde podrá usted sacar su móvil y mostrar el registro de llamadas. Incluya la fecha y hora de la llamada en la propia denuncia, será una prueba de su diligencia en caso necesario. ¿Y si usted tuvo un mal día y también le han robado el móvil? En ese caso, pida ayuda a algún transeúnte, pídale prestado su móvil, busque el número de atención al cliente de su tarjeta y proceda a su anulación.

Finalmente, cuando la tarjeta deje de ser útil, por ejemplo en caso de que haya

caducado, o si la anuló usted y al final apareció debajo del sofá, no se limite a tirarla a la basura. Recuerde que la información que los ladrones buscan sigue allí, en la banda magnética y el chip. Antes de tirarla, inutilícela. Basta con unas tijeras. Asegúrese de cortar la región de la banda magnética, e incluso el chip si puede. La mejor recomendación que le puedo hacer en este punto es que adquiera usted una máquina destructora de documentos, una inversión que se paga sola. No solamente le servirá para destruir su tarjeta vieja, sino que también podrá usarla con todos esos documentos que ya no sirven y que llevan datos personales: extractos bancarios, facturas, informes del médico de los hijos, etc. Un defraudador puede obrar milagros con esa información que cae en el cubo de basura.

Y un último detalle. Quizá crea que soy un poco paranoico con la seguridad. Está usted en su derecho, y puede no hacerme caso, que no soy su padre. Pero no crea que es usted lo bastante poco importante para que una banda organizada se plantee a robarle a usted. No necesita usted ser un empresario millonario o una estrella del cine para que quieran robarle su información o sus tarjetas de crédito. Los ladrones no se interesan en usted personalmente, para ellos es tan sólo negocios. Sencillamente, buscan víctimas vulnerables y aprovecharán la ocasión si usted se la facilita. Niéguesela.

CALCULANDO CON PRIMOS

El 30 de junio de 2009, un desconocido publicó en Internet un conjunto de números primos grandes. Ese sencillo gesto conllevó una gran labor previa de computación y provocó la ira de un fabricante de calculadoras. El motivo no es meramente académico, ya que esos números primos se utilizan para operaciones criptográficas. Una buena parte de todo el comercio electrónico mundial, incluyendo la banca online, sustenta su seguridad en números primos.

Entender la hazaña del desconocido calculador de primos nos llevará un buen rato, y me temo que necesitaremos echar mano de las matemáticas. Imagino su cara de desilusión en este momento. Usted, amable lector, llega a estas líneas esperando leer una historia amena, y en su lugar el autor le amenaza con una clase de matemática avanzada. Bien, le propongo un trato. Voy a saltarme la parte complicada y mantendré lo justo para que usted sepa de qué estamos hablando; por su parte, usted no se enfadará si no soy riguroso en mis explicaciones. Incluso le daré la posibilidad de saltarse la parte del cálculo.

¿Hay trato? Estupendo. Vamos allá.

1) CLAVES PÚBLICAS, CLAVES PRIVADAS

Que se pueda cifrar información con números primos es en sí llamativo, pero el concepto subyacente roza lo asombroso. Durante siglos, los sistemas de cifrado han tenido un grave talón de Aquiles: utilizamos la misma clave para cifrar y para descifrar. Si le damos a una persona una clave para que pueda enviarnos un mensaje cifrado, le estamos dando también la posibilidad de descifrar todos los mensajes cifrados con esa clave.

La situación mejoraría si las claves de cifrado y de descifrado fuesen distintas. Sería algo así como un buzón de correos: cualquiera puede introducir una carta, pero solamente el poseedor de la llave puede abrir el buzón para leer las cartas. La clave para cifrar sería accesible para cualquiera que desease cifrar un mensaje, en tanto que la clave de descifrado solamente estaría accesible al destinatario.

Usar claves distintas para cifrar y para descifrar se consideró algo imposible durante muchos años, una especie de santo grial de la criptografía. Pero en los años setenta, lo imposible se hizo realidad. Investigadores de Estados Unidos y el Reino Unido descubrieron los principios de lo que hoy conocemos con el nombre de criptografía de clave pública, a veces llamada también criptografía asimétrica. En lo que sigue, me referiré a ella mediante las siglas en inglés PKC (*Public Key Criptography*).

La PKC se basa en la existencia de dos claves. Una de ellas (la clave pública) puede diseminarse a los cuatro vientos para que cualquiera pueda usarla; la otra (la clave privada) permanece firmemente bajo el control del dueño. Cualquiera puede cifrar mensajes con la clave pública, pero solamente el poseedor de la clave privada asociada podrá descifrarlos. La PKC permite resolver otros problemas como el intercambio de claves o la firma digital.

La PKC fue desarrollada por los investigadores norteamericanos Whitfield Diffie, Martin Hellman y Ralph Merkle a finales de los años 70. No fueron los primeros, ya que pocos años antes sus homólogos británicos James Ellis, Clifford Cocks y Malcolm Williamson habían descubierto los principios de la PKC. Por desgracia para ellos, trabajaban para la agencia criptoanalítica británica GCHQ, y su trabajo se mantuvo en secreto durante muchos años (solamente a finales de los años 90 se les permitió hablar de su papel en la PKC). Fueron los norteamericanos quienes revolucionaron el campo de la criptografía civil y se llevaron todos los honores.

La PKC basa su fortaleza en problemas matemáticos difíciles. Las claves pública y privada no son independientes, y en teoría podría obtenerse una de ellas conociendo la otra; la seguridad del sistema se basa en escoger un problema tan difícil que la posibilidad teórica de resolverlo derive en una imposibilidad práctica. No es fácil encontrar un problema matemático adecuado. Algunos no son lo bastante difíciles,

otros lo son demasiado, o bien las claves de cifrado son demasiado largas.

Uno de los primeros ejemplos de PKC se basó en lo que se conoce como "problema de la mochila" (*knapsack problem*). Podemos enunciarlo de la siguiente manera: dado un conjunto de elementos, cada uno de ellos de un peso determinado, ¿es posible llenar una mochila con algunos de ellos, de forma que la mochila pese una determinada cantidad? O dicho matemáticamente: dados un conjunto de valores *V1, V2… Vn* y una suma *S*, determínense los valores *Ai* (cero o uno) que cumplan la condición:

*S=A1*V1 +A2*V2… + An*Vn*

En la película *Jungla de Cristal 3: la venganza*, el detective McClane y su compañero deben resolver una variante del problema de la mochila. En una fuente pública hay una bomba con una báscula incorporada. Las instrucciones del malo para desactivarla son sencillas:

"*Debería haber dos garrafas en la fuente, una de cinco y otra de tres galones. Llene una de ellas con cuatro galones justos de agua y póngala sobre la báscula; el contador se parará… si siguen vivo dentro de cinco minutos volveremos a hablar*".

Si quiere usted probar suerte, adelante, es fácil. Cuando tenemos pocos elementos, el problema es fácilmente tratable; pero conforme el número de elementos *n* aumenta, el problemas se hace cada vez más difícil. Eso forma la base de un sistema de cifrado asimétrico diseñado por Ralph Merkle y Martin Hellman en 1978[1].

Puesto que los valores *Ai* son ceros o unos, la secuencia (*A1, A2, A3… An*) se escoge de forma que sean una representación binaria del mensaje. Los procesos de cifrado y descifrado están basados en dos series distintas de valores (*V1, V2, V3… Vn*). La diferencia entre ambas estriba en que la clave privada está formada por lo que se llama una secuencia supercreciente, en la que un valor es mayor que la suma de los valores anteriores, o sea: *Vk > V1+V2+V3… +V(k-1)*.

El lector tiene en sus bolsillos un ejemplo de secuencia supercreciente. Fíjese y verá cómo cualquier billete o moneda tiene más valor que todas las de menor cuantía juntas. Tener monedas en sucesión supercreciente facilita las transacciones. Si tiene usted que pagar 62 céntimos, no tiene más que tomar una moneda de 50, una de 10 y una de 2. Por supuesto, existen otras posibilidades (50+5+5+1+1 céntimos), pero no son muchas. Con un sistema de monedas no supercreciente, habría muchas más posibilidades, y tendríamos que llevar muchas más monedas en el bolsillo.

El hecho de que la clave privada esté formada por una sucesión supercreciente, y la clave pública no, garantiza que el proceso de descifrado será sencillo si se conoce la clave (espero que me crea en esto, porque no quiero aburrirle con detalles matemáticos); en caso contrario, es lo que los matemáticos llaman un problema NP-

completo. Para entendernos, es el tipo de problemas que no puede resolverse con un algoritmo matemático que, al final de un cierto conjunto finito de pasos binarios, diga "sí, hay solución" o "no, no hay solución".

Al principio se creyó que el problema de la mochila podría ser usado como "problema duro" para basar en él un sistema de cifrado. Sin embargo, Adi Shamir demostró en 1984 que el problema de la mochila puede "casi resolverse" en un tiempo relativamente corto. Ese "casi resolverse" significa que no hay seguridad de resolverlo, pero se puede considerar resuelto con una probabilidad muy alta de éxito[2].

Durante los años setenta y ochenta se pusieron a prueba muchos problemas matemáticos difíciles. Uno de ellos finalmente dio origen a un sistema de clave pública conocido con el nombre de RSA, por las iniciales de sus creadores: Rivest, Shamir, Adleman. Fue uno de los más importantes hitos que permitieron crear una Internet segura, con páginas web protegidas mediante criptografía. Su éxito fue tal que la empresa que comercializa RSA es en la actualidad una de las más importantes en el campo de la seguridad en Internet[3].

A partir de aquí, nuestro camino se llena de matemáticas. El siguiente apartado será tan sencillo como puedo explicarlo, pero aun así puede que usted no quiera volver a clase de matemáticas. Llega, por tanto, el momento en el que usted escoge. Bienvenidos a las matemáticas interactivas. Por favor, escoja una de las dos opciones:

—Continuar adelante (Apartado 2)

—Saltar la explicación matemática (Apartado 3)

2) DESCRIPCIÓN MATEMÁTICA DE RSA (SÓLO PARA VALIENTES)

¡Enhorabuena por su coraje, amable lector! Comencemos sin más preámbulos. Ya sabe, si se aburre no tiene más que saltarse este apartado. Con total confianza. A fin de cuentas, éste es su libro.

El algoritmo RSA se basa en la dificultad matemática de factorizar números primos grandes: dado el número *n = p*q*, conocer los valores de *p* y *q*. En lo que sigue, supondremos que todos los números que aparecen son enteros positivos.

El método que nos enseñaron de pequeños consiste en dividir *n* por todos los números inferiores a él. Podemos refinar este método, y dividir sólo por los números primos menores que la raíz cuadrada de *n*. Por ejemplo, ¿es 143 primo? Es impar, así que no es divisible por dos. En cuanto a los demás posibles factores, probémoslos uno por uno:

```
143/3  = 47.666  no
143/5  = 28.6    no
143/7  = 20.428  no
143/11 = 13      sí
```

Como 13 es primo, tenemos la descomposición única 143=11*13.

Este sencillo procedimiento de división quizá le suene, es el típico test de primalidad que aprendemos en la escuela. Es fácil de usar para números pequeños. Pero cuando *n* es muy grande, resulta inviable. Hay otros métodos de factorización más eficientes y rápidos, pero del mismo modo se convierten en poco prácticos si *n* es lo bastante grande.

Afortunadamente, existen tests de primalidad más sencillos. Uno de ellos, denominado *Test de Primalidad de Fermat* (TPF) está basado en el Pequeño Teorema de Fermat, que dice:

"*Si p es un número primo, y a es un número cualquiera entre 1 y p, entonces se cumple que la división a^(p-1) / p tiene resto igual a uno*"

No parece tan obvio, pero los matemáticos, que saben de eso más que nosotros, nos afirman que ese teorema se cumple. Así, tenemos una prueba que no nos exige probar todos los posibles factores de un número grande. Estupendo, salvo por una pega. Dos pegas. No, mejor tres pegas.

La primera pega es que el TPF es un test probabilístico. Quiero decir con esto que no tenemos certeza de que un número sea primo, sino tan sólo que puede que lo sea. El problema es que, en el campo de la lógica, a => b no significa necesariamente b => a. Por ejemplo, supongamos que todos los coches Ford sean azules. Si usted ve un Ford, entonces es azul. Pero si usted ve un coche azul, no tiene por qué ser un Ford.

Eso es lo que le pasa aquí. Si *p* es primo, entonces se cumple el TPF; pero si se cumple el TPF ¿es *p* primo? La respuesta es: a veces sí, a veces no. No hay certeza. Pero podemos utilizar el TPF como prueba probabilística. Escojamos un número *a* y hagamos el TPF. Si *p* no pasa el test, entonces no es primo. Si, por el contrario, lo pasa, entonces volveremos a aplicarle el TPF con otro valor de *a*. Y luego otro. Y luego otro. Y otro. Hagamos el test un número elevado de veces. Si *p* los pasa todos, entonces la probabilidad de que sea primo será muy elevada.

Desafortunadamente, existen números que pueden pasar el TPF para todos los valores posibles de *a*, y a pesar de ello, no ser primos; reciben el nombre de números de Carmichael. Si nos ha tocado uno, estamos de mala suerte, porque el TPF no nos permitirá detectar que realmente es un número compuesto. Afortunadamente, los números de Carmichael son muy raros: para números de cien dígitos, hay aproximadamente un número de Carmichael por cada trillón de números primos.

Le dije que había tres pegas. He dejado la más gorda para el final. Según el TPF, hay que comprobar si resto de dividir *a*^(*p-1*) por *p* es igual a la unidad. Esa es una operación endiabladamente lenta. Por poner un ejemplo sencillo, sea *a*=2 y *p*=997. Son números sencillos y pequeños, y sin embargo 2^996 ¡es un número de trescientas dígitos de longitud! ¿Esto es un test de primalidad práctico? ¿En qué estaba usted pensando, señor Quirantes?

Afortunadamente, hay un método de calcular el resto de *a*^(*p-1*) / *p* sin necesidad de efectuar la división, ni por supuesto la potenciación. El inconveniente es que requiere un pequeño cambio de chip mental, pero es algo que vamos a utilizar después para describir el algoritmo RSA, así que vamos a intentarlo. Se trata del concepto de **aritmética modular**.

Habitualmente, cuando hacemos la división *c=a/b* lo que suele interesarnos es el cociente. Si *a*, *b* son dos números reales, el cociente es *c*, y el resto es cero. Si, como en el caso que nos ocupa, se trata de números enteros, entonces el cociente no tiene por qué ser exacto. Bien, pues en la aritmética modular no nos interesa el cociente, sino el resto.

Para esta nueva operación, utilizaremos la notación “mod”. Cuando escribimos *a mod n* (“*a* módulo *n*”) nos estamos refiriendo a “el resto que obtenemos al dividir *a* por *n*” Cuando queramos indicar que al dividir *a* entre *n* obtenemos *b*, lo escribiremos así:

$$a \bmod n = b$$

Quizá no lo sepa, pero usted utiliza de vez en cuando la aritmética modular sin darse cuenta. Eche un vistazo a un reloj digital. Son las once de la noche, es decir, las 23:00. Dentro de cuatro horas, ¿qué hora será? 23+4 son 27, pero nadie dice “son las veintisiete”. Su reloj no marca 27:00 sino 3:00, que es el resto de dividir 27 entre 24;

podemos decir que nos muestra la hora “módulo 24”. De forma similar, el famoso reloj Big Ben londinense marca las horas “módulo 12” y nunca —salvo por error— dará quince campanadas.

La aritmética modular es algo más engorrosa que la tradicional, pero es la pieza fundamental para poder operar con números muy grandes. Volvamos al ejemplo del TPF. Se trata de averiguar si el resto de la división *a*^(*p-1*) / *p* es igual a uno. En aritmética modular, lo escribimos como *a*^(*p-1*) mod p = 1

Para calcular, echaremos mano de algunas propiedades de la aritmética modular. Para ello, tendremos que olvidarnos de las reglas que nos enseñaron en el colegio. Por ejemplo, la propiedad distributiva de la multiplicación decía que (*a+b*)**c* = *a***c* + *b***c*. Aquí la cosa es algo distinta. Vamos a utilizar esta propiedad de la aritmética modular:

(*a***b*) mod *n* = [(*a* mod *n*) * (*b* mod *n*)] mod *n*

Voy a decirlo con palabras. Para averiguar cuál es el resto de dividir (*a***b*)/*n*, haremos lo siguiente:

1) Hallar el resto de la división *a/n* (eso será *a* mod *n*)

2) Hallar el resto de la división *b/n* (eso será *b* mod *n*)

3) Multiplicar ambas cantidades.

4) Hallar el resto de dividir el resultado 3) por n.

Y listo. Fíjense que no hemos tenido que multiplicar *a***b*, ni dividir el resultado por *n*. Apliquémoslo al TPF, donde tenemos que hallar *a*^(*p-1*) mod *p*. No hay más que generalizar la propiedad anterior:

a^(*p-1*) mod *p* = [(*a* mod *p*)^(*p-1*)] mod *p*

y obtener fácilmente el resto que buscamos, incluso si *p* es un número de cien dígitos.

Creo que de momento ya hemos introducido suficientes matemáticas para comenzar con el algoritmo RSA. Partiremos de dos números primos grandes (*p*, *q*) y su producto *n=p***q*. Calculemos también el número *F=(p-1)(q-1)*. El número *n* es público, pero *p* y *q* no lo son.

Vamos ahora a construir la clave privada y la clave pública. La **clave pública** *e* será un número tal que *F* y *e* sean primos relativos. Con “primos relativos” queremos decir que no tengan divisores comunes, es decir, que su máximo común divisor sea uno. No tienen por qué ser primos ellos mismos. Por ejemplo, los números 15 y 14, a pesar de ser números compuestos, son primos relativos.

Ahora vamos con la **clave privada** *d*, un número que cumpla que *ed* mod *F* = 1; se dice en este caso que *d* es el inverso (multiplicativo) de *e* módulo *F*. El problema

es que, en aritmética modular, el inverso es muy difícil de calcular. Peor aún, a veces existe inverso y a veces no. Se puede demostrar que la condición necesaria para que exista inverso es que los números *e* y *F* sean primos relativos. Por eso le hemos exigido esa condición específica al número *e*.

Ya estamos preparados para cifrar el mensaje *M*, que vamos a representar mediante un número entero. Es preciso que *M* sea menor que *n*; en caso de lo que no lo sea, lo dividiremos en bloques m_1, m_2, etc. Por comodidad, vamos a suponer que nos encontramos en el caso *M<n*. Para obtener el mensaje cifrado *C*, realizaremos la siguiente operación:

$$C = (M^e) \bmod n$$

Fíjense que tenemos un número *M* enorme, elevado a una potencia *e*, pero no importa porque, como ya hemos visto, la aritmética modular me permite operar con facilidad. El proceso opuesto, el descifrado, lo realizamos calculando el siguiente número:

$$(C^d) \bmod n$$

Veamos cómo el proceso nos devuelve el mensaje *M*. O mejor, no lo veamos. O mejor aún, decida usted. Vuelvo a darle el control, querido lector. Estrictamente hablando, puede usted saltarse la explicación que sigue y pasar al apartado siguiente. No se perderá gran cosa. Sin embargo, me considero en la obligación de proporcionarle esa explicación. Es usted quien escoge, que para eso ha pagado el libro (bueno, *confío* en que habrá usted pagado el libro). Así, pues, si desea seguir la explicación, siga leyendo estas líneas. Si por el contrario, desea poner fin a la clase, salte al apartado siguiente pulsando este enlace.

Si está leyendo usted esta frase, es que ha decidido quedarse y apuntarse a la explicación completa. Así sea. Lo haré lo más fácil que pueda, aunque no le garantizo nada. Vamos, en primer lugar, a desarrollar la potencia *C^d*:

$$C^d = (M^e)^d = M^{(ed)}$$

Como *ed* mod *F=1*, eso significa que existe un número *k* tal que *ed = kF+1*, así que:

$$\begin{aligned} C^d &= M^{(ed)} \\ &= M^{(kF+1)} \\ &= M*M^{(kF)} \\ &= M*(M^F)^k \end{aligned}$$

El resto de la demostración dependerá de si *M* y *p* tienen factores comunes o no. Y con "factores comunes" me refiero a los no triviales, de forma que el número *1* no vale. Vamos a cubrir las dos posibilidades

1) M y p no tienen divisores comunes. Puesto que *p* es primo, podemos aplicar el test de primalidad de Fermat:

$$M^{(p-1)} \bmod p = 1$$

Calculemos el cuadrado de dicha cantidad módulo *p:*

$$[M^{(p-1)}]^2 \bmod = [M^{(p-1)} \bmod p)*(M^{(p-1)} \bmod p)] \bmod p$$
$$= 1 \bmod p$$

Siguiendo el mismo razonamiento, cualquier potencia del tipo $M^{a(p-1)} \bmod p$ es igual a la unidad. En particular, vamos a elevar $M^{(p-1)}$ a la potencia $k(q-1)$:

$$M^{[k(p-1)(q-1)]} \bmod p = 1$$

Ahora, multipliquemos $M^{[k(p-1)(q-1)]}$ *por M*. Tenemos:

$$M^{[k(p-1)(q-1)+1]} \bmod p$$
$$= M*M^{[k(p-1)(q-1)]} \bmod p$$
$$=[(M \bmod p)* M^{[k(p-1)(q-1)]} \bmod p$$
$$= [(M \bmod p)] \bmod p$$
$$= M$$

En el ultimo paso, he hecho un poco de trampa. "M mod p" es el resto de dividir M entre p, y lo habitual es suponer que *p* está comprendido entre 0 y *M-1*. Si dividimos 55 entre 9, decimos que el resto es uno porque 55=9*6+1. Pero también podemos decir que 55=9*5 +10. Según eso, "55 mod 9" es igual a uno, pero también es igual a diez. Incluso podríamos decir que 55 mod 9 = 55, ya que es cierto que 55=9*0+55. Habitualmente es una tontería escribirlo así, pero lo importante es que la aritmética modular me permite ponerlo en la forma que me resulte más favorable. De modo que es perfectamente válido afirmar que $M \bmod p = M$. Y eso es lo que acabamos de hacer en el desarrollo anterior.

Se estará preguntando el por qué de todo este baile modular. El motivo es que, como dijimos antes, $ed = kF+1$, y F era igual a $(p-1)(q-1)$, así que $k(p-1)(q-1)+1 = ed$. Es decir, lo que acabamos de demostrar es que:

$$(M^{ed}) \bmod p = M$$

pero eso sí, solamente en el caso de que M y p no tengan divisores comunes.

Veamos ahora el caso de que sí los tenga.

2) M y p tienen divisores comunes. Puesto que el número *1* no vale (por ser un divisor común trivial) y *p* es un número primo, la única posibilidad es que *M* sea divisible por *p*. Eso significa que cualquier potencia de *M* también será divisible por *p*, es decir, dará resto cero. En particular *M^(ed)* es exactamente divisible por *p*, o dicho con aritmética modular:

$$M^{(ed)} \bmod p = 0$$

Pero dar resto cero también es dar resto M. Si dividimos 54 entre 9 el resto es cero (54 = 9*6+0) pero también nueve, ya que podemos escribir 54=9*5+9. Por lo general no lo hacemos así, pero es perfectamente válido, de modo que nadie puede objetar a que nosotros escribamos:

$$M^{(ed)} \bmod p = M$$

Con ello, hemos demostrado que, en cualquiera de los dos casos (con o sin divisores comunes), se cumple la relación:

$$M^{(ed)} \bmod p = M$$

Eso es eso es lo mismo que decir que el número *(M^ed – M)* es exactamente divisible por *p*. A continuación, podemos repetir el mismo argumento para el número *q*. El resultado será:

$$M^{(ed)} \bmod q = M$$

que nuevamente lo podemos expresar diciendo que *(M^ed – M)* es exactamente divisible por *p*.

Y ahora viene el elemento final, que espero sea de su agrado. Si el número *(M^ed – M)* es exactamente divisible por *p*, y también por *q*, significa que también es divisible por *p*q*, es decir por *n*. Lo que significa que:

$$M^{(ed)} \bmod n = M$$

Y con eso cerramos el círculo.

3) RSA EN LA PRÁCTICA

En este punto enlazamos con los lectores que se han saltado el desarrollo matemático. Les pongo en antecedentes. Acabamos de demostrar que

$$C^d \bmod n = M^{(ed)} \bmod n = M$$

Lo que significa que, en aritmética modular, elevar a la potencia *e* y elevar a la potencia *d* son dos pasos inversos:

—Para cifrar, tomamos *e* y hacemos $C = (M^e) \bmod n$

—Para descifrar, tomamos *d* y hacemos $M = (C^d) \bmod n$

Para aplicaciones prácticas, les recuerdo que (*n*,*e*) son números conocidos por todos, y forman la clave pública; *d* es la clave privada.

La seguridad del sistema se basa en varios factores. El número *e* puede ser pequeño, pero si combinamos un valor muy pequeño de *e* con un mensaje *M* de longitud muy corta, un atacante puede descifrarlo con cierta facilidad. Para ello, basta con imponer a *M* una longitud mínima, añadiendo bits aleatorios de relleno si fuese necesario. Hay claves en uso que utilizan valores de *e* tan pequeños como 3. Sin embargo, la gran mayoría de las claves RSA utilizadas hoy día usan un valor *e*=65537. El motivo es que ese número, en notación binaria, se escribe como 10000000000000001, y tener tan pocos unos facilita mucho el cálculo.

También es necesario que *d* sea grande, y hay otros ataques criptoanalíticos que hacen recomendable algunas restricciones adicionales. Pero el elemento básico que garantiza la seguridad del algoritmo RSA es sencillo y contundente: la inviabilidad de pueda factorizar *n*. Si consiguiese extraer sus factores primos (*p*,*q*) podría construir $F=(p-1)(q-1)$ y también la clave de descifrado *d*. Debemos, por tanto, escoger números primos lo bastante grandes como para que no puedan ser factorizados en un intervalo de tiempo razonable.

Sin embargo, el concepto de “lo bastante grandes” ha cambiado sensiblemente con el tiempo. En 1977, Ron Rivest afirmó que factorizar un número de 125 dígitos (unos 415 bits) requeriría unos 40 000 billones de años. El número que propuso como ejemplo tenía 129 dígitos, y fue factorizado en 1994.

El problema fundamental consiste en que, con los años, nos hemos vuelto muy hábiles factorizando números primos. Por un lado, se han ido descubriendo algoritmos de factorización cada vez más eficientes. Por otro, la potencia de cálculo de los ordenadores ha aumentado de forma espectacular desde los años setenta. Los sistemas informáticos son no solamente más potentes sino también más baratos, lo que significa que cualquiera puede acceder a una gran potencia de cálculo. El primer ordenador capaz de pasar la barrera del gigaflop (mil millones de operaciones aritméticas por segundo) fue el Cray-2, puesto en marcha en 1985. Un iPad actual

supera esa velocidad.

Para ver cómo de grande es “un número primo lo bastante grande,” lo ideal sería ideal disponer de un conjunto de números *n=p*q* y ver en cuánto tiempo y con qué recursos pueden ser factorizados. Y eso es precisamente lo que hizo la empresa RSA Laboratories. Durante años retó a todos los criptólogos del mundo al *RSA Factoring Challenge*. Aunque el desafío no está activo, siguen mostrando ejemplos de números que no han sido factorizados[4]. Para que se hagan ustedes una idea, he aquí la lista vigente de los números factorizados hasta la fecha:

Número	Bits	Fecha
RSA-140	463	1999
RSA-155	512	1999
RSA-160	530	2003
RSA-576	576	2003
RSA-640	640	2005
RSA-200	663	2005
RSA-768	768	2009

El esfuerzo no es poca cosa, ya que todos estos ataques requirieron redes de ordenadores y grandes cantidades de tiempo y memoria. Pero incluso usted puede convertirse en un destructor de primos. El primer número de 512 bits (RSA-155), factorizado en 1999, necesitó una capacidad de cálculo enorme: casi un cuarto de trillón de operaciones. Impresionante en términos absolutos, pero un ordenador de sobremesa moderno podría hacerlo en un par de meses.

Usuarios con acceso a mayor potencia de cálculo pueden hacerlo aún mejor. Mientras escribo estas palabras, participo en un proyecto científico de la Red Española de Supercomputación. Me ha sido asignada una potencia de cálculo que se aproxima al teraflop (un billón de operaciones aritméticas por segundo). A esa velocidad, podría factorizar RSA-155 en apenas tres días. Y mi proyecto dura cuatro meses.

A tenor de la potencia de cálculo desplegada incluso por pequeños usuarios, no es de extrañar que las recomendaciones de los expertos apunten a claves RSA de al menos 2048 bits. Incluso hoy día, factorizar una clave de 1024 se considera tarea titánica, pero puesto que ya ha caído una de 768 bits, es mejor ir a lo seguro.

Las preferencias de los usuarios, influidos seguramente por las recomendaciones de los expertos, reflejan también esta preocupación. Las claves RSA se utilizan principalmente en comunicaciones seguras por Internet, tanto en navegación web (certificados X.509) como en correo electrónico (PGP). Un estudio realizado en 2012 sobre más de seis millones y medio de claves RSA muestra que solamente el 2.5% de los usuarios usa claves de 768 bits o menos. Los tamaños más comunes son 1024 (73,9%) y 2048 bits (21,7%)[5]. En sintonía con esta idea de que más vale prevenir

que lamentar, Microsoft emitió un aviso de seguridad en agosto de 2012, según el cual ya no se aceptarían certificados digitales que utilizasen claves de menos de 1024 bits[6].

4) LOS PRIMOS DE TEXAS INSTRUMENTS

A pesar de las recomendaciones de los expertos, hay aplicaciones que usan RSA de 512 bits, y que a pesar de todos los avisos siguen confiando en claves pequeñas. Eso significa que cualquiera que confíe para su seguridad en claves RSA de 512 bits o menos se arriesga a que su secreto deje de serlo en cualquier momento.

Eso fue precisamente lo que le sucedió a la compañía norteamericana Texas Instruments (TI). Esta empresa fabrica una gran variedad de calculadoras programables, que utilizan criptografía de clave pública para firmar su software. Oficialmente, el motivo es verificar si el sistema operativo es válido; en la práctica, sirve para controlar el uso de las calculadoras, impidiéndoles ejecutar programas no autorizados.

El 30 de julio de 2009, Benjamin Moody (probablemente, un seudónimo) afirmó haber factorizado el módulo de la clave RSA-512 usada por la calculadora TI-83+. La información, publicada en el foro United_TI[7], incluye información sobre los medios que necesitó: un ordenador con procesador Athlon64 a 1900 MHz, cinco gigabytes de disco duro, hasta 2,6 gigabytes de RAM y 73 días[8]. La información pronto se propagó por la Red.

La respuesta de TI fue fulminante y contundente: envió tanto a Moody como a otros internautas avisos exigiendo la retirada de los datos sobre la clave RSA de la calculadora TI-83. Invocando la ley DMCA (Digital Millennium Copyright ACT), afirmó que el cifrado le permitía proteger su propiedad intelectual:

"*Texas Instruments Incorporated ("TI") posee el copyright del software del sistema operativo de [la calculadora] TI-83 Plus. El sistema operativo de TI-83 Plus usa cifrado para controlar de modo efectivo el acceso al sistema operativo y para proteger sus derechos como propietario del copyright de dicho código. Cualquier uso no autorizado de esos archivos está estrictamente prohibido*"[9].

En un principio, las amenazas legales surtieron efecto, y son el motivo por el que el mensaje original de Moody ya no contiene la clave original[10]. Pero a la larga, de poco le sirvieron a TI sus bravatas. Otros internautas comentaron la noticia y algunos unieron fuerzas en un intento por romper otras claves RSA de TI. Animados por el éxito de Moody, y quizá indignados por el modo en que fue atacado, el grupo de usuarios de ticalc.com organizó un esfuerzo de computación distribuida, mediante el cual una tarea grande puede ser ejecutada en multitud de ordenadores que ceden ciclos de CPU no usados[11]. Para finales de agosto, se había factorizado tres claves más, y la información fue publicada en Internet por Wikileaks[12].

A mediados de septiembre, el proceso de computación distribuida se dio por concluido. Para finales de septiembre, ya se habían hechos públicos los datos de las claves para las siguientes calculadoras de Texas Instruments: TI-92+, TI-73, TI-89,

TI-83+/TI-83+ Silver Edition, Voyage 200, TI-89 Titanium y TI-84+/TI-84 Silver Edition, además de la claves de firma para fechado temporal (*date-stamp*) de las TI-73, Explorer, TI-83 Plus, TI-83 Silver Edition, TI-84 Plus, TI-84 Silver Edition, TI-89, TI-89 Titanium, TI-92 Plus y Voyage 200.

Entonces, los censurados recibieron ayuda legal. La EFF (*Electronic Frontier Foundation*) se ofreció para defender a los primeros internautas que habían publicado la primera clave. El argumento legal de defensa era que dichas claves no protegían realmente ninguna pieza de software propiedad de TI (en realidad, dicho software estaba disponible en la propia web de TI, sin cifrar), sino que "*sólo evitan que el propietario de una calculadora TI pueda usar software alternativo, una actividad que no infringe ninguna disposición de propiedad intelectual*"[13].

Los defendidos procedieron a restaurar las claves que habían retirado, y no hay noticias de que hayan sido molestados más por Texas Instruments. El fabricante de calculadoras, por su parte, respondió casi dos años después mediante un cambio técnico, una segunda clave RSA, de 2048 bits de longitud. El proceso de "validación" dura varios minutos, y según parece, es inútil debido a un fallo de programación que permite saltárselo[14].

El caso Texas Instruments puso sobre la mesa un tema muy controvertido: ¿pueden protegerse ciertos números con medidas legales? TI pretendía tener derechos de propiedad intelectual sobre los números primos que conformaban sus claves RSA. En cualquier caso, parece una discusión meramente académica, ya que la información viaja a gran velocidad por el ciberespacio, particularmente en los casos de censura; es lo que se conoce como Efecto Streisand.

A favor de Texas Instruments, es de justicia reconocer que en 1999, cuando sus calculadoras comenzaron a llevar claves RSA, 512 bits se consideraba bastante grande para proporcionar una seguridad más que aceptable. Por otro lado, una clave grande implica una gran cantidad de cálculos para las tareas de cifrado y descifrado, lo que significa que hay un incentivo para no usar claves demasiado grandes. Pero diez años después, TI seguía usando las mismas claves. Su negativa a adaptarse a los nuevos tiempos fue el primer error. Intentar acallar su problema técnico mediante amenazas legales fue el segundo.

Y ahora, siguiendo una vocación de servicio público, voy a proporcionarles a ustedes las claves RSA de Texas Instruments. Las publiqué en una de mis webs hace años, así que no creo que nadie me demande[15]. Por si acaso, le adjunto una copia de ellas en este mismo libro. Las tiene disponible en el Anexo. Que las disfrute.

TELÉFONOS MÓVILES

Hubo un tiempo en que tener un teléfono móvil era un símbolo de estatus social comparable a conducir un deportivo de lujo. El privilegiado dueño se pavoneaba con él en todo momento, lo dejaba bien a la vista y no lo dejaba ni para dormir. Parecía que le fuese a llamar un ministro de un momento a otro.

Si usted es lo bastante joven para no conocer aquella época, enhorabuena. Hoy, por el contrario, lo que llama la atención es no tener un móvil. Incluso yo tengo uno, y si preguntan a mi madre les podrá comentar que mi fobia a los móviles era legendaria. La tecnología móvil se ha convertido en algo tan común en nuestras vidas que resulta invisible, y hoy hablamos o tuiteamos con la misma naturalidad con que abrimos el grifo para beber un vaso de agua. Que se lo digan a mi sobrino de tres años: una vez lo perdí de vista un momento en una tienda Apple, y durante sus escasos segundos de libertad se las arregló para coger un iPod, encontrar la sección de juegos y ponerse a jugar con el Angry Birds.

Lo cierto es que, con los modelos actuales de móviles tipo smartphone, a los que solamente les falta hablar (y sí, algunos ya hablan y todo), es difícil resistirse. Los móviles se usan para hacer fotos, conectarse a Twitter, escuchar música y leer libros electrónicos. Son tan polivalentes que sirven incluso para efectuar llamadas telefónicas.

Y, por supuesto, disfrutan de un abanico amplio de vulnerabilidades. En la actualidad son noticia los fallos de aplicaciones que funcionan en smartphones; pero la telefonía móvil en sí, analógica al principio y ahora digital, conlleva sus propios problemas de seguridad. Veamos qué soluciones se aportaron para evitarlos, y en qué medida resultaron eficaces.

1) LOS ABUELOS DE LA SEGURIDAD TELEFÓNICA

Apenas cinco años después de que el teléfono hubiese sido patentado, un inventor norteamericano llamado James Harris Rogers solicitó en 1881 la primera patente sobre un dispositivo para evitar escuchas telefónicas no autorizadas. No consta que su invento tuviese utilidad inmediata en aquellos tiempos, pero conforme pasaron los años se hicieron diversos esfuerzos para proporcionar seguridad en las comunicaciones telefónicas.

En esto, los medios de comunicación tradicionales llevaban la delantera. Cuando se tendieron las primeras líneas telegráficas, se hizo evidente que un rival comercial podría obtener ventaja captando las comunicaciones ajenas. La intrusión es mucho más fácil en el caso de la telegrafía óptica (donde las señales se transmitían por medio de paneles móviles ubicados en torres especiales), y más aún en el caso de la telegrafía sin hilos que conocemos con el nombre de radio. En esos casos, una buena codificación de la información permitía enviar los datos de forma más eficiente a la vez que protegida.

La solución más sencilla era utilizar libros de código o lenguaje convenido entre las partes, siguiendo una tradición criptográfica heredada de los servicios diplomáticos. Durante finales del siglo XIX y comienzos del XX, se publicaron diversos libros de código para uso civil, con aplicación especial a las comunicaciones comerciales y de transportes. Un texto como *"el encargado de los negocios está ausente, repitan telegrama usando la clave adecuada"* se convierte fácilmente en *ATGOVLIARH*, *Adapertos Calzaio* o *05107 40781* gracias al Código Telegráfico Lieber de 1915. Esta técnica permitía ahorrar trabajo al operador y costes al emisor; pero también se podían modificar a voluntad para convertir los códigos en un sistema de comunicación segura.

Por supuesto, esta solución deja de ser útil en el caso telefónico. La comodidad e inmediatez de una conversación de viva voz se perdería por completo si los dos interlocutores tuviesen hablar con un código en la mano, cifrando y descifrando cada palabra. Es indudable que ayudaría mucho disponer de una solución técnica que permitiese llevar a cabo una conversación telefónicas fluida sin esos artificios. David Kahn describe en su libro *Codebreakers* cómo durante las décadas de 1920-30 se desarrollaron soluciones de seguridad para las operadoras radiotelefónicas. Las llamadas telefónicas protegidas eran un bien deseable no sólo para los usuarios habituales (diplomáticos, militares y otros empleados del gobierno) sino también para los primeros servicios de pago que se estaban estableciendo.

Las técnicas tradicionales pasaban por efectuar transformaciones a la conversación oral una vez ésta había sido transformada en una corriente eléctrica para su transmisión. Las diversas frecuencias que componían dicha corriente eran

alteradas y “barajadas” entre sí como si fuesen naipes. Los agudos se transformaban en graves, y viceversa. Se añadía ruido a la señal para enmascararla. Por supuesto, lo que haga un teléfono para mezclar la señal ha de ser deshecho por el teléfono que se halla en el otro lado de la línea.

Tras diversos intentos, la empresa Bell Telephone construyó un dispositivo llamado A-3. Se trataba del teléfono más seguro de la época y el gobierno de EEUU lo utilizó para sus comunicaciones de alto nivel con interlocutores en el extranjero. Durante la primera parte de la Segunda Guerra Mundial, Roosevelt estuvo en contacto telefónico frecuente con Churchill gracias al nuevo invento.

El problema es que el A-3 solamente proporcionaba seguridad contra un adversario débil, y no lograría nada frente a uno bien equipado técnicamente. Por supuesto, no serviría para nada frente a la Alemania nazi, cuyos ingenieros habían descubierto la forma de recuperar el sonido original. Los alemanes pronto instalaron un gran puesto de escucha en Holanda.

A día de hoy, sigue pendiente para los historiadores la tarea de averiguar hasta qué punto las escuchas alemanas del A-3 influyeron en el curso de la Segunda Guerra Mundial. Se sabe, por ejemplo, que el 29 de julio de 1943 los alemanes interceptaron una llamada entre Roosevelt y Churchill, relativa a la reciente destitución de Mussolini, y la consideraron como una prueba de que los italianos estaban planeando separarse del Eje. Podemos considerar como muy probable que dicha llamada ayudase a crear en Hitler el temor de que su antiguo aliado iba a desertar, lo que intentó impedir con la ocupación alemana de Italia en septiembre de ese mismo año.

La seguridad que el A-3 debía prestar al gobierno norteamericano contribuyó a una de las mayores derrotas en la historia militar de los Estados Unidos. El 6 de diciembre de 1941, una emisora de la US Navy captó una serie de 14 mensajes del gobierno japonés a su embajada en Washington. Como punto y final de una larga serie de negociaciones infructuosas, el embajador japonés recibió instrucciones de interrumpir las negociaciones con el gobierno norteamericano y destruir todas las claves y equipo criptográfico de la embajada. Este detalle, que suele ser el preludio inmediato a una declaración de guerra, no pasó desapercibido a los norteamericanos, quienes gracias al colosal trabajo del criptoanalista William Friedman habían conseguido descifrar los códigos japoneses.

El gobierno estadounidense podía leer los telegramas japoneses antes incluso que los propios operarios de la embajada, algo que ha llevado a algunos a conjeturar en una conspiración de Roosevelt para involucrar a Estados Unidos en la guerra. En cualquier caso, las hostilidades eran inminentes, y el territorio de Hawai era el primero en la lista de objetivos. El general George Marshall, jefe del ejército, lo sabía. Cuando recibió la información, los primeros aviones japoneses estaban despegando de las cubiertas de sus portaaviones. Todavía había tiempo para coger el

teléfono y alertar a las autoridades militares de Pearl Harbor.

Pero Marshall también sabía que su teléfono A-3 no era lo bastante seguro. Él mismo había advertido a Roosevelt al respecto. Transmitir una alerta de ataque inminente, dando a Pearl Harbor un par de horas críticas de preaviso, permitirá salvar vidas y quizá incluso repeler la agresión, pero revelaría que los norteamericanos conocían las claves japonesas. Marshall decidió proteger la información obtenida, y por tanto envió la alerta por medio del telégrafo. La tarea de componer, cifrar y enviar el mensaje llevó su tiempo, y a eso se unieron interferencias y problemas técnicos. El mensaje llegó finalmente a las manos del oficial al mando de las instalaciones de Hawai, general Short, a las tres de la tarde hora local. El ataque japonés había terminado al mediodía.

Resulta sorprendente que la técnica más avanzada fuese incapaz de proteger conversaciones telefónicas analógicas. El motivo de ello es sencillo: la voz es muy difícil de cifrar. El oído humano está especialmente bien “cableado” para escuchar y entender conversaciones. Podemos filtrar y entender una conversación en el seno de una fiesta ruidosa, con música de fondo y con varios diálogos a nuestra alrededor. El oído es una excelente máquina descifradora de sonidos.

La respuesta aliada a la inseguridad del A-3 fue la invención de los primeros rudimentos de la telefonía digital. A finales de los años treinta, los laboratorios Bell habían desarrollado una técnica para transformar señales de voz en datos digitales, un dispositivo conocido como *vocoder* (codificador de voz). A partir de ahí, diseñaron un instrumento para cifrar y descifrar conversaciones telefónicas. La idea es lo que hoy conocemos como libreta de uso único (OTP). Los bits correspondientes a la conversación se sumaban a bits de una “clave” que se guardaba en la forma de ruido aleatorio, y se transmitían; el receptor, a su vez, restaba el ruido (para lo cual tenía una copia del mismo ruido aleatorio pregrabado) y obtenía de nuevo la conversación original.

El cifrador que surgió recibió el nombre de *Green Hornet* (“avispón verde,” en referencia a un serial radiofónico) por el zumbido que generaba, pero posteriormente recibió el nombre clave de SIGSALY. Se trataba de un instrumento enorme, con un peso de unas 55 toneladas y que ocupaba una habitación entera. Se cuenta que Roosevelt, temeroso de que Churchill utilizase la nueva maravilla técnica a cualquier hora del día (o de la noche), se negó a instalarlo en la Casa Blanca, así que acabó en una habitación del Pentágono. El mandatario británico, por el contrario, tenía acceso a través de un pasillo especial que llegaba hasta sus habitaciones en su oficina de guerra.

SIGSALY fue inaugurado el 15 de julio de 1943, y fue un rotundo éxito. Los alemanes captaron sus mensajes, pero fueron incapaces de descifrarlos. Durante la guerra, formaron un tapiz que protegía las comunicaciones telefónicas aliadas de más

alto nivel[1], incluyendo las de los comandantes en los diversos teatros de guerra. Las propias patentes de SIGSALY se mantuvieron secretas hasta 1976, y cuando fueron hechas públicas, se descubrió que había sido un verdadero precursor de muchas técnicas habituales hoy día como la compresión de ancho de banda o la propia telefonía cifrada,[2].

Durante la Guerra Fría, se desarrollaron diversos tipos de teléfono seguro que utilizaban principios digitales. Su enumeración sería larga y, hasta cierto punto, aburrida. Los más conocidos son los de la gama STU (*Secure Telephone Unit*), que fueron usados para dotar de seguridad a los principales miembros del gobierno y contratistas militares. La familia STU-III fue creada en 1987, y fue uno de los teléfonos utilizados por el presidente Bush para recibir noticias sobre los atentados del 11-S.

Sin embargo, no existían teléfonos seguros para uso civil. La red de telefonía era analógica, lo que significa que la voz era convertida en señales eléctricas no digitales. En esas condiciones, como hemos visto, resulta muy difícil proteger la confidencialidad de lo que se habla por teléfono. La única posibilidad accesible al ciudadano medio (o incluso a una gran empresa) sería recurrir a un “mezclador” con principios similares al A-3, lo que solamente proporcionaría seguridad frente a un atacante de pocos recursos.

2) LLEGA LA TELEFONÍA MÓVIL

A pesar de carecer de seguridad criptográfica, el usuario de un teléfono fijo podía tener la seguridad de que la interceptación de su llamada no sería tarea fácil. Un atacante tendría que acceder físicamente, bien al cable telefónico, bien a una estación conmutadora.

La situación cambió por completo con el advenimiento de la telefonía móvil. Al contrario que la telefonía fija tradicional, en la que los teléfonos están físicamente anclados a la red de comunicación, ahora el terminal puede moverse libremente. El secreto radica en una red de estaciones de telefonía, que son las encargadas de enviar y recibir las llamadas a los terminales móviles. Podemos verlo como una combinación de teléfono tradicional y walkie-talkie, o como un teléfono inalámbrico de gran alcance.

La telefonía móvil comenzó a ser comercializada a gran escala en los países nórdicos en 1981. En España, tras dos intentos más o menos exitosos en 1976 y 1982, la compañía Telefónica lanzó en 1990 el sistema Moviline, que utilizaba señales de radio en la banda de frecuencia de 900 MHz[3]. Durante los años noventa, esta telefonía móvil de primera generación dominó el mercado en España. Se trataba de un sistema analógico, lo que implica un nivel de seguridad bajo, ya que el tramo aéreo entre el teléfono y la antena más cercana carecía de cualquier protección de cifrado.

El 26 de abril de 1991 saltó a la prensa un escándalo político que ilustró, entre otras cosas, la vulnerabilidad de los nuevos teléfonos móviles analógicos. Ese día, la cadena SER desveló dos conversación telefónicas del dirigente socialista José Marí ("Txiqui") Benegas. Realizadas el día 18, el contenido de las conversaciones revelaba las tensiones internas en el seno del PSOE. Benegas, que por aquél entonces era Secretario de Organización y hombre fuerte del partido, estaba enzarzado en una polémica con el ministro de Economía y Hacienda, Carlos Solchaga, acerca de un plan de financiación de viviendas[4]. En la conversación, un enfadado Benegas llamaba *enano* a Solchaga, y el propio presidente del gobierno, Felipe González, recibía apelativos chulescos como *Dios* y *el One*[5].

El desconocimiento sobre los detalles técnicos de la telefonía analógica hizo creer incluso a algunos funcionarios de alto nivel que su funcionamiento era similar al de un sistema de walkie-talkies. La Secretaria General de Comunicaciones, Elena Salgado, se aventuró a conjeturar que la grabación "*pudo haberse producido desde otro coche cercano*"[6]. El propio Benegas afirmó, según el diario ABC, que en el momento de la interceptación viajaba a 200 km/h, "*razón por la cual difícilmente pudo hacerlo un radioaficionado*"[7], declaraciones que fueron posteriormente matizadas por el ministro del Interior: "*Benegas nunca declaró ante el juez que fuera*

a 200 kilómetros por hora. Exactamente dijo: "En virtud de la velocidad a la que iba era muy difícil que se pudiera efectuar la grabación de manera casual" [8].

Aunque la actividad de la SER pudo haber constituido un delito de revelación de secretos, la cadena de radio se escudó en la libertad de prensa[9]. No dieron detalles técnicos, pero el director de la cadena afirmó que "*no se necesitan informes de expertos. La SER no realizó las grabaciones, ni las encargó. La persona que las captó no realizó ningún pinchazo. Las obtuvo de una manera casual e inocente*"[10]. Posteriormente, los periodistas señalaron a un radioaficionado, cuyo nombre no revelaron, como origen de la grabación, que calificaron de fortuita[11]. El juez encargado del caso declaró en junio que, en su opinión, la difusión de las conversaciones no era constitutiva de delito porque en ese caso el derecho a la libertad de expresión tenía prioridad sobre el derecho a la intimidad de una figura pública, y en octubre archivó las diligencias de la denuncia interpuesta por Benegas[12].

Un segundo caso, menos conocido, enfrentó de nuevo a la prensa con la justicia. Joaquín Abad, director del diario *La Crónica del Sur*, interceptó y publicó conversaciones llevadas a cabo el 23 de abril de 1990 por el empresario cinematográfico Juan Asensio con el exjefe de la Policía Local de Roquetas de Mar Rafael Montoya y con el periodista de *Ideal* Miguel Ángel Blanco. La defensa pedía su absolución, entre otros factores, porque "*las emisiones captadas eran de carácter público al ser por aire*"[13], y en su opinión el artículo del Código Penal que castigaba las interceptaciones solamente se refería a las comunicaciones por cable, no las inalámbricas. El tribunal no estuvo de acuerdo, y condenó a Abad en febrero de 1992. Entre otras cosas, el tribunal consideró demostrado que el propio Abad había realizado las escuchas (al contrario que en el caso Benegas). Los recursos posteriores que elevó el periodista confirmaron la sentencia, incluida una sentencia del Tribunal Constitucional en marzo de 1994.

Las escuchas telefónicas fueron mucho más allá de los casos de investigación periodística. A comienzos de los años noventa, saltaron a la prensa diversos escándalos sobre pinchazos telefónicos a altos cargos políticos en España, en una complicada red que involucraba al gobierno, las fuerzas policiales y el servicio de inteligencia del Estado (CESID). La complejidad y extensión de las escuchas hace prácticamente imposible determinar qué parte fue realizada sobre el tramo radio de la telefonía analógica, pero es indudable que representó una fuente de información fácil de obtener. Si cualquier periodista podía usar un escáner en su propio despacho para interceptar llamadas telefónicas, ¿qué posibilidad de protección tendrá el ciudadano de a pie?

La red Moviline tenía una gran cobertura a nivel nacional, lo que permitió la popularización de la telefonía móvil. El número de usuarios, que apenas pasaban de

los 50 000 en 1990, superó los 400 000 en 1994. Por primera vez, la telefonía móvil resultaba algo accesible al ciudadano medio tanto en precio como en prestaciones. Sin embargo, la transmisión de señales analógicas era problemática, se prestaba a interferencias y era incapaz de dar cobijo a la creciente demanda de servicios de telefonía. Lo cierto era que la falta de seguridad era el menor de sus problemas.

A la luz de estas y otras limitaciones, las principales redes europeas comenzaron a desplegar las redes de lo que ahora llamamos segunda generación de telefonía móvil, de tipo digital. Con ello se acabaría la época en la que las voces se transformaban en señales eléctricas analógicas. Ahora, la digitalización de la voz permitiría aplicar muchas técnicas de mejora, desde la compresión de datos a la corrección de errores; pasando, por supuesto, por la criptografía.

La primera red de telefonía móvil digital fue lanzada por Telefónica en julio de 1995, con el nombre de MoviStar. El consorcio Airtel (hoy Vodafone) comenzó a operar en octubre del mismo año, y en 1999 se autorizó una tercera red, llamada entonces Amena (hoy Orange). A pesar de su menor superficie de cobertura, la telefonía GSM tuvo un gran éxito en España. Los cuatrocientos mil clientes de telefonía móvil de 1994 pasaron a más de treinta millones en 2001[3b]. La implantación de los sistemas de telefonía digital de segunda generación (GSM) supusieron una fuerte competencia para el sistema Moviline, que finalmente echó el cierre en 2003[14].

Es en este punto donde los aficionados a la criptografía debemos prestar atención. Ahora ya no cabe excusa para enviar comunicaciones en el aire sin protección. Las operadoras eran muy conscientes de ello, y los protocolos de la nueva telefonía incorporaron mecanismos criptográficos de autenticación y cifrado. Y, al igual que otras grandes empresas, el proceso de desarrollo tuvo cuatro partes. Primera: se crean los algoritmos según el principio de “seguridad mediante oscuridad”. Segunda: los algoritmos son atacados por los criptólogos. Tercera: las operadoras niegan los fallos. Cuarta: se crean nuevos sistemas, más seguros.

Pero antes de examinar la situación de la telefonía habitual en España, vamos a dar un salto y examinaremos las desventuras del primo americano. En los Estados Unidos, la encargada de proporcionar estándares comunes para la telefonía móvil digital fue la Asociación de Industrias de Telecomunicación (TIA, *Telecommunication Industry Association*). El resultado, en lo que respecta a protección criptográfica, fue un conjunto de algoritmos. Los más relevantes son:

—Un sistema basado en sumas XOR para proteger la confidencialidad de los datos de voz.

—ORYX, para servicios de datos inalámbricos

—CAVE (*Cellular Authentication, Voice privacy and Encryption*), para autenticar el equipo móvil y generar claves criptográficas.

—CMEA (*Cellular Message Encryption Algorithm*), para proteger los datos de control, como el número de teléfono al que se ha llamado, o números PIN para identificación frente al banco por vía telefónica.

La elección hecha por la TIA no fue muy afortunada. En el primer caso, la suma XOR (⊕) se lleva a cabo con un paquete de datos (llamado "máscara") que se va repitiendo con el tiempo. Eso lo hace vulnerable. En efecto, supongamos que tenemos dos textos A, B que han sido sumados con el mismo segmento de clave, al que llamaremos K. El observador tiene solamente acceso a los paquetes cifrados (A⊕K), (B⊕K). Ahora bien, si sumamos ambos paquetes obtenemos:

$$(A \oplus K) \oplus (B \oplus K) = A \oplus B$$

con lo que recuperamos una suma de dos segmentos de datos en texto llano. El lector interesado tiene más información sobre la suma XOR en el capítulo "Los códigos de Dan Brown".

El segundo algoritmo, ORYX, no es mucho mejor. Se trata de un sistema que proporciona un flujo de datos pseudoaleatorios a partir de un sistema LFSR (*Lineal Feedback Shift Register*). En este caso, se utilizan tres LFSRs de 32 bits cada uno, lo que constituye en teoría un sistema de cifrado con clave de 96 bits. Algo impresionante si funcionase bien, lo que no es el caso. En 1998, un artículo firmado por David Wagner, Leone Simpson, Ed Dawson, John Kelsey, William Millan y Bruce Schneier detallaba un fuerte ataque criptoanalítico[15]. Conociendo tan sólo unos 25 bytes del texto llano, se podían recuperar los 96 bits de la clave, y por tanto tener acceso a todo el contenido del texto cifrado, con una complejidad de 2^16. Es decir, es como si estuviésemos utilizando una clave de tan sólo 16 bits. Es posible asimismo realizar un ataque conociendo tan sólo el texto cifrado y algunas características del lenguaje utilizado, para de nuevo obtener la clave. A todos los efectos, ORYX estaba desarbolado y hundido.

En cuanto al tercer algoritmo, CAVE, también le llegó su turno. Se trataba básicamente de una función hash con valores de salida de 128 bits. En 2004, William Millan y Praveen Gauravaram demostraron que es un algoritmo inseguro para funciones de autenticación e integridad de datos[16].

Finalmente le llegó al turno a CMEA, que fue atacado en 1997 por David Wagner, Bruce Schneier y John Kelsey. CMEA es un algoritmo de cifrado en bloque, con una clave de 64 bits generada por CAVE, y bloques de longitud variable. El ataque reveló que conocer del orden de 40-80 textos llanos conocidos permite rebajar la seguridad de CMEA al de un sistema con clave de 24-32 bits[17]. La conclusión de los autores es aplicable a otros casos similares, y no nos coge de sorpresa: "*nuestro criptoanálisis de CMEA subraya la necesidad de un proceso de revisión criptográfica abierto. Apostar por algoritmos nuevos es siempre peligroso, y los estándares*

propietarios y cerrados no llevan a buen puerto".

Las reacciones al ataque a CMEA no se hicieron esperar. La Asociación de Industrias de Telecomunicación Móvil (CTIA) se defendió minimizando los efectos del descubrimiento, afirmando que requería equipo muy sofisticado y declarando que el impacto real sería prácticamente nulo[18]. Los autores del artículo replicaron que la TIA podía haberlo hecho mucho mejor, y que en su lugar escogieron algoritmos patéticamente fáciles de romper[19].

Las respuestas de algunas telecos fueron recogidas por Bruce Schneier bajo el epígrafe de "*miren cómo corren las ratas*". Qualcomm, sin reconocer ni negar la veracidad del ataque, se ratificó en la necesidad de aumentar la seguridad para sus clientes[20]. Pacific Bell Mobile Services se quedó ancha tras limitarse a afirmar tajantemente que las escuchas telefónicas son actividades ilegales[21]; si bien tanto ellos como Omnipoint[22] y Powertel[23] afirmaron ser inmunes a este ataque. El motivo es que no usaban CMEA sino una tecnología de origen europeo conocida como GSM. "*Nuestra tecnología pone una sólida barrera contra intentos de intrusión electrónica*," llegó a afirmar Omnipoint.

El tiempo mostraría a esas empresas que confiar en GSM tampoco era la panacea, pero en aquellos momentos parecía la mejor apuesta. Los principales algoritmos de protección creados por la industria de telecomunicaciones norteamericana se limitaban a sistemas con claves bastante grandes (64 a 96 bits) que, con un poco de ingenio, se convierten en algoritmos de juguete. Se desarrolló una versión mejorada de CMEA, llamado CMEA-I, pero su seguridad no resultó ser mucho mejor, ya que los investigadores franceses Thomas Chardin y Raphaël Marinier utilizaron en 2008 un ataque similar al de Wagner, Schneier y Kelsey para demostrar que el nuevo algoritmo no era mucho mejor que el antiguo. En sus propias palabras, "*esto demuestra que las mejoras hechas a CMEA son inadecuadas para evitar estos ataques, y confirman que la seguridad de CMEA y sus variantes deber ser reconsiderada desde el principio*"[24].

3) LA TELEFONÍA GSM: ACIERTOS Y FALLOS

El sistema de telefonía móvil utilizado en Europa se denomina GSM. Las iniciales significan *Group Speciál Mobile*, o bien *Global System for Mobile Communications*, que en esto no parece haber acuerdo. Cualquiera que sea su nombre real, representa un grupo de estándares gestionado por la *GSM Association*[25]. Entre esos estándares se encuentran los necesarios para la autenticación y cifrado.

Para entender el uso que se hace de la criptografía en la telefonía GSM, hemos de considerar los diferentes elementos que componen el sistema. Por un lado, está el terminal telefónico; por otro… bueno, en realidad hay un conjunto de subsistemas: la antena o estación base, el centro de conmutación, el de autenticación y otros más. Para simplificar, hablaremos simplemente de tres elementos: el móvil, la estación (o antena) y la red.

La estructura de seguridad se configura en dos fases: autenticación y cifrado. En la primera fase, el móvil y la estación han de asegurarse de que "el otro" está autorizado para entrar en la conversación; en la segunda fase, se cifra la comunicación. Por supuesto, ello conlleva la necesidad de intercambiar alguna clave de cifrado de forma segura. Eso significa que, en conjunto, necesitaremos tres algoritmos criptográficos:

—A3 para la autenticación

—A8 para la generación de claves

—A5 para el cifrado de la conversación

Vamos a comenzar el proceso de negociación entre el móvil y la estación. Comienza por una petición de sesión segura por parte del móvil, una especie de "hola, soy yo, ¿puedo conectarme?" La estación, escéptica y con buen motivo para ello, envía la solicitud al llamado centro de autenticación, quien responde enviando un triplete de datos: en primer lugar, una clave *Ki*; en segundo lugar, un paquete de datos aleatorios *RAND*; y en tercer lugar, un paquete llamado *SRES* (Respuesta Firmada), que se obtiene aplicando el algoritmo de autenticación A3 a partir de *Ki* y *RAND*. Es decir, podríamos escribirlo como *SRES*=A3*(Ki,RAND)*.

El motivo de hacerlo de este modo es que así se preserva el secreto de *Ki*. Esa clave solamente existe en dos lugares: en el centro de autenticación de la red, y en la tarjeta SIM del móvil. El móvil ha de demostrar que posee dicha clave *Ki*, pero no puede transmitirla por la red; y la estación no puede decir "eh, móvil, ¿es Ki tu clave secreta?", porque si lo hiciese cualquiera podría captarla.

El problema estriba en que ambas partes deben convencerse mutuamente de que conocen un secreto, pero sin revelarlo. Para conseguirlo, el sistema le dice al móvil: "si eres tú de verdad, toma este paquete *RAND* que te envío, pásalo por el algoritmo A3 usando tu clave *Ki*, y envíame la respuesta". Y eso hace el móvil. Si el *SRES*

calculado por el móvil es el mismo que el generado por el centro de autenticación, eso significa que el móvil realmente posee la clave *Ki*, y por tanto, es quien afirma ser.

Puede parecer que el paquete *RAND* aleatorio es superfluo, pero no lo es. Imaginemos que el paquete *SRES* solamente se calculase a partir de la clave *Ki*. En tal caso, un atacante podría captar A3*(Ki)* en cualquier momento. No sabe qué vale *Ki*, pero sí sabe que A3(Ki) es la respuesta que hay que dar a la red para hacerse pasar por el interlocutor válido; así que en cualquier caso no tiene más que decir "*A3(Ki)*" y acceder al sistema. Por el contrario, el paquete de datos A3*(Ki,RAND)* varía con cada petición de conexión, de modo que un valor correcto le dice a la red que a) el móvil es del dueño de la clave *Ki*, y b) la petición es actual, no una grabación de hace dos semanas. Y, por supuesto, si ambos valores (el del móvil y el de la red) no coinciden, la red puede bloquear la comunicación, y en caso necesario informar de que un móvil ha sido robado. Este proceso tiene lugar cada vez que conectamos el móvil y también, como mínimo, cada vez que el móvil pasa a la zona de cobertura de otra antena distinta.

Puesto que no hace falta que nadie conozca la identidad de nadie cada vez que alguien llama, la red nos identifica con una identidad temporal, en la forma de un código llamado Identidad Temporal de Suscriptor Móvil (TMSI). Esto garantiza la privacidad, de forma que un intruso no puede conocer quién está al otro lado de la línea. Es algo así como un edificio con control de acceso donde, para no tener que llevar el DNI a la vista, el guardia de seguridad me entrega una acreditación que pone "visitante temporal autorizado número 812". La idea es que, una vez me he identificado a satisfacción en la puerta, no necesito ir mostrando mi identificación a todos, y solamente necesito demostrar que estoy autorizado a estar allí.

De acuerdo, ya estamos identificados ante la red. Ahora tenemos el problema de acordar una clave común a ambos para cifrar las conversaciones que hagamos en el futuro. Por supuesto, no vamos a usar nuestra clave *Ki*, porque es la "clave" de nuestra identidad como usuarios de móviles, y resulta demasiado valiosa para tontear con ella. En estos casos, se recurre a una clave de sesión a la que llamaremos *Kc*, válida únicamente para una sola llamada. Para obtener *Kc* se usa un algoritmo de generación de claves denominado A8. Como el A3, también usa el paquete aleatorio *RAND* y la clave *Ki* como datos de entrada. De este modo, *Kc*=A8*(Ki,RAND)* es calculada de modo independiente por el móvil y por la estación.

En realidad, tanto A3 (autenticación) como A8 (generación de claves) se basan en el mismo algoritmo. Lo que hace ese algoritmo es tomar *Ki*, *RAND* como datos de entrada y generar un paquete de 128 bits de salida. Los primeros 32 bit constituyen la respuesta SRES, y los últimos 64 bits forman la clave *Kc* para el algoritmo de cifrado A5. Ya hemos autenticado el móvil frente a la red, hemos obtenido una clave para

cifrar, y lo único que queda es hablar.

Ahora bien, un detalle muy importante que el lector tiene que tener claro es que A3, A5 y A8 no son los nombres de algoritmos concretos. La GSMA no apoyaba el uso de un algoritmo u otro, así que dejaba a las operadoras telefónicas que escogiesen sus favoritos con tal que cumpliesen ciertas características técnicas. Para entendernos, es como si usted, al comprar un piso, tuviese libertad para escoger la caldera de gas que quisiese. Cualquiera vale, con tal de que cumpla ciertos parámetros: diámetro de los tubos, caudal de agua, etc.

Ahora bien, si usted tiene mil cosas en la cabeza y no sabe gran cosa de calderas, su elección más probable será la que le sugiera el constructor. Eso pasó en la telefonía móvil GSM. La GSMA sugirió un algoritmo de autenticación/generación de claves A3/A8, de origen alemán, llamado COMP128, y un algoritmo de cifrado A5, al que a falta de nombre mejor dejó bautizado como A5 (con lo que A5 pasó a representar un algoritmo de cifra concreto, en lugar de "uno cualquiera"). Estos algoritmos fueron desarrollados por el Grupo de Expertos sobre Seguridad de Algoritmos (SAGE) del Instituto Europeo de Estándares de Telecomunicación (ETSI), con la advertencia de que cualquier operadora podría escoger el que desease. ¿Y qué sucedió? Pues que las telecos, que no son agencias criptológicas ni aspiran a serlo, se limitaron en su mayoría por adoptar los algoritmos que se le ofrecieron de serie.

Se trató de una elección que simplifica la tarea, pero se basaba en el supuesto de que los algoritmos eran eficaces y seguros. Y seguros, lo que se dice seguros… vale, le estropearé la sorpresa: no lo son. El fallo fue el mismo que el cometido por otras empresas usuarias de productos criptográficos: crear ellos mismos sus algoritmos, en un proceso cerrado y con seguridad mediante oscuridad, en la esperanza de que nadie más que ellos conocerán sus entresijos.

Los algoritmos de cifrado y autenticación GSM fueron desarrollados en secreto, e incluso hoy día sus detalles son confidenciales. Quien esto escribe realizó una petición oficial de información en 2001. La respuesta fue: *desafortunadamente, las reglas que rigen la distribución de las especificaciones de los algoritmos GSM solamente me permiten distribuirlas a los operadores y fabricantes GSM* (James Moran, Director de Seguridad y Fraude de la *GSM Association*, Mayo 2001). Incluso en la actualidad, los algoritmos de cifrado y autenticación GSM originales siguen sin ser oficialmente publicados y permanecen bajo control de la *GSM Association*[26].

A pesar de ello, los algoritmos se filtraron e hicieron públicos, o se reconstruyeron mediante ingeniería inversa, al menos desde 1994. En el caso del algoritmo COMP128, el algoritmo de autenticación y generación de claves, se filtró una copia en código fuente C, recreada por los investigadores Marc Briceno (de la Smartcard Developers Association), Ian Goldberg y David Wagner (ambos de la Universidad de California-Berkeley) en 1998[27]. Se trata de una función

unidireccional tipo hash, que recibe una entrada de 256 bits para producir una salida de 128 bits. Ya puestos en faena, realizaron un estudio criptoanalítico, y descubrieron que el algoritmo de autenticación de GSM es débil. Un fallo permite obtener la clave de 128 bits Ki. Los requisitos: acceso físico a la tarjeta SIM y unas ocho horas para efectuar un total de 150 000 interrogatorios electrónicos[28].

Para entender el mecanismo exacto del ataque hay que tener conocimientos criptográficos, pero vamos a simplificar un poco. En general, un algoritmo se puede ver como una máquina de entrada y salida. Los bits entran por un lado, se los mezcla cuidadosamente como las cartas de un juego de naipes, y luego salen. En el caso de un algoritmo de cifrado, ese proceso de mezcla ha de ser tal que no revele información en absoluto ni sobre la clave ni sobre la entrada de datos inicial. Pero si no está bien diseñado, es posible obtener información en ciertas circunstancias. Si el esfuerzo para obtener la clave es inferior al que necesitaríamos para probar todas las posibles claves, tenemos un ataque criptoanalítico con éxito.

En nuestro caso particular, tenemos la función que hemos llamamos A3, y que ahora viene representada por el algoritmo COMP128. Se trata de una función que toma un paquete de datos RAND como entrada, una clave *Ki*, y a partir de ahí produce un paquete de 128 bits. Matemáticamente podemos representarlo como COMP128(*Ki*,*RAND*). Su esquema interno está basado en la repetición de dos procesos: una función de compresión, que toma los datos y los combina con los bits de la clave; y una función de permutación, que cambia los bits de posición. Esta operación se repite un número determinado de veces de veces, y el resultado final son los bits de salida, que como vimos servían para el proceso de autenticación y para obtener la clave de cifrado de la conversación telefónica.

El problema es que el esquema de barajado es imperfecto, concretamente la función de compresión. En teoría, en cada ronda todos los bits de la clave deben estar bien “mezclados” con los datos de entrada, de forma que usamos un valor de *RAND*, o uno de la clave, que varíen siquiera en un solo bit, el resultado COMP128(*Ki*,*RAND*) sea totalmente distinto. Ese es el objetivo a conseguir. Debido a un fallo en un paso de la función de compresión, algunos bits de salida dependen solamente de unos pocos bits de entrada. Gracias a eso, se pueden escoger valores de *RAND* muy parecidos entre sí (llamémosles *RAND1* y *RAND2*) que producen la misma salida final, esto es:

COMP128(*Ki*,*RAND1*) = COMP128(*Ki*,*RAND2*)

Cuando sucede esto, recibe el nombre de “colisión”. En este caso particular, una colisión permite conocer parte de la clave. Concretamente, si *RAND1* y *RAND2* son iguales salvo los bits 1 y 9, se produce una colisión que revela los bits 1 y 9 de la clave Ki. A continuación tomamos otros valores de *RAND1* y *RAND2* que solamente

se diferencien en los bits 2 y 10, y cuando encontremos una colisión tendremos los bits 2 y 10 de la clave. Repitiendo el proceso acabaremos recuperando los 96 bits de *Ki*.

La obtención de la clave *Ki* representa un suceso muy grave, ya que está asociada unívocamente a la tarjeta SIM, lo que significa que un atacante que la conozca podría clonar la tarjeta. Por ello, resultó desconcertante la respuesta de la *GSM Association*. En primer lugar, negaron la mayor: según ellos, duplicar una tarjeta SIM no es clonarla porque la red no permite el uso de dos teléfonos con el mismo número al mismo tiempo (algo como mínimo discutible). Luego afirmaron, tan tranquilamente, que copiar una tarjeta SIM "no es nada nuevo," pero que requiere un equipo electrónico muy sofisticado. Sin embargo, fueron desmentidos por el grupo hacker alemán Chaos Computer Club, que pronto consiguió aplicar el ataque Wagner-Briceno-Goldberg (WBG) y clonar con éxito un móvil[29], llegando incluso al extremo de hacer público el código informático que utilizaron[30].

Un clavo (ardiendo, en mi opinión) al que se agarraron los representantes de la industria fue el hecho de que el ataque requería acceso físico[31], algo en lo que también hizo hincapié la Asociación de Industrias de Telecomunicación Móvil (CTIA), quienes también esgrimieron el hecho de que clonar una tarjeta de móvil es un delito federal[32]. Una empresa de seguridad relacionada con temas de telefonía y redes afirmó que el nuevo ataque no permitía clonar tarjetas SIM por el aire (es decir, sin presencia física), pero al mismo tiempo confirmaba que el ataque WBG era "*verdadero, reproducible y técnicamente sólido*"[33].

Para los lectores interesados en una refutación más a fondo, les recomiendo consultar el excelente compendio que en su día hizo Jesús Cea Avión[34]. Yo quisiera subrayar un punto importante: el ataque no requería necesariamente acceso físico. Los autores del estudio WBG afirmarían más tarde que

"*Extensas conversaciones con ingenieros de GSM nos hacen concluir que los ataques 'por el aire' deben ser considerados posible en la práctica para un atacante sofisticado. No hemos intentado aún construir una demostración de laboratorio (parece que sería ilegal, bajo las leyes de EEUU, hacer este tipo de investigación), pero los expertos de GSM con los que hemos hablado han confirmado que sería posible en teoría y en la práctica. Han informado de que diversos aspectos de los protocolos GSM se combinan para permitir montar el ataque matemático de entrada escogida sobre COMP128, si se pudiese construir una estación base falsa. Tal estación base no necesita soportar todo el protocolo GSM, y podría ser construida por unos 10 000 dólares*"[35].

En cualquier caso, sea por el aire o con acceso físico, el ataque WBG puso contra las cuerdas a las telecos por varios motivos. Primero, porque un algoritmo de autenticación supuestamente robusto caía de forma estrepitosa. Segundo, porque la

industria vio expuestas sus vergüenzas al aire al escoger, casi sin excepción y sin evaluación independiente, el algoritmo por defecto en lugar de escoger uno propio. Tercero, y fue algo que en su momento no se subrayó, el ataque WBG no solamente permite clonar teléfonos, sino algo mucho peor: clonar estaciones de telefonía móvil. La "estación base" descrita en el párrafo anterior podría servir como un centro para interceptar llamadas, lo que constituiría una valiosa herramienta para agencias de inteligencia y otros curiosos no autorizados.

Una de las soluciones que adoptaron algunas operadoras de telefonía fue restringir la vida de sus tarjetas SIM a 2^16 (65 536) operaciones. Un investigador que en 2003 presentaba una mejora del ataque WBG sugirió esta misma idea, y añadió que "*muchos teléfonos GSM no harán 50 000 llamadas en toda su vida*"[36]. En efecto, ayudaría a proteger hasta cierto punto contra el ataque WBG original, pero no contra otros ataque mejorados que podrían producirse en el futuro, ya que —no hay que olvidarlo— los ataques criptoanalíticos van mejorando en potencia y prestaciones con el tiempo. Lo que hoy requiere 150 000 interrogatorios electrónicos puede necesitar solamente 1000 mañana. Como ejemplo, un artículo de 2002, de Josyula Rao y otros, mostró un ataque que solamente requiere ocho textos llanos escogidos por el atacante[37]. Su duración es del orden de minutos.

Puesto que el fallo correspondía al algoritmo COMP128, lo lógico sería sustituirlo, o como mínimo modificarlo; y así se hizo. El viejo algoritmo se rebautizó como COMP128-1, y a continuación se redactaron dos versiones: COMP128-2 y COMP128-3. No han sido publicados, así que no se conocen sus detalles ni su fortaleza, y tampoco se sabe cuándo fueron utilizados para sustituir a COMP128-1 o qué redes de telefonía los usaron, aunque hay referencias que indican que COMP128-1 seguía siendo utilizado por algunas operadoras a finales de 2009[38]. Al parecer, COMP128-2 fue diseñado para contrarrestar el ataque WBG, en tanto que COMP128-3 fue una "mejora" para fortalecer la calidad de la clave de cifrado de la conversación[39]. El motivo por el que he escrito "mejora" entre comillas lo veremos en breve. No se lo pierdan, porque no tiene desperdicio.

Comencemos con A5. En teoría, los detalles del algoritmo son secretos. Cuando un instituto de investigación o una operadora telefónica recibe la información técnica de algoritmos secretos, lo habitual es firmar antes un acuerdo de no divulgación por el que se comprometen a mantener confidencial la información recibida. Pero a veces hay filtraciones. Una operadora de telecomunicaciones británica envió las especificaciones de A5 a la Universidad de Bradford, pero olvidó decirles que firmasen el acuerdo de confidencialidad. Eso permitió que los documentos se filtrasen. En la actualidad, hay multitud de copias del algoritmo en Internet, como la que se guarda en[40], y eso nos brinda la oportunidad de examinarlo.

Desde un punto de vista técnico, podemos clasificar el algoritmo A5 como una

cifra de flujo (*stream cipher*). Tres registros LFSR de 19, 22 y 23 bits, se combinan para producir una clave pseudoaleatoria que se suma (XOR) al mensaje para obtener el mensaje cifrado. A efectos prácticos, por tanto, A5 funciona con una clave de 19+22+23=64 bits, lo que lo haría mucho más resistente a ataques de fuerza bruta que otros algoritmos de la época como el conocido DES. Según Bruce Schneier cuenta en su libro *Applied Cryptography*:

"*Se está haciendo evidente que las ideas básicas tras A5 son buenas. Es muy eficiente. Pasa todos los test estadísticos; su única vulnerabilidad conocida es que los registros son lo bastante cortos para hacer posible una búsqueda exhaustiva*".

La vulnerabilidad mencionada por Schneier se basa en que existe un ataque trivial: suponiendo conocidos los dos primeros LFSR, y conocida la clave pseudoaleatoria, se puede deducir el estado del tercer LFSR. Eso significa que la fortaleza del algoritmo es solamente equivalente a la de un algoritmo de 39 bits. El criptoanalista británico Ross Anderson llegó a afirmar que, por dicho motivo, A5 debería estar libre de controles a la exportación[41]. En aquella época todavía existían fuertes controles a la exportación de material criptográfico fuerte, y aunque a mediados de los años 90 la Unión Soviética había desaparecido, los países occidentales temían un mal uso por parte de países como el Irak de Sadam Hussein. Por ese motivo, había dos "sabores" de A5: el A5/1, reservado para uso doméstico, y el A5/2 debilitado para exportación. Puede el lector acceder al código fuente de ambas versiones en[42] y [43], respectivamente. Y, para ser correcto, existe una tercera variante, llamada A5/0, que significa "sin cifrado en absoluto".

Resulta paradójico que un sistema de cifrado demasiado fuerte para ser exportado acabe siendo débil en exceso. Toda la historia del algoritmo A5 siempre ha sido controvertida. Según Anderson y Schneier, cuando se comenzó a plantear la seguridad del algoritmo de cifrado GSM, allá por los años 80, hubo una fuerte trifulca entre las diversas agencias de inteligencia occidentales. Alemania, que compartía frontera con el Imperio del Mal (en palabras de Ronald Reagan) deseaban un algoritmo fuerte, pero otros países aficionados a poner la oreja se decantaban por un algoritmo débil, y finalmente estos últimos acabaron imponiéndose. El actual A5 es un sistema basado en una cifra naval francesa, especialmente eficiente cuando está implementado en hardware[44].

Rumores aparte, lo cierto es que una cifra con una fortaleza real de 40 bits entra dentro de lo que se consideraba tan débil que podía exportarse a terceros países, así que muy bueno no es. Eso, por supuesto, suponiendo que no fuese susceptible a otros ataques criptoanalíticos. Para su desgracia, lo fue, y enseguida veremos algunos ejemplos.

Antes hemos de explicar un detalle escandaloso. Cuando el grupo WBG efectuó su análisis del algoritmo COMP128, se reveló algo sorprendente. Como recordarán,

uno de los productos de salida de COMP128 era un paquete de 64 bits (*Kc*) que funciona como clave de sesión para cifrar la conversación. Bien, pues lo primero que descubrieron los investigadores fue que el algoritmo había sido debilitado, de tal forma que 10 de esos 64 bits ¡son siempre iguales a cero! Sí, como lo oyen: un algoritmo que se vende con 64 bits teóricos, es en la práctica 1024 veces más débil.

La *GSM Alliance* intentó explicar este estropicio dándole la vuelta a la tortilla: no es que hayamos debilitado el algoritmo, dijeron, sino que los diez bits adicionales "*proporcionan a los operadores [de telefonía] una flexibilidad adicional como respuesta a amenazas de seguridad y de fraude*" [31b]. Decida el lector lo que quiera creer. En cualquier caso, si la versión COMP128-2 se desarrolló para contrarrestar el ataque WBG, el escándalo de los diez bits obligó a la industria a desarrollar una nueva variante que, por fin, proporcionase una clave de 64 bits, y esa fue la COMP128-3. Por eso antes la llamé "mejora," ya que aunque técnicamente es una mejora respecto a la situación anterior, no hacía sino volver al supuesto *statu quo ante*.

Ahora que los algoritmos A5 (en sus dos variantes) eran públicos, los ataques criptoanalíticos se sucedieron. Comencemos por la versión de exportación A5/2. Es bastante parecido a su "hermano mayor" ya que también cuenta con registros LFSR de 19, 22 y 23 bits. Se diferencia en el mecanismo que hace funcionar esos registros: ahora los tres registros se ponen en marcha y se detienen siguiendo las instrucciones de un cuarto LFSR, que actúa como un reloj.

Al ser destinado específicamente para la exportación, no nos sorprenderá descubrir que había sido debilitado. El mismo equipo (Wagner, Briceno, Goldberg) descubrió en 1999 un ataque criptoanalítico contra A5/2, que presentaron de forma oral en el simposio *Crypto 99* de Santa Bárbara, California. Aunque nunca se molestaron en darle forma de artículo, las descripciones que hicieron informalmente nos dan suficientes detalles: se necesita una pequeña cantidad de mensaje cifrado, y el factor de trabajo es de 2^16[45]. Es decir, A5/2 es tan débil como un buen algoritmo de 16 bits. Esto significa que los ataques pueden hacerse en tiempo real con equipo sencillo… así que ya podemos imaginarnos el banquete que se han dado las agencias de espionaje electrónico como la NSA o la GCHQ durante estos años. A todos los efectos, es como si no se cifrase nada.

Las propias operadoras lo reconocieron, pero tardaron mucho en tomar medidas al respecto. Parece que la idea subyacente era que, puesto que A5/2 es débil y todo el mundo lo sabe, ¿qué importa? El ataque WBC ponía en evidencia la vulnerabilidad de un algoritmo ya de por sí débil. Pero finalmente, las consideraciones económicas pesaron más que las de seguridad. Ya no interesaban tanto tener un algoritmo débil para la exportación. La Guerra Fría había acabado, y si la telefonía GSM era incapaz de proporcionar cifrado mínimamente fuerte a terceros países, otros lo harían. Los

Sadam Hussein del mundo podían ser espiados por otros medios.

La desaparición de A5/2, por tanto, fue cuestión de tiempo. De mucho tiempo. Un interesante seguimiento hecho por Harald Welte en 2010 nos muestra que en fecha tan tardía como 2006, la oposición más fuerte a la retirada de A5/2 provenía de algunos operadores móviles norteamericanos, algo inusitado si se considera que EEUU estaba en la lista de países que podían utilizar la versión fuerte A5/1. Eso significa que las comunicaciones GSM norteamericanas eran tan vulnerables como las de cualquier país tercermundista, algo que realmente hace pensar de qué lado están agencias como la NSA, por no hablar de las propias operadoras, que necesariamente habrían de saber lo que se estaba cociendo. La retirada del algoritmo A5/2 no fue decretada hasta julio de 2007[46].

Eso por lo que respecta a la versión débil para la exportación. Veamos ahora qué pasó con A5/1, una variante supuestamente tan fuerte que nadie fuera de los privilegiados países occidentales podía disfrutarla. Como ya hemos visto, del dicho al hecho hay mucho trecho. Incluso dejando al margen la jugarreta de los diez bits nulos, ya hemos visto que, de entrada, su fortaleza es equivalente a la de una cifra de 39 bits, no más.

Esta conjetura fue confirmada por Jovan Golic en 1997[47] al diseñar un ataque teórico que, por desgracia, requería demasiado espacio de memoria para ser práctico. El trabajo de Golic utilizó lo que se denomina un compromiso (*trade-off*) entre tiempo y memoria, lo que significa que un atacante puede tener éxito en poco tiempo si antes ha hecho una extensa labor de cálculo previo. Este truco fue aprovechado en diciembre de 1999 por Alex Biryunov, Adi Shamir y David Wagner para diseñar un ataque contra A5/1[48]. Primero es necesario un trabajo previo de cálculo de unos 2^28 pasos, tarea difícil pero que solamente hay que realizar una vez. Después, aprovecharon diversos “fallos sutiles” para recuperar la clave de cifrado.

El atacante puede incluso escoger el tipo de ataque. En un caso se requiere el equivalente a un par de segundos de conversación cifrada y el ataque requiere unos minutos; en otro, es preciso obtener dos minutos de conversación, pero la obtención de la clave lleva apenas un segundo. Este ataque NO aprovechó el hecho de que diez de los 64 bits de la clave fuesen cero. Los autores dejaron caer que “*es un problema abierto muy interesante el ver si podemos hacer nuestro ataque más rápido suponiendo que los diez bits son cero*”.

En esta ocasión, las operadoras de telefonía no podrían replicar que “es un ataque muy sofisticado que requiere equipo técnico especializado,” ya que el material que usaron los investigadores fue un PC con 128 MB de memoria, una capacidad de almacenamiento equivalente a entre dos y cuatro discos duros de 73 GB, un escáner digital para captar conversaciones (esto no lo dicen los investigadores, pero se sobreentiende), y poco más.

A pesar de su simplicidad técnica, en declaraciones al New York Times, un portavoz de la compañía Omnipoint (la teleco que presumía de ser inmune a los ataques contra el algoritmo norteamericano CMEA tres años antes), calificó los resultados como "*ridículos… lo que están describiendo es un ejercicio académico que nunca funcionaría en el mundo real*"[49]. Por supuesto, en estos casos hay que tener en cuenta que, si los investigadores hubieran demostrado su "ejercicio académico" en el mundo real, habrían acabado en prisión.

Mundo real o no, los investigadores demostraron que el algoritmo de A5/1 es vulnerable. Y más aún, según Bruce Schneier:

"*Lo que es más interesante sobre esos algoritmos es lo robustamente malos que son. Ambos mecanismos de cifrado de voz tienen fallos, pero no obvios. Los ataques tanto sobre A5/1 como sobre A5/2 hacen uso de estructuras sutiles del algoritmo, y resultan en la habilidad de descifrar tráfico de voz en tiempo real con equipo informático medio. Al mismo tiempo, la salida de algoritmo A8 que proporciona material de clave para A5/1 y A5/2 ha sido debilitada artificialmente ajustando diez bits a cero. Y asimismo, el algoritmo COMP128 que proporciona el material de clave… es débil*"[50].

A la vista de todo ello, quedaba claro ya en 1999 que "seguridad GSM" era una contradicción en sus términos. Tristemente, se trataba del sistema de telefonía móvil más seguro, lo que no contribuye precisamente a tranquilizarnos.

Por el contrario, ahora que se conocían los detalles del algoritmo A5/1, se desarrollaron otros métodos de ataque. Uno de ellos aprovechaba correlaciones estadísticas entre los datos de los LFSR y el flujo de clave resultante. La ventaja de este procedimiento de ataque es que no exige largas etapas de procesamiento previo ni discos duros enormes. El primer ataque de este tipo se debe a Patrik Ekdahl y Thomas Johansson en 2003, a los que siguieron otros en 2004 (Alexander Maximov, Thomas Johansson y Steve Babbage) y 2005 (Elad Barkan y Eli Biham). En general, se requiere un ordenador tipo Pentium 4, un conjunto de material conocido (texto no cifrado) y un tiempo de procesamiento de entre dos y diez minutos, con una tasa de éxito variable de entre el 3% y el 99.99%[51].

Al año siguiente, Barkan y Biham, con la colaboración de Nathan Keller, asestaron un golpe aún mayor a la serie A5 en sus tres sabores conocidos (doméstico, exportación y nulo). Hasta entonces, la mayoría de los ataques conocidos eran del tipo "texto llano conocido" (*known-plaintext*), lo que significa que hay que conocer parte de la conversación. Esto podía conseguirse en algunos casos, por ejemplo suponiendo momentos en los que no hay conversación, pero en general había que confiar en tener conocimiento previo del contenido en texto llano. Hasta cierto punto, nos recuerda los ataques criptoanalíticos aliados contra la máquina cifradora Enigma durante la Segunda Guerra Mundial, donde conocer ciertas palabras (conocidas en

inglés como *cribs* o “chuletas”) en los mensajes permitía obtener las claves.

El título del artículo de Barkan, Biham y Keller lo dice a las claras: “*Criptoanálisis instantáneo mediante texto cifrado de una comunicación cifrada por GSM*”[52]. En este caso no hay que suponer nada sobre la conversación, sino que basta con captar el contenido cifrado. Comenzaron por el débil A5/2, y demostraron que bastaba con unos milisegundos de conversación cifrada y un segundo de procesamiento en un PC. Para ello, se aprovecharon de un cambio en el orden de dos operaciones esenciales: el uso de cifrado (para proteger la comunicación) y de códigos de corrección de errores (para detectar la existencia de fallos de la transmisión).

La regla en estos casos suele ser: primero cifrar, luego añadir códigos de corrección (“*first encrypt then MAC*”). No se suele hacer al revés, porque entonces un atacante podría montar un ataque del tipo “texto cifrado conocido” (*known-ciphertext*). El problema con el protocolo GSM es que primero aplica la corrección de errores y luego el cifrado, justo el orden incorrecto. Eso permite una vía de entrada a un atacante hábil. Eso sí, necesitaría gran cantidad de memoria y ordenadores potentes, pero la idea estaba demostrada: se puede descifrar una conversación examinando el contenido cifrado.

Los investigadores también diseñaron ataques sobre el algoritmo A5/1, más resistente. Sin embargo, pronto se dieron cuenta de un detalle: ¿por qué esforzarse para romper un cifrado difícil cuando pueden persuadir al móvil de que no utilice cifrado en absoluto? Esto puede hacerse por un fallo en el proceso de autenticación del protocolo GSM. El problema es que el móvil se autentifica ante la estación telefónica, pero no al revés. Es decir, el móvil asume por defecto que la señal más fuerte en su proximidad se corresponde con una antena legítima de su operadora. Eso significa que, en principio, un atacante puede montar una estación telefónica falsa para engañar al móvil, aun careciendo de la clave de autenticación *Ki*. La conecta y el móvil considera que está en la zona de cobertura de una antena legítima.

De ese modo, el atacante puede hacer cosas como escoger el algoritmo de cifrado usado, incluyendo la opción de no usar cifrado en absoluto. El móvil puede recibir la orden de, por ejemplo, usar A5/2 (puede que el móvil avise al dueño cuando no se está usando cifrado, así que la opción “no cifrar” puede no ser la más aconsejable). De ese modo, se puede obtener fácilmente la clave de cifrado *Kc*, con lo que el atacante ya puede descifrar las comunicaciones que efectuó el móvil desde que se autenticó por última vez con la estación legítima.

El atacante también puede captar conversaciones en tiempo real mediante la técnica del hombre interpuesto (*man-in-the-middle*). Para el usuario, el intruso aparece como la red; para la red telefónica, el intruso es otro usuario legítimo. En este caso, el intruso hace de correveidile entre la red y el usuario, pero con una sutil

diferencia: instruye al móvil del usuario para que use cifrado débil A5/2, de modo que el intruso tiene acceso a la conversación telefónica en tiempo real. Fíjese el lector que este tipo de ataques no se realiza sobre el algoritmo de cifrado, sino sobre el protocolo de comunicaciones GSM, y que por tanto funcionaría incluso en el caso de un algoritmo de cifrado matemáticamente impenetrable. Existen soluciones contra este ataque, pero son básicamente parches que requieren requisitos extra de vigilancia por parte de la red para protegerse contra un tipo de ataques que un sistema bien diseñado no debería temer.

Y aún no hemos terminado. Cuando comenzaron a operar las redes GSM, incluso una clave de 54 bits era algo serio (luego se convirtió en 64 una vez se hubo calmado el furor por el descubrimiento de que 10 bits eran cero). Atacar un sistema mediante técnicas de fuerza bruta quedaba fuera del alcance para el investigador o el hacker medio, y las agencias gubernamentales capaces de hacerlo callaban, pero hoy día cualquier persona tiene acceso a una capacidad de computación increíble, por no hablar de grupos de investigación. Para demostrarlo, un grupo formado por investigadores de las universidades alemanas de Bochum y Kiel presentaron en 2006 un dispositivo computador asequible creado con el fin específico de ayudar en los ataques criptoanalíticos. El nombre del aparato es COPACOBANA (*Cost-Optimizer Parallel Code Breaker*), y con un coste de unos 10 000 euros es capaz de probar todas las claves del algoritmo DES (de 56 bits) en un máximo de diecisiete días, algo impresionante si recordamos que estamos hablando de probar miles de billones de claves[53].

En A5/1, la clave de cifrado determina el estado inicial de los registros LFSR, y éstos producen un flujo pseudoaleatorio que usamos para proteger la conversación. En principio, podríamos ir calculando una tabla donde, para cada estado inicial, me de un resultado en forma de flujo de datos. Por ejemplo, el estado inicial A52F8C02 podría corresponder con el resultado 52E01001, lo que significa que, cuando ponga la oreja y escuche “52E01001,” ya sé que la clave corresponde al estado A52F8C02. De ese modo, creamos una tabla (*look-up table*) que nos liga claves y estados. Se trata de un ataque de fuera bruta puro y duro.

El problema es que, incluso para una clave de 54 bits, las necesidades de memoria y computación son enormes, millones y millones de terabytes. Eso no es práctico. Sin embargo, el proceso puede simplificarse “encadenando” bloques de estadios iniciales y resultados finales gracias a tablas alfanuméricas denominadas “tablas arcoíris” que son relativamente pequeñas, nada de millones de terabytes. Es uno de esos casos en los que el ataque supone un compromiso entre tiempo y memoria, lo que se conoce como TMTO (*Time-Memory Trade-Off*). El lector podrá encontrar más información sobre las tablas arcoíris en el capítulo “*12345*” de este mismo libro.

En 2008, los creadores de COPACOBANA evaluaron la utilidad de la máquina

para diversos compromisos tiempo-memoria en los que se utilizaban tablas arcoíris precomputadas. El ataque más rápido contra el algoritmo A5/1 tardaba solamente siete segundos, pero necesitaba más de 7 terabytes de datos y un conjunto de cálculos previos que llevaba casi tres meses, con una tasa de éxito del 60%[54]. Los compromisos permitían jugar con los principales parámetros, a saber: tasa de éxito, tiempo de computación previo y cantidad de datos.

En 2008, Timo Gendrullis, Martin Novotný y Andy Rupp mostraron al mundo sus resultados. En promedio, necesitaban seis horas para obtener la clave de cifrado completa. Es mucho tiempo para poder entrar en una conversación, pero demuestra que, incluso en el caso de que los algoritmos de protección en la telefonía GSM fuesen perfectos (que no lo son), un atacante podría ir a por todas y hacer una búsqueda exhaustiva. El coste de funcionamiento de COPACOBANA fue evaluado por los autores como 36 céntimos en electricidad. Y eso con una máquina hecha con un presupuesto inferior al precio de un utilitario[55].

No obstante, quedaba por ver si un esfuerzo conjunto de "internautas de a pie" podía crear tablas arcoíris para uso práctico contra A5/1 (los artículos relativos a COPACOBANA afirmaban que se habían calculado, pero no fueron abiertas para acceso público). En agosto de 2009, Karsten Nohl anunció el lanzamiento de un proyecto dedicado a crear tablas arcoíris para A5/1, demostrando así de una vez por todas que los algoritmos de GSM eran débiles[56]. Cuando efectuó su anuncio, la *GSM Association* (GSMA) se apresuró a declarar que eso queda muy lejos de constituir un ataque práctico: "*requiere la construcción de una tabla de aproximadamente 2 terabytes, equivalente a la cantidad de datos contenidos en una torre de libros de 20 kilómetros de altura*"[57].

Nohl estimó que, con la ayuda de colaboradores dispuestos a donar tiempo de computación, las tablas arcoíris podrían estar listas en tres meses. Y así fue. En diciembre de 2009, durante la celebración del *Chaos Communications Congress* de Berlín, Nohl desveló que el proyecto había concluido con éxito[58]. La GSMA se mantuvo en sus trece. Su razonamiento fue: no importa que la cripto sea vulnerable porque, de todos modos, no dicen cómo capturar las conversaciones del aire, y si lo intentan tampoco lograrán el éxito porque lo impediremos gracias a nuestros derechos de propiedad intelectual[59].

En mi humilde opinión, un grupo capaz de crear tablas arcoíris de 2 terabytes es perfectamente capaz de cualquier cosa. Las tablas arcoíris, ya calculadas, están disponibles en Internet, donde pueden ser descargadas vía torrent[60]. El software necesario para reventar una conversación GSM real está disponible desde julio de 2010[61], y en la actualidad el *Chaos Computer Club* está trabajando en el llamado proyecto AirProbe con el fin de demostrar que se pueden capturar paquetes GSM del aire, algo que la GSMA considera imposible[62].

A tenor de todo lo anterior, podemos concluir que el protocolo de seguridad GSM adolece de una alarmante falta de seguridad. El proceso de creación en secreto produce algoritmos vulnerables. Más allá de los problemas de un sistema de cifrado en concreto, lo cierto es que incluso en un caso perfecto la protección del sistema GSM es incompleta. Preocupados tan sólo por proteger el tramo aéreo de la comunicación, los responsables se olvidaron de prepararse contra otros tipos de amenazas. Algunos fallos que podemos señalar son los siguientes:

—Protección limitada del cifrado. Los algoritmos de cifrado, perfectos o no, solamente protegen el tramo entre el móvil y la estación base. Durante el resto del tramo, la llamada va "en claro", es decir, sin cifrado, lo que la hace vulnerable a una escucha no autorizada.

—Vulnerabilidad en la verificación. Al verificar una llamada, el centro de conmutación recibe del centro de autenticación un triplete de datos (clave+semilla+respuesta). Pero ese triplete se envía sin cifrar, por medio de líneas de comunicación vulnerables. Cualquiera que acceda a dichas líneas podrá obtener Ki, la clave del móvil, y con ello podrá clonarlo.

—Asimetría en la autenticación. El móvil se identifica ante la red, pero no al revés, y el móvil asume que las transmisiones se hacen a través de la red adecuada. No hay forma de poder asegurarse de que la estación base es la auténtica, ya que los diseñadores tomaron por evidente el hecho de que la red no sería duplicada. Las policías de varios países utilizan desde hace más de una década en sus investigaciones un equipo apodado *Stingray*, que no es más que una estación base falsa. El móvil es engañado por esta estación de pega, que puede ordenarle transmitir las llamadas ""en claro" para poder interceptarlas, así como darle un nuevo identificador temporal que permita identificarlo. Otros atacantes pueden montar estaciones falsas para interrogar móviles y extraer sus claves Ki.

—Falta de integridad. Existe autenticación y confidencialidad, pero no integridad. Es decir, no hay métodos para detectar si un paquete de datos ha sido alterado.

—Falta de flexibilidad. Los algoritmos para autenticación (A3/A8) y cifrado (A5) podían haber sido escogidos por las telecos, pero casi todas se limitaron a utilizar los que venían "de serie". Cuando éstos mostraron su vulnerabilidad, no existía suplente. La única defensa de la industria consistió en negar la evidencia, cosa que hace incluso en nuestros días.

A ello podríamos añadir la vulnerabilidad añadida de insertar servicios de "interceptación legal". En teoría, la interceptación legal solamente se utilizaría con orden judicial y bajo estricta supervisión legal, aunque en la práctica el sistema técnico sencillamente, está allí, listo para interceptar lo que le manden. Durante los años noventa, Estados Unidos y su poderosa Agencia de Seguridad Nacional (NSA)

presionaron a otros países para que incorporasen estas vías de acceso a las comunicaciones, que por supuesto podrían ser usadas (y abusadas) por agencias de inteligencia[63].

¿Paranoia, dijo usted? Durante los años 2004 y 2005, un grupo desconocido interceptó las llamadas de más de cien personalidades griegas, incluido el propio primer ministro, lo que provocó un gran escándalo mayor en Grecia[64]. Al parecer, en la red de telefonía móvil Vodafone había un módulo de interceptación ilegal. Poco tiempo después se conoció que dicho módulo ilegal era en realidad de lo más legal: estaba a disposición de la policía griega para obtener capacidad de interceptación legal.

El sistema de interceptación estaba oculto en los equipos conmutadores AXE que utilizaba Vodafone Grecia para sus operaciones. Dichos conmutadores, construidos por la empresa Ericsson, contenían una opción para proporcionar acceso legal a las autoridades policiales[65]. En lugar de acceder al contenido de la comunicación en el aire (donde está, como hemos visto cifrado), el acceso legal se lleva a cabo en estos conmutadores, que transportan el contenido de las llamadas sin cifrado alguno. El lector interesado puede leer muchos más detalles sobre el caso en[66].

Al parecer, el servicio de acceso legal era opcional, con un precio de varios millones de euros. El gobierno griego no pagó por esa opción, pero en su lugar un grupo desconocido implantó software que hacía dos cosas: activar la opción de interceptación legal, y al mismo tiempo borrar cualquier evidencia de que el sistema estaba activado. La primera noticia de su existencia provino de los problemas que algunos usuarios tenían durante el envió de mensajes; las quejas llegaron a Vodafone, quien solicitó a Ericsson una auditoría, y fue entonces cuando se descubrió que el sistema de pinchazos legales estaba activado[67].

Los detalles del caso no se supieron hasta febrero de 2006, cuando terminó la investigación judicial. Al parecer, Vodafone se había apresurado en borrar los registros (logs) y las propias copias del software de interceptación, lo que le valió una multa de varios millones de euros tanto a ellos como a Ericsson[68]. El técnico que ordenó el borrado fue encontrado muerto en su casa, aparentemente por suicidio, justo un día antes de que el primer ministro griego fuese informado del caso. Varios altos cargos técnicos de Vodafone fueron acusados de la intrusión[69].

A día de hoy, no se sabe quién estuvo detrás de los pinchazos telefónicos en Grecia. Lo único seguro es que, si se construye un sistema de interceptación de comunicaciones, alguien lo usará tarde o temprano. Y no siempre de forma legal. Desde bandas mafiosas a grupos de interés político, llegando incluso a los propios gobiernos, hay muchos interesados en cotillear los secretos ajenos.

4) GPRS

En los últimos años, se han popularizado servicios de Internet móvil como correo electrónico y navegación por páginas web (www), además de los conocidos mensajes SMS. El sistema GSM no fue diseñado para este tipo de tráfico. La solución, tanto para GSM como para el sistema de telefonía de tercera generación 3G, pasa por utilizar un protocolo para conmutación de paquetes de datos llamado GPRS (*General Packet Radio Services*); existe una versión mejorada que funciona bajo los nombres de EDGE o EGPRS.

GPSR utiliza para el proceso de autenticación el mismos algoritmo que GSM (el que antes llamamos COMP128), aunque utiliza su propia clave de cifrado Kc. Como vimos antes, ese algoritmo es vulnerable, por lo que GPRS es inseguro si se utiliza con algunas de las viejas versiones (COMP128-1, o -2). Además de ello, participa de los fallos de diseño existentes en las redes GSM, mencionados en el apartado anterior.

El sistema de cifrado, por el contrario, es diferente, y se denomina GEA (*Gprs Encryption Algorithm*). En realidad, hay más de un sabor de este algoritmo. GEA/0 indica el caso en que no haya cifrado, en tanto que GEA/1 y GEA/2 funcionan con clave de 64 bits. Como puede verse, recuerda un poco a los algoritmos A5 de la telefonía GSM. También es un algoritmo propietario, y sus detalles no han sido revelados oficialmente. Una ventaja de GEA es que protege la comunicación más allá de la antena de telefonía móvil, aunque sigue sin cubrir todo el trayecto que recorren los datos.

Lo que no significa que el cifrado sea bueno. En el *Chaos Communication Camp* de agosto de 2011, Karsten Nohl y Luca Melette mostraron varias vulnerabilidades que permitían derrotar el cifrado de GPRS. El método más sencillo consiste en montar una estación base falsa y forzar al móvil atacado a que no utilice cifrado, igual que en el caso de GSM con el algoritmo A5. Aunque no dieron detalles sobre los algoritmos GEA, mostraron que la versión GEA/1 es apenas mejor que A5/1, y es asimismo vulnerable a ataques como los de compromiso de tiempo-memoria *Time-Memory TradeOff* o los de fuerza bruta. Su conclusión fue que, puesto que GPRS transmite y recibe paquetes de datos IP como los de Internet, sería mejor utilizar protocolos de protección propios de Internet, como SSL[70].

Increíblemente, Nohl comentó en declaraciones al New York Times que algunas operadoras ni siquiera cifran los datos, es decir, pasan al modo GEA/0 de “no cifrado”. El motivo no es que esas operadoras sean descuidadas en temas de seguridad, sino que ese cifrado no les permitiría detectar servicios que no desean que utilicen sus abonados, como Skype. De ese modo, perjudican a sus clientes de forma doble: no permitiéndoles utilizar servicios IP legítimos, y dejando sus datos sin protección[71]. Nohl citó expresamente algunas operadoras italianas (Wind, Telecom

Italia), así como algunas alemanas que usan un cifrado sumamente débil (T-mobile, O2 Germany, Vodafone, E-Plus). Lo peor de todo: existe una versión GEA fuerte, con clave de 128 bits, pero ninguna operadora la utiliza todavía[72].

5) LA TELEFONÍA POR SATÉLITE

Un campo en el que las vulnerabilidades del sistema GSM provocaron, digámoslo así, bajas colaterales fue en la telefonía móvil por satélite. El lector lo habrá visto en el cine. Uno de los protagonistas habla por teléfono con un móvil grande y pesado, de esos que recuerdan a un ladrillo. Es grueso, negro y tiene una antena enorme. No se parece en nada a esos smartphones finos y estilizados que nos regalan al cambiarnos de operadora.

El éxito del sistema GSM en todo el mundo relegó a la telefonía por satélite a pequeños nichos de negocio, pero sigue existiendo. Resulta particularmente útil en zonas donde la cobertura de móvil tradicional es escasa o inexistente: zonas en conflicto, países subdesarrollados, alta mar. Thuraya, por ejemplo, proporciona servicio telefónico en zonas de Oriente Medio y África: La red Inmarsat, más conocida, permite comunicar en prácticamente cualquier lugar de la tierra y del mar.

En febrero de 2012, Benedikt Driessen y Ralf Hund, de la Universidad del Ruhr - Bochum decidieron averiguar qué tipo de sistemas de cifrado utilizan los teléfonos de Thuraya e Inmarsat. Descubrieron que los principales algoritmos criptográficos, propietarios y que se mantienen en secreto, tienen los nombres código de GMR-1 (Thuraya) y GMR-2 (Inmarsat), y utilizan claves de 64 bits.

Para destripar GMR-1, Driessen y Hund tuvieron un par de elementos a su favor. El primero consiste en cómo se utiliza el sistema de cifrado en el móvil. Como la telefonía satélite no tiene tantos usuarios y había que ahorrar, la operadora decidió implementar el cifrado en software, no en hardware. Eso significa que resulta mucho más fácil hacer ingeniería inversa. Solamente hay que extraer el programa donde esté guardado, o mejor aún, esperar a que hagan una actualización de firmware.

Driessen y Hund descubrieron que los protocolos de seguridad de los sistemas GMR son muy similares a los de la telefonía GSM. Los algoritmos de autenticación también se llaman A3/A8 y funcionan de la misma forma. En cuanto al algoritmo de cifrado de GMR-1, resultó ser muy similar al A5 (recibe el nombre de A5-GMR). No parece que se hayan esforzado mucho para hacer el sistema.

Eso no es malo *per se*, ya que los estándares GSM están ahí, y nada les impide en principio utilizarlos. Lo malo fue el descubrimiento de que GMR-1 es una modificación de A5/2, la versión para exportación del algoritmo de cifrado usado en telefonía móvil GSM. Imagino que esto sería por motivos de interoperabilidad: según la Wikipedia, el último teléfono de Thuraya puede usar una tarjeta SIM convencional (como la de su móvil, lector) y operar con redes GSM además de con los satélites. El problema es que, como ya hemos visto, A5/2 es tan resistente al criptoanálisis como una hoja de papel frente a una motosierra.

El algoritmo GMR-2 de Inmarsat no es mucho mejor. Aunque se trata de un

sistema diferente y no parece basarse en A5, tiene asimismo sus rarezas: parece que incorpora elementos del sistema de cifrado DES, un algoritmo de cifrado simétrico que no tiene nada que ver con telefonía. Con una cantidad mínima de texto cifrado y de computación, Driessen y Hund consiguieron extraer la clave del sistema. A la vista de la facilidad con que ambos tipos de cifrado cayeron, no es de extrañar que el artículo que escribieron lleve el esclarecedor título de "*No confíe en los teléfonos por satélite*"[73].

6) 3G, LA NUEVA GENERACIÓN

Mientras la telefonía GSM estaba en pleno auge, se iban poniendo los cimientos de una nueva tercera generación de teléfonos móviles. Se suponía que iba a representar una revolución en las comunicaciones, capacitando al usuario para efectuar videoconferencias y todo tipo de servicios en red. La realidad fue más sobria. Al altísimo coste de las licencias para los operadores telefónicos (en total, 50 000 millones de euros tan sólo en Alemania) se juntaba la dificultad técnica de poner en marcha una infraestructura nueva de comunicaciones, todo mientras los teléfonos GSM proporcionaban cada vez más y mejores servicios a los usuarios.

Pero finalmente la telefonía de tercera generación comenzó a desplegarse. El resultado se llama UMTS (*Universal Mobile Telecommunications System*). La arquitectura de este nuevo sistema, definidos por el 3GPP (*3rd Generation Partnership Project*) comparte elementos con la telefonía GSM, pero es muy distinto en otros aspectos. La integración con servicios de Internet es ahora mucho más profunda, y ni que decir tiene que todos los sistemas fueron profundamente reformados. No sólo entraban en juego algoritmos de cifrado y autenticación nuevos, sino que los fallos de concepto de la telefonía GSM fueron corregidos.

Los estándares de seguridad de 3GPP se dividen en cinco tipos:

1) **Seguridad en el acceso a red**. Es el conjunto de protecciones que proporcionan a un usuario un acceso seguro a la red 3G. En particular, protegen contra ataques en el tramo que se desarrolla en el aire (teléfono a estación base)

2) **Seguridad en el dominio del proveedor**. Proporciona una conexión segura entre los centros de autenticación y los centros conmutadores para el intercambio de datos

3) **Seguridad en el dominio del usuario**. Asegura el acceso del usuario al teléfono (para evitar que otro usuario no autorizado pueda usarlo). Se consigue mediante un código secreto similar al usado en la telefonía GSM (PIN)

4) **Seguridad en el dominio de aplicación**. Protege las comunicaciones de las aplicaciones (programas, servicios…) entre el usuario y el proveedor

5) **Seguridad de configuración**. Permite al usuario habilitar o deshabilitar diversas opciones de seguridad, como el uso de cifrado

El tipo 1 (seguridad en el acceso a red) es lo que vimos en páginas anteriores en el caso de la telefonía GSM, y es lo que vamos a ver aquí de nuevo. En términos generales, la seguridad en el acceso a red comprende cuatro puntos fundamentales:

1) Confidencialidad de la identidad del usuario

2) Autenticación e intercambio de claves

3) Confidencialidad de los mensajes

4) Integridad de los datos

El primer punto se resuelve, de forma similar al caso GSM, con la emisión de un número de identidad temporal. Los otros tres puntos constituyen el problema de autenticación e intercambio de claves, aunque más bien deberíamos hablar de "acuerdo", ya que realmente no intercambiamos la clave. El esquema es algo más complejo que en el caso GSM, puesto que ahora tienen que autenticarse tanto el teléfono como la red. Antes solamente era el móvil el que tenía que demostrar su identidad, pero como hemos visto existe la posibilidad de que una estación base falsa intente engañar al usuario, así que ahora serán las dos partes las que se identifiquen mutuamente. También hay que acordar una clave de integridad *IK* que impedirá alteraciones en los datos transmitidos o recibidos, además de la habitual clave para cifrado *CK*.

Los pasos iniciales son similares a los de la red GSM. El teléfono emite un mensaje a la red, o bien la red decide que hay que iniciar un nuevo proceso de autenticación e intercambio. Para ello, el centro de conmutación, envía un paquete de datos a la estación de telefonía que conectará con el móvil. En el caso GSM, dicho paquete era un triplete (*RAND*, *SRES*, *Ki*): un paquete aleatorio, la respuesta que deberá enviar el móvil y, si todo va bien, la clave de autenticación de la tarjeta SIM. Pero ahora lo que tendremos no es un triplete, sino un quintuplete:

a) Un paquete de datos aleatorios *RAND*, de 128 bits

b) Un paquete de datos *RES* (de 64 bits), que se usará para contrastar la respuesta del móvil

c) Una clave de confidencialidad *CK*, de 128 bits

d) Una clave de integridad *IK*, de 128 bits

e) Un "vale de autenticación" [*Authentication Token*] *AUTN*, que se compone a su vez de tres partes:

—Un número de secuencia *SQN* sumado ("xoreado" más bien) con una clave de autenticación *AK*, es decir, $SQN \oplus AK$ (tanto *SQN* como *AK* tienen 48 bits de longitud)

—Un campo para gestión de autenticación *AMF*, que contiene datos sobre el tipo de algoritmo usado, la clave y su período de vigencia (16 bits)

—Un código de autenticación del mensaje *MAC* (64 bits)

Todo eso es lo que el centro de conmutación tiene en su poder, y que necesitará para "negociar una sesión" con el móvil. Como ven ustedes, la cosa se complica, aunque podemos reconocer algunos de estos elementos. Nuevamente, hay una clave secreta que poseen solamente la red y el móvil. En el caso GSM, la llamábamos *Ki*, ahora será simplemente *K*. E igual que antes, ahora tampoco podemos transmitir *CK* por el aire, ya que cualquiera podría interceptarla. Lo que hemos de hacer es producir un conjunto de datos que permita a ambas partes identificarse a plena satisfacción, y acordar una clave de cifrado, todo ello sin revelar nada sobre *K*.

Para ello, necesitaremos un conjunto de algoritmos, llamados f1 a f5. En realidad,

son todos el mismo algoritmo, pero que nos dan datos diferentes según los necesitemos (como A3 y A8 en el caso GSM). Hay otro algoritmo, llamado f0, que produce el paquete aleatorio *RAND*. Los algoritmos f2 a f5 toman como entrada el paquete de datos *RAND* y la clave secreta *K*; f1 además toma como datos de entrada los paquetes *SQN* y *AMF* que hemos mencionado antes.

Comencemos entonces. El centro de conmutación (o la antena más próxima, si lo prefiere) envía al móvil el paquete aleatorio y el vale de autenticación, es decir, el paquete (*RAND, SQN⊕AK, AMF, MAC*). Ahora el móvil efectúa las siguientes operaciones:

—f5(*RAND, K*) = *AK*. Con este primer paso, recuperamos la clave de autenticación.

—(*SQN* ⊕ *AK*) ⊕ *AK* = *SQN* (recuerden que ⊕ aquí representa la operación XOR, que es su propia inversa). Con la clave *AK* del paso anterior, y el paquete de datos *SQN⊕AK*, recuperamos el número de secuencia *SQN*. El centro conmutador, en realidad, no ha recibido un quintuplete de datos, sino varios quintupletes, para que se vayan usando conforme sean necesarios. El número de secuencia *SQN* indica qué quintuplete se está utilizando en ese momento

—f1(*RAND, K, AMF, SQN*) = *XMAC*. A partir de los datos obtenidos en los dos pasos anteriores, y del paquete *AMF* enviado por el centro de conmutación, el móvil construye *XMAC*. Si coincide con el código *MAC* que tiene el centro de conmutación, es decir si *XMAC=MAC*, significa que los datos no han sido alterados, es decir, que tanto el móvil como el centro de conmutación están usando el mismo tipo de cifrado (*AMF*), el mismo quintuplete de autenticación (*SQN*), están realizando la autenticación en ese momento (*RAND*), y además comparten la misma clave secreta (*K*). De ese modo, la red ha quedado autenticada frente al móvil

—f2(*RAND, K*) = *XRES*. Si la clave *K* y el paquete aleatorio *RAND* son iguales, *XRES=RES* y el móvil queda autenticado frente a la red

En caso de que los pasos anteriores hayan sido cumplimentados satisfactoriamente, ambas partes quedan autenticadas, es decir, convencidas de que el otro es quien dice ser. Ahora pasamos al proceso de creación de las claves. Ese proceso es sencillo:

—f3(*RAND, K*) = *CK* nos da una clave temporal de cifrado, para proteger la comunicación

—f4(*RAND, K*) = *IK* proporciona un clave de integridad, para garantizar que los datos intercambiados no han sido alterados en ningún momento

Y por fin, la llamada comienza. El algoritmo para cifrar dicha llamada se denomina f8 (en la red GSM era el A5), y utilizará la clave *CK*. Como datos de entrada de f8 se tienen, además de la clave y del bloque que se va a cifrar, un conjunto de datos que indican la longitud y número del bloque de texto, la dirección

de transmisión y el portador. El tipo y función exactos de estos datos no son de importancia aquí. Baste saber que sirven para saber cómo es el bloque de datos que vamos a cifrar. Para descifrar, no hay más que invertir el proceso.

Una novedad en la red 3G respecto a la GSM es que no solamente aseguramos la confidencialidad de la comunicación, sino su integridad, de forma que nadie pueda inyectar paquetes cifrados falsos. Para ello se utiliza un algoritmo de integridad f9, que con su correspondiente clave *IK* y datos sobre el bloque de datos a cifrar nos da un código de autenticación de mensaje (MAC). Comparando el MAC que calcula el móvil con el que posee el centro de conmutación podemos saber si el mensaje ha sido transmitido de forma íntegra, o si por el contrario ha sido alterado.

He dejado cabos sueltos en esta explicación para no cansarle, querido lector. Por ejemplo, hay otras dos funciones, llamadas f1* y f5*, que son algoritmos de "resincronización", "y que no entraremos a detallar porque son esencialmente idénticos a los f1 y f5. Hay protocolos para diversas situaciones (cuando hay una asincronía de datos, cuando no se ha establecido una autenticación adecuadamente, cuando una petición de autenticación no es respondida…); e incluso existe la posibilidad de que los teléfonos GSM puedan usarse en la red 3G, lo que en la jerga de los programadores se denomina "compatibilidad hacia atrás"."

Vayamos a lo que aquí más nos interesa: ¿son buenos esos algoritmos? Afortunadamente, no tenemos que buscar copias filtradas, ya que son públicos y están disponible en Internet[74]. Diferenciemos entre los de autenticación y cifrado. Los primeros, esos que hemos llamado f1-f5, reciben el nombre genérico de MILENAGE. Como en el caso GSM, los creadores del estándar permiten que cada teleco utilice el algoritmo concreto que desee, e igual que antes se incorpora uno de serie, como ese disco de prueba que se regala con la cadena musical. En este caso, el "regalo" recibió el nombre de Rinjdael. Se trataba de un algoritmo de cifrado tan seguro que, después de un largo proceso de selección, fue seleccionado por el gobierno norteamericano para sustituir al viejo algoritmo de cifrado DES. Ahora Rinjdael se llama AES (*Advanced Encryption Standard*), y en cuestión de seguridad es difícil escoger algo mejor. En este punto, la elección del consorcio 3GPP merece un sobresaliente.

En lo que respecta al algoritmo dual de cifrado f8 e integridad f9 el proporcionado "de serie" por el 3GPP recibe el nombre de KASUMI. En este caso los responsables decidieron sabiamente no repetir los errores del pasado, al menos en lo relativo al secreto, y en septiembre de 2000[75] se anunció la distribución pública de las especificaciones de f8 y f9[76]. No solamente se utiliza en la telefonía 3G, sino que también es empleado en la telefonía GSM, como sustituto de A5/1, y también en el protocolo de datos GPRS, sustituyendo a GEA1 y GEA2. En esos casos, KASUMI recibe los nombres de A5/3 y GEA3, respectivamente[77]. La presentación en

sociedad del nuevo algoritmo fue hecha oficialmente en julio de 2002[78].

Por si se ha confundido con esta especie de trastorno de personalidad múltiple, se lo repetiré:

—En GSM, se llama A5/3

—En GPRS, recibe el nombre de GEA3

—En la telefonía UTMS de tercera generación (3G), es el conjunto de f8 y f9.

Cualquiera que sea su personalidad, se trata de un algoritmo de cifrado en bloque, con clave de entre 64 y 128 bits. Las especificaciones 3GPP indicaban que, cuando fuese usado en redes GSM y GPRS, operaría con claves de 64 bits, incluyendo la advertencia de que dicha clave sería "*no estructurada, no debe suponerse, por ejemplo, que ninguno de sus bits tenga valores predeterminadas*," es decir, que los tiempos de debilitar deliberadamente la clave con bits cero habían pasado a la historia. Cuando se utiliza en la telefonía 3G, procesa una clave más larga, de 128 bits[79].

Desafortunadamente, KASUMI no es la solución ideal. Como ya hemos visto, los fallos del protocolo GSM permite la existencia de diversas vías de ataque, fruto de las cuales el móvil puede ser forzado a utilizar cifrado débil, o incluso nada de cifrado. En este sentido, la adopción de KASUMI en su función de A5/3 no constituye realmente una mejora. Se supone, eso sí, que es más resistente que A5/1 desde el punto de vista criptoanalítico, aunque su clave de 64 bits lo hace vulnerable a ataques con tablas arcoíris. Idénticas consideraciones ha de hacerse en GPRS cuando KASUMI asume el papel de GEA3.

En cuanto a su uso en telefonía 3G UTMS, KASUMI resistió hasta 2005. Ese año, los investigadores Eli Biham, Orr Dunkelman y Nathan Keller (BDK) desvelaron una serie de ataques, el mejor de los cuales requería casi 2^55 mensajes en texto llano, y tenía una complejidad equivalente a la de efectuar 2^76 operaciones de cifrado. Esto, evidentemente, no es práctico. Sin embargo, hay que tener en cuenta que KASUMI tiene una clave de 128 bits, lo que significa que, de ser un algoritmo seguro, la única vía de ataque válido sería probar las 2^128 claves posibles. Lo que demostraron Biham, Dunkelman y Keller es que, en la práctica, la resistencia de KASUMI es equivalente a la de un algoritmo de cifra de 76 bits, y que resulta vulnerable a un tipo de técnica de ataque denominada criptoanálisis diferencial de clave relacionada[80].

Siguiendo la máxima de Bruce Schneier de que los ataques criptoanalíticos no pueden más que mejorar, en 2010 el equipo BDK hizo otro intento contra KASUMI[81]. El resultado fue contundente: las vulnerabilidades pasaron del "teórico pero no práctico" a "podemos hacerlo en el sótano de casa". Una técnica llamada "ataque sándwich" consiguió obtener los 128 bits de la clave. Los requisitos computacionales son tan bajos (un gigabyte de memoria) que el ataque puede llevarse

a cabo en apenas dos horas en un PC. La fortaleza de KASUMI quedó reducida al de un cifrado con clave de 32 bits (recordemos que el A5/2, usado para la exportación en la telefonía GSM, tenía clave de 40 bits).

Un descubrimiento adicional fue el hecho de que, unos años antes, el consorcio 3GPP metió la pata tontamente. El primer algoritmo de cifra e integridad que se consideró para la telefonía de tercera generación se basaba en un sistema llamado MISTY, diseñado por Mitsubishi. Con el doble objeto de evitar ciertos ataques criptoanalíticos y hacerlo más rápido en hardware, se efectuaron algunas modificaciones y se convirtió en KASUMI. Lo sorprendente fue que el ataque BDK de 2010 hacía picadillo la seguridad de KASUMI, ¡pero era ineficaz contra MISTY! En palabras de los autores del estudio: "*esto pone en cuestión tanto el diseño de KASUMI como la evaluación de seguridad frente a ataques de clave relacionada*". El cambio que buscaba una mejora acabó dando un algoritmo peor que el anterior. El único consuelo es que, debido a las características particulares del ataque, los autores no creen que sea aplicable a KASUMI tal y como se usa en la telefonía 3G; eso sí, no dijeron una palabra sobre su vulnerabilidad al usarlo en telefonía de segunda generación (GSM o GPRS).

Lo que resulta un enigma es el motivo por el que los expertos del consorcio 3GPP diseñaron un protocolo de autenticación e intercambio de claves basado en AES, probablemente el algoritmo criptográfico más resistente que pudieron encontrar, y luego confiaron el cifrado a un algoritmo recién creado. A estas alturas, y visto el ejemplo de otros casos del pasado, podríamos pensar que KASUMI fue deliberadamente debilitado para facilitar el acceso a las agencias de inteligencia nacionales. Quizá lo descubramos dentro de veinte años.

7) LA RESPUESTA: CÓMO PROTEGERNOS

Esta vez el apartado final sobre qué puede usted hacer para protegerse va a ser muy fácil de escribir. En dos palabras: no puede. El uso de la telefonía GSM, como hemos visto, es equivalente en términos de seguridad a abrir la ventana y llamar al vecino a gritos. El uso de interceptores de llamadas (*IMSI Catcher* o *Stingray*) por parte de particulares será moneda común en los próximos años; en la República Checa, se han convertido en prácticamente una epidemia[82]. En cuanto a los nuevos sistemas 3G, ya hemos visto que la seguridad es marginalmente mejor, pero está siendo erosionada rápidamente.

Por supuesto, para las agencias de inteligencia y los diversos servicios de policía y seguridad, pinchar es pan comido. Al margen de la potencia criptoanalítica a disposición de entidades como la NSA o nuestro CNI, lo cierto es que los móviles de tercera generación fueron diseñados con circuitos especiales para lo que se ha dado en llamar “acceso legal”. Una orden del juez, y todos los algoritmos criptográficos que supuestamente protegen nuestras comunicaciones quedan en nada. En España, las interceptaciones se llevan a cabo por medio del sistema SITEL, adquirido en octubre de 2001 a la empresa danesa ETI A/S por el Ministerio de Interior, entonces dirigido por Mariano Rajoy.

Lo cierto es que los gobiernos del mundo, pura y simplemente, nunca han estado por la labor de proporcionar o permitir comunicaciones telefónicas seguras para sus ciudadanos. Siempre ha imperado la razón de Estado y la justificación de la seguridad nacional para dejar abierto un canal de escucha. En teoría, solamente deberían utilizarse en casos controlados bajo control judicial; pero en una época en que la lucha antiterrorista alcanza niveles de paranoia en Estados Unidos, y en la que los policías españoles se niegan a llevar su placa de identificación, ¿realmente tenemos motivos para confiar en nuestros protectores?

Decida el lector cómo responder a esa pregunta. En cualquier caso, mi recomendación es: tenga cuidado con lo que cuenta por teléfono, y si no quiere que se sepa, no lo diga. Algunas opciones son más seguras que otras, pero no se confíe en exceso. Son malos tiempos para la lírica.

DE TODO UN POCO

1) ¿DÓNDE ESTÁ LA FUERZA 34?

El uso práctico de los sistemas criptográficos en el mundo real tropieza en la práctica con mil y un fallos. A lo largo de este libro hemos visto muchos. Es muy difícil esconder información a un enemigo ingenioso, y por eso la experiencia de uso sugiere nuevas formas de aplicar el cifrado.

Una de las más habituales consiste en rellenar el mensaje con caracteres nulos. Esta técnica, bien utilizada, complica la tarea al criptoanalista. Ya en el siglo XIV, el antipapa Clemente VII lo utilizaba desde su sede de Avignon. Su secretario, Gabriel de Lavinde, era consciente de que las técnicas criptoanalíticas de la época, aunque toscas para los cánones actuales, podía reveler el contenido de los mensajes cifrados, así que tuvo la idea de crear símbolos nulos (*nihil importantes*), que no significaban nada. La historia de la criptografía revela que, en la práctica, los secretarios rara vez utilizaban los nulos, ya que les complicaba la tarea de cifrar y descifrar, a pesar de que era un elemento de protección muy eficaz.

El relleno con símbolos nulos se aplicó con gran profusión durante el siglo XX. Habitualmente, los mensajes eran cifrados y enviados en bloques de cinco caracteres, en aplicación de los convenios internacionales de telegrafía. Si un mensaje tiene un número de caracteres que no es múltiplo de cinco, el último grupo de letras queda mutilado. La solución es, sencillamente, añadir algunas letras de relleno al final, una técnica conocida como “padding” (relleno). De ese modo, NECESITO REFUERZOS puede ser agrupado como NECES ITORE FUERZ OSXXX.

En ocasiones, el mensaje entero está compuesto de relleno sin sentido. Un enemigo hábil, simplemente tomando nota del número, longitud y procedencia de los mensajes, puede obtener información de gran valor sobre la situación e intenciones del adversario. En el pasado, el silencio por radio era en sí un mensaje cuyo significado estaba claro: se avecina una acción militar inminente. La solución es enviar multitud de mensajes falsos para engañar al enemigo; por supuesto, el cifrado es imprescindible, o de otro modo no habría engaño en absoluto.

Por supuesto, un mal cifrado puede dar al traste con un buen plan, incluso en el caso de mensajes con relleno. Durante la Segunda Guerra Mundial, los italianos enviaban mensajes falsos o “dummies,” que los ingleses intentaban por supuesto descifrar. Una criptoanalista llamada Mavis Batey se dio cuenta de que algunos de esos mensajes tenían una característica curiosa, a saber: contenían todas las letras del alfabeto excepto la L. Batey sabía que la Enigma tenía la propiedad de no reciprocidad, esto es, ninguna letra podía cifrarse como sí misma. ¿Acaso esos

mensajes sin sentido eran el resultado de cifrar largas ristras de LLLLL con la máquina Enigma? Lo probó, y efectivamente, así fue. Ese pequeño detalle evitó a los ingleses desperdiciar largas horas intentando descifrar mensajes; y lo que es mejor, les permitió en al menos una ocasión determinar el cableado de los nuevos rotores que los italianos introdujeron para sus máquinas cifradoras[1].

Hablando de los mensajes falsos, Batey dijo que "*hubiera sido exasperante pasarse días trabajando en un mensaje para que al final resultase ser un pasaje del Infierno de Dante*". Los franceses debieron sentir esa misma frustración siglos antes. En 1562, el embajador español en Francia, Tomás Perrenot de Chardonnay, envió un mensaje cifrado a Felipe II. En aquella época los franceses interceptaban y descifraban las comunicaciones españolas siempre que podían, así que el mensaje de Chardonnay fue prudentemente cifrado. Por algún motivo, la copia descifrada desapareció de los archivos españoles. El historiador Miguel Gómez del Campillo encontró el mensaje original en la década de 1950 y consiguió descifrarlo:

"No se rompan la cabeza / en descifrar esta carta
porque es cifra perdida / para engañar a los que / abren las cartas
Mira Nero de Tarpeya / a Roma cómo se ardía
por los bosques de Cartago / salían a montería
la Reina Dido y Eneas / con muy gran caballería
no faltaban caballeros / que los tienen compañía …"

Toda la carta era un mensaje "dummy," incluida la postdata: "*por eso no se rompan la cabeza en esto ni en lo demás, que será tiempo perdido y en balde*". Es evidente que el erudito embajador Chardonnay estaba al tanto no sólo de la interceptación y robos de mensajes diplomáticos, sino también de su descifrado por parte de criptoanalistas hábiles. No puedo sino imaginarme la cara de sorpresa que puso el ilustre señor historiador al leer el mensaje… por no hablar de los criptoanalistas franceses del siglo XVI[2].

Una de las aplicaciones del padding consiste en añadir palabras o grupos de letras al comienzo y al final del mensaje. Esto resulta especialmente útil en el ámbito de las comunicaciones militares y diplomáticas, donde los tratamientos protocolarios son muy rígidos. Términos como *excelencia, con respecto a mi telegrama anterior, con referencia a, beso las manos de VM o respetuosamente suyo*, usados una y otra vez al comienzo o al final de un mensaje, son campo abonado para un ataque criptoanalítico. La solución es añadir relleno tanto al comienzo como al final.

El uso de relleno en sus diversas variantes es una de las herramientas más eficaces para dificultad el trabajo de los criptoanalistas. Hay, eso sí, una condición imprescindible: el sentido del mensaje original debe ser preservado, y por tanto el relleno no debe inducir a error a los usuarios legítimos. Imagínense la confusión que

causaría un mensaje que diga REFUERZOS EN CAMINO al que se añade la palabra INNECESARIOS.

Incluso palabras de relleno aparentemente inocuas pueden dar un sentido diferente a un mensaje. Si usted está casado, ya habrá aprendido el cambio de significado que frases como "*¿has bajado la basura*?" tienen cuando se altera la entonación o se añade una inocente palabra. Los políticos echan mano de estas técnicas lingüísticas, y pueden fácilmente convertir un error aislado ("el gobierno se equivoca") en un reproche continuado ("de nuevo, el gobierno se equivoca"). Eso ocurrió en cierta ocasión durante la Segunda Guerra Mundial, provocando el desconcierto en una de las mayores operaciones jamás vistas. Y con esto comienza nuestra historia.

Viajemos atrás en el tiempo hasta octubre de 1944. El general MacArthur se dispone a cumplir su famosa promesa de volver a las Filipinas. Los japoneses, diezmados pero no vencidos, se disponen a impedirlo con todos los medios a su alcance. Para rechazar la inminente invasión, la armada japonesa se dividió en tres grupos. La Flota Norte (almirante Ozawa) se dirigió al norte de las Filipinas, en tanto que la Flota Sur (almirante Nishimura) rodeó el archipiélago por el sur. Su propósito era alejar a los grupos navales de Estados Unidos que protegían los flancos Norte/Este (almirante Halsey) y Sur/Oeste (vicealmirante Kinkaid). Mientras tanto, una poderosa tercera Flota japonesa (Centro), al mando del almirante Kurita, cruzaría las Filipinas por su zona central, saldría por el Estrecho de San Bernardino y tomaría por sorpresa a las fuerzas de desembarco.

La estrategia japonesa les salió muy cara, pero tuvo éxito. La Flota Sur fue destrozada por los acorazados norteamericanos, en tanto que la Flota Norte, perdidos sus majestuosos portaaviones en la batalla de Cabo Engaño (y no es broma, se llama así), huyó en dirección norte perseguida por los buques de Halsey. Pero la estrategia suicida de los japoneses tuvo éxito: la poderosa Flota Centro llegó a la zona de desembarco, y comenzó a atacar con dureza a la fuerza de invasión estadounidense.

El mando norteamericano, para empeorar la situación, estaba disperso. El almirante Kinkaid, en el sur, estaba a las órdenes de MacArtuhr; Halsey, en el norte, respondía ante el almirante Nimitz, en Hawaii. El propio MacArthur, en sus memorias, atribuiría los fallos de la operación a la división del mando y la falta de enlaces entre las flotas. Cualquiera que fuese el motivo, durante la noche del 24 de octubre un buque japonés tras otro, entre ellos seis enormes acorazados, desfilaron por el Estrecho de San Bernardino. Lo único que los norteamericanos tenían en las inmediaciones era la Séptima Flota, una débil fuerza de destructores y escoltas.

Imaginen ahora el dilema que se le planteó al almirante Halsey. No había sido informado de los problemas en San Bernardino por fallos en las comunicaciones. Las fuerzas navales bajo su mando habían hundido ya cuatro portaaviones de la Flota

Norte de Ozawa, y otros catorce buques estarían pronto al alcance de los cañones norteamericanos. Era una oportunidad de oro. Pero las llamadas de auxilio por parte de las débiles fuerzas que defendían las playas de desembarco eran cada vez más acuciantes. ¿Qué hacer? ¿Dar la vuelta para socorrer a los hombres que sus barcos debían estar protegiendo, aun a riesgo de llegar tarde? ¿Continuar la misión y terminar la destrucción de la Flota Norte?

Desde su puesto de mando a cinco mil kilómetros de distancia, el almirante Nimitz le envió un mensaje preguntando dónde se hallaba en esos momentos la fuerza naval que debía estar protegiendo la salida del estrecho de San Bernardino. Su nombre era *Task Force 34*, que se puede traducir como "fuerza de combate" o "fuerza de tareas", y englobaba la mayor parte de los acorazados y cruceros de Halsey. Nimitz se preguntaba dónde estaba la Fuerza 34 y por qué se había unido a la Tercera Flota, abandonando su papel de piquete frente a las playas de desembarco. En realidad, la *Task Force 34* sólo existía sobre el papel, ya que formaba parte integrante de la Tercera Flota, pero algunos (Nimitz entre ellos) pensaban en ella como una fuerza separada.

Y es aquí donde el gran error hace su aparición. Tres horas después de que las primeras llamadas de auxilio llegasen al almirante Halsey, Nimitz envió el siguiente mensaje, que incluyo aquí en su forma original:

TURKEY TROTS TO WATER GG FROM CINCPAC ACTION COM THIRD FLEET INFO COMINCH CTF SEVENTY-SEVEN X WHERE IS RPT WHERE IS TASK FORCE THIRTY FOUR RR THE WORLD WONDERS

Que puede traducirse aproximadamente así:

a) **Relleno inicial**: Pavo Trota al Agua
b) **Indicador de comienzo del mensaje**: GG
c) **Destinatarios**: de CINCPAC [Comandante en Jefe del Pacífico] Acción Com [para acción del Comandante] Tercera Flota Info [para Información de] COMINCH [Comandante en jefe de la Armada] CFT 77 [Fuerza Combinada de Combate 77]
d) **Separador**: X
e) **Mensaje:** Dónde está RPT [repito] dónde está la Fuerza de Combate 34
f) **Indicador de fin del mensaje:** XX
g) **Relleno final:** The World Wonders

Según cuenta David Kahn en su *Codebreakers*, el cifrador de Pearl Harbor introdujo una frase que "le saltó a la mente," en violación de la regla según la cual el relleno final debía ser totalmente ajeno al mensaje. Hay quien afirma que se trataba de una alusión al famoso poema de Alfred Tennyson *La carga de la brigada ligera.*

La acción que describió el poema de Tennyson se enmarcaba dentro de la batalla de Balaklava, cuyo 90 aniversario se cumplía el 25 de octubre, justo el día en que fue enviado el mensaje.

El problema viene de que *wonder,* como verbo, significa tanto "maravillarse" como "preguntarse," en función del contexto. En el poema de Tennyson, *the world wonders* puede traducirse como "el mundo entero se maravilla". Pero una traducción más literal puede convertirlo en "el mundo se pregunta" o, con algo más de estilo literario, "se pregunta todo el mundo". De modo que, cuando el mensaje llegó al destinatario, el descifrador eliminó correctamente el relleno inicial, pero decidió dejar el relleno final, temeroso de que tal vez fuese parte del mensaje. De esta forma, Halsey leyó: "*¿Dónde está, repito, dónde está la Fuerza de Combate 34? Se pregunta todo el mundo*".

Puedo imaginarme lo que sintió el almirante Halsey en ese momento. Se encontraba a punto de destruir a la Flota Norte japonesa, los mensajes de socorro procedentes de la débil flota que protegía las playas de invasión eran cada vez más angustiosos, la tensión se podía cortar con un cuchillo… y, de repente, recibe un mensaje con la coletilla "se pregunta todo el mundo". Halsey lo consideró como un insulto a su honor. Tenía al enemigo a punto de caramelo, ¡y alguien a medio mundo de distancia le reprochaba haber dejado su puesto de guardia! Según cuenta el propio almirante Halsey,

"*Me quedé de piedra, como si me hubieran abofeteado. El papel temblaba en mis manos. Me quité la gorra, la arrojé sobre la cubierta y grité algo que me avergüenza recordar… estaba tan furioso que no podía ni hablar*".

Humillado y furioso, dio la orden de girar al sur. La flota japonesa, a apenas sesenta kilómetros de los cañones norteamericanos, escapó del mismo destino que había convertido en chatarra la Flota Sur ¿Habría actuado Halsey igual de no haber tenido su juicio nublado por la ira? El era consciente de que su Tercera Flota llegaría demasiado tarde para socorrer a los soldados de las playas, pero su orgullo herido le impedía tomar otra decisión: dio la vuelta y dejó escapar al almirante Ozawa y los restos de su flota. Como se esperaba, llegó demasiado tarde para intervenir en la batalla entre la Séptima Flota norteamericana y la Flota Centro de Kurita. Afortunadamente para los norteamericanos, Kurita fue incapaz de aprovechar el éxito táctico que tenía en bandeja, y tras sufrir fuertes pérdidas se retiró por el estrecho de San Bernardino, sin que los buques de Halsey pudiesen impedírselo.

Quizá la mayor víctima del "fallo de relleno" fue el orgullo del almirante Halsey. En sus propias palabras: "*perdí la oportunidad con la que soñé desde mis días como cadete*"". Nada de duelos a cañonazos, nada de hundir una poderosa flota a la antigua usanza. Dos grandes acorazados convertidos y una docena de unidades de superficie (destructores y cruceros) consiguieron escapar de las garras de Halsey, cuya decisión

de marchar al sur le valió solamente el hundimiento de dos cruceros ligeros y un destructor enemigos.

La flota de Ozawa constituía una fuerza poderosa sobre el papel, pero casi inerme, ya que apenas disponía de combustible, municiones o aviación. La misión de Ozawa era alejar a Halsey de la Flota Centro todo el tiempo posible, y lo consiguió. A pesar de ello, la batalla del Golfo de Leyte constituyó una aplastante victoria norteamericana. En cuanto a la Flota Norte japonesa, su supervivencia tuvo escasa relevancia. Después de las batallas aeronavales de octubre de 1944, la flota japonesa perdió prácticamente cualquier posibilidad de efectuar operaciones a gran escala. Incluso con los restos de la Flota Norte, el destino del imperio japonés estaba sellado.

2) ACEITE DE SERPIENTE

Los sistemas de cifrado simétrico actual pueden dividirse en dos tipos: algoritmos de flujo (*stream ciphers*) y algoritmos de bloque (*block ciphers*). Los primeros manipulan el texto cifrado bit a bit, en tanto que los segundos trabajan con “bloques” de varios bits. Podemos visualizar un algoritmo de flujo como un taxi, que sale disparado en cuanto el viajero se sube en él. Por contra, el algoritmo de bloque sería como un autobús que no sale de la estación hasta que todos los asientos estén ocupados.

La mayoría de los algoritmos simétricos que nos suenan son de tipo bloque: DES, IDEA, AES. Tienen asignado un tamaño de clave, y un tamaño de bloque (algunos algoritmos tienen varios tamaños posibles de bloque). Por ejemplo, DES tiene una clave de 64 bits, o lo que es lo mismo, de 8 bytes. Esto significa que toma un mensaje M de 64 bits y lo cifra mediante una clave k para dar un bloque cifrado C de 64 bits de tamaño. En este caso, el tamaño de la clave y el del bloque coinciden, pero no es una condición que tenga que cumplirse necesariamente en otros algoritmos.

Matemáticamente, podemos representar el proceso de cifra como una función E que depende tanto del mensaje M como de la clave k: C=Ek(M). Como el algoritmo es simétrico, la función de descifrado es igual que la de cifrado, de forma que M=Ek(C). Sencillo hasta aquí. Pero ¿qué hacemos si el mensaje es mayor que un bloque? Lo lógico sería trocear el mensaje en bloques de, digamos, N Bits:

```
M = (M1,M2,M2… Mh)
```

A continuación, ciframos cada bloque independientemente:

```
C1=Ek(M1)
C2=Ek(M2)
   …
Ch=Ek(Mh)
```

El mensaje cifrado resultante es

```
C=(C1, C2, C3… Ch)
```

Este modo de cifrar bloques de bits es el denominado Modo de Libro de Código Electrónico o ECB (*Electronic Codebook Mode*). Como han visto, en el ECB cada bloque se cifra de modo independiente, como si los demás bloques no existiesen. Es el modo que, a priori, nos parece más natural. Resulta especialmente útil a la hora de cifrar grandes bases de datos, ya que no necesitamos cifrar todo de nuevo cuando añadimos, alteramos o borramos información. Es la forma más rápida de cifrar, y la verdad, uno al principio puede plantearse por qué pensar siquiera que hay otras

formas de cifrar. Por eso, ECB es el modo habitual en que muchos creadores de software incorporan sistemas de cifrado.

El problema con el modo de encadenado EBC es que resulta vulnerable a ciertos tipos de ataques. En este punto, el término “ataque” debe entenderse no en sentido estrecho (descifrar un mensaje) sino en el de dar a un intruso cualquier tipo de ventaja. Puede que un atacante no pueda leer un mensaje, pero si es capaz de insertar o sustituir parte del mensaje cifrado, eso también vale.

Para ilustrar la vulnerabilidad del modo ECB, vamos a imaginar una base de datos con la lista de clientes de un banco: nombre, número de cuenta, saldo. Supongamos que el criptoanalista tiene acceso a toda la base de datos cifrada (cosa que, a tenor de las noticias sobre robos de bases de datos, fallos de vulnerabilidad y pérdidas accidentales, resulta cada vez menos raro). Como toda la base de datos está cifrada con la misma clave, las redundancias que aparezcan pueden dar información sobre el contenido.

Por ejemplo, digamos que el atacante examina la base de datos, cifrada, y comprueba que hay diversas cadenas alfanuméricas que se repiten. ¿Puede tratarse de saldos en números redondos? Quizá *928hng4l* significa *diez mil*, o *kkwg972c* es un nombre de pila común. Puede que el enemigo haya averiguado que yo fui el único cliente del banco online que realizó una retirada de fondos entre las 9:56 y las 9:57 de la mañana, y de ese modo averiguar cuál fue la única línea de la base de datos que ha sido alterada; eso le dice que *9n1ff2ka* es el resultado de cifrar mi nombre.

He aquí el esquema de un ataque alternativo. Imaginemos que el Banco A envía un mensaje electrónico al Banco B, en el que se detalla una transferencia a un cliente suyo. Todo el paquete de datos está cifrado con la misma clave. Los datos de la transferencia están divididos en bloques: un bloque para el nombre del banco emisor, uno para el del banco receptor, tres para el nombre del destinatario, uno para la cantidad transferida y dos para el número de cuenta de destino: *ERDDDMCC*.

En esta ocasión, yo soy el atacante (qué quieren, me han vuelto a recortar el sueldo y hay que llegar a fin de mes como sea). En primer lugar, ordeno una transferencia legítima desde mi cuenta en el banco A a mi otra cuenta en el banco B, y grabo la transmisión *ERDDDMCC*. A continuación, intercepto otra transferencia, digamos *erdddmcc*, le quito los bloques correspondientes al banco de destino, nombre de destinatario y número de cuenta, los sustituyo por los míos, y envío el resultado *eRDDDmCC*. De ese modo, un desconocido a quien no tengo el gusto de conocer ha transferido sobre el papel (bueno, sobre el cable) una cantidad de dinero *m* a mi cuenta *CC*.

Fíjense que no conozco la clave, no sé quién es mi anónimo benefactor, y ni siquiera sé la cuantía de la transferencia. Tarde o temprano el banco receptor se dará cuenta de que el banco emisor no paga, se dirigirán al cliente y éste dirá que no es

cosa suya, pero para entonces yo ya he vaciado mi cuenta y me he largado a las Seychelles, desde donde les estoy escribiendo bajo el nombre supuesto de Arturo Quirantes.

Por supuesto, los ataques que les estoy describiendo aquí son elementales, y el sistema bancario tiene mecanismos para detectar y perseguir este tipo de fraudes, así que recuerde: no lo intente usted en casa. Aun así, estos ejemplos ilustran el hecho de que el cifrado no es una herramienta mágica que nos protege de todos los problemas. Puede ocultar información hasta cierto punto, pero no evita ataques de repetición o alteraciones de datos. Es necesario algo más.

Una de las soluciones es olvidarnos del cifrado en modo ECB. Si conseguimos que cada bloque cifrado dependa de otros bloques de texto cifrado, será más difícil engañar al sistema mediante trucos como los que acabo de mostrarles. Si se fijan ustedes, la construcción de paredes sigue un principio similar. A nadie se le ocurre limitarse a apilar ladrillos unos justo encima de los otros, formando torres de ladrillos que pueden fácilmente venirse abajo. En su lugar, los ladrillos se entrelazan unos con otros, lo que hace que la pared sea mucho más resistente.

Algo así es lo que sucede con el encadenado de bloques en un sistema de cifrado. Ahora, el resultado de cifrar el bloque de texto *Mn* dependerá de los bloques que hemos cifrado anteriormente. Es decir, la ruta del autobús en que viajo dependerá del recorrido que hayan efectuado los autobuses que le precedieron. El encadenado hace que el bloque cifrado *C[i]* dependa del bloque cifrado *C[i-1]*, que a su vez depende del *C[i-2]*, y así sucesivamente. Un problema que aparece es que un fallo en uno de los bloques cifrados (por una mala encriptación, fallos en el hardware, el software o la transmisión) puede propagarse a los demás bloques.

Se conocen diversos modos de encadenamiento, cada uno de los cuales tiene sus ventajas e inconvenientes. El primero es el llamado Encadenado de Bloque Cifrado, o CBC (*Cipher Block Chaining*). Funciona a base de tomar el bloque cifrado anterior, sumarlo al bloque de texto nuevo, y cifrar el resultado:

$$C[i] = Ek(M[i] \oplus C[i-1])$$

En este caso, la “suma” es la operación XOR (que hemos denotado con el símbolo ⊕), y que tiene la propiedad de que es su propia operación opuesta, es decir, la suma es igual que la resta. La operación de descifrado se haría de forma similar:

$$M[i] = Ek(C[i]) \oplus C[i-1]$$

Ya hemos mencionado algunas de sus ventajas. Como C[i] depende de C[i-1], no se puede sustituir un bloque cifrado por otro sin que el resultado pase inadvertido. Por desgracia, también tiene sus inconvenientes. Dos textos con el mismo comienzo darán como resultado el mismo texto cifrado hasta el punto en que comiencen las

diferencias. Este problema es muy habitual en archivos con idénticos encabezamientos (documentos de Word, cartas estereotipadas, mensajes de e-mail, etc), y nos recuerda que incluso en el mundo electrónico de hoy sigue siendo válida la máxima de evitar las regularidades. Que ya no se use la máquina Enigma no significa que comenzar los mensajes con “tengo el gusto de comunicarle…” sea una buena idea desde el punto de vista de la seguridad. Para evitarlo, una práctica habitual consiste en insertar al principio del texto una ristra de datos aleatorios de relleno llamada Vector de Inicialización (IV). El IV no significa nada, no aparece en el texto llano, y su única razón de ser es hacer que cada texto sea único.

Hay un precio a pagar en prestaciones. En el modo ECB, era posible paralelizar el proceso: si necesitamos acceder un centenar de datos en una base de acceso aleatorio, podemos irlos descifrarlos en paralelo. Pero el descifrado en modo CBC ha de hacerse en serie, operando con los bloques de datos de uno en uno. En cuanto a la propagación de errores, existe pero tiene poca trascendencia. Si en modo ECB un error en un bloque cifrado impedía leer un bloque en texto llano, el mismo error en modo CBC afecta solamente a dicho bloque y a un bit del bloque siguiente. Se dice que el modo CBC es autorrecuperativo.

Un problema común a los modos ECB y CBC es el de la sincronía. Si por error se introduce o se pierde un bit del conjunto de datos cifrados (por una transmisión errónea, por ejemplo), el texto llano que se obtiene resulta ilegible. Si no se introducen técnicas capaces de detectar y corregir variaciones en el texto cifrado, un sólo bit alterado nos convertirá el resto del texto cifrado en basura. Existe una variación del modo CBC en la cual el texto llano se puede ir cifrando en bloques más pequeños. Es decir, si el algoritmo de cifrado funciona en bloques de 128 bits, el modo CBC nos permite cifrar en bloques de 64 bits, o de 16, o incluso de uno sólo. Este modo, llamado Modo de Retroalimentación Cifrado o CFB (*Cipher-FeedBack*), es más complejo, pero elimina los errores de sincronización.

Otra variante, llamada Modo de Retroalimentación de Salida, OFB (*Output-FeedBack*), es algo más sencilla que la CFB. Tiene la ventaja que el error en un bit cifrado solamente afecta a un bit de texto llano —lo que cierra las puertas a algunos tipos de ataques—, pero también tiene errores de sincronización: un sólo bit añadido o eliminado en el mensaje cifrado, y estamos fritos.

Las recomendaciones generales para el uso de estos tipos de encadenado son:

—ECB. Es el más simple y rápido, pero también el más vulnerable. No se recomienda para cifrar mensajes.

—CBC. Es el mejor para cifrar archivos, ideal para aplicaciones basadas en software (como bases de datos).

—CFB. Es el modo “de rigueur” para cifrar flujos de datos transmitidos, cuando cada carácter ha de ser tratado individualmente, como los enlaces entre servidor y

terminal. Suele usarse en sistemas de alta velocidad donde no se puede permitir ninguna propagación de errores.

—OFB. Muy bueno en situaciones donde pueda haber errores, ya que no los propaga.

Visto lo visto, queda claro que el modo ECB es el menos seguro, ya que entre otras cosas permite notar patrones en el texto cifrado. Eso nos permite poder entender mejor un artículo criptográfico que apareció recientemente, y que ha provocado mucho revuelo. Una de las aplicaciones de ECB es el cifrado de volúmenes cifrados. Tales bichos son, sencillamente, grandes archivos cifrados que, al descifrarlos, se convierten en carpetas enteras (incluso unidades de disco virtuales) en nuestro disco duro. No es factible usar otros modos de encadenamiento en este caso, porque de hacerlo el sistema tendría que re-cifrar todo el volumen cada vez que yo modifique algún archivo de éste, convirtiendo así un volumen de acceso aleatorio en uno de acceso secuencial, mucho más lento de manejar.

Puesto que el modo EBC, como hemos visto, es vulnerable, los buenos fabricantes introducen añadidos tales como usar una clave que cambie para cada posición del volumen, o de algún modo hacer que el proceso de cifrado, aun con la misma clave, sea dependiente de la posición en el volumen cifrado. Sin embargo, un fabricante poco escrupuloso puede preferir usar el modo ECB por ser el más rápido y simple. El usuario, que no entiende de sutilezas criptográficas, tenderá a valorar más la velocidad de cifrado que la seguridad.

Ocasionalmente, un pretendido experto en seguridad “descubre” las vulnerabilidades del modo ECB con el oculto propósito de anunciar algún producto. Algo así sucedió en octubre de 2008, cuando C. B. Roellgen, de la empresa PMC Ciphers Inc, escribió un artículo con el título de “*Visualización de debilidades potenciales de implementaciones existentes de cifrado en software comercial para disco*”[3]. El autor muestra cómo una imagen cifrada en modo ECB permite adivinar contornos de la imagen original. Para ello, toma una fotografía de una chica guapa reducida a cuatro colores: blanco, negro y dos tonos de gris. Al cifrar con el algoritmo AES en modo ECB, se observa cómo algunos de los rasgos de la foto son aún visibles: el contorno de la figura de la chica, su cabello, brazo, cintura. Demasiado detalle para una imagen que supuestamente está bien protegida mediante cifrado.

¿Qué es lo que sucedió? Sencillamente, que al usar cifrado en modo ECB, las zonas de color uniforme se convierten en el mismo tono de “color cifrado”. Cada bloque de datos correspondiente a un solo color queda cifrado de la misma forma; los bloques que incluyan dos colores (por ejemplo, uno que contenga parte de la blusa y parte del pelo) se cifrarán de otra forma. El resultado es una imagen difuminada que todavía puede proporcionar información al ojo.

Es decir, nos están mostrando la obviedad de que bloques iguales de texto llano se convierten en bloques iguales de texto cifrado. Cuando se utiliza un modo ECB combinado con un parámetro dependiente de la posición de cada bit en la foto (algo similar a tomar una clave distinta para cada bloque de datos), la fotografía se convierte en un conjunto de puntos aleatorios sin patrones que podamos visualizar.

¿Por qué publicó este investigador lo evidente? En mi opinión, para asustar. El autor, dando un paso más, afirmaba en su artículo lo siguiente. Imaginemos dos volúmenes cifrados, el original (1) y una copia exacta (2). El volumen 1 se utiliza para guardar la fotografía, en tanto que el volumen 2 no contiene nada. Lo que dice este señor es que podemos "restar" ambos volúmenes (haciendo un XOR entre cada bit del volumen 1 y el bit correspondiente del volumen 2), y al hacerlo aparece el contorno de la fotografía más o menos reconocible.

No es difícil entender por qué. Si la fotografía sólo tiene cuatro colores, uno de ellos será el blanco, que representamos con ceros. En el volumen 1, por tanto, las partes de la fotografía en blanco se cifrarán de igual forma que los bits del volumen 2 que se encuentren en la misma posición. Y, al hacer un XOR, esas partes aparecen. En el artículo, lo sacan de color negro (restan en lugar de sumar), pero las partes que aparecen son las blancas: la cara de la chica y su camiseta. El resultado será igual si usamos ECB con una clave dependiente de la posición, porque estamos comparando bits en la misma posición en ambos volúmenes.

El propio artículo reconoce a las claras que "es la consecuencia lógica de cifrar información idéntica con idéntica clave". De hecho, esto es solamente posible porque el volumen cifrado 2 no fue creado, sino copiado. De haber creado el volumen 2 independientemente y haberlo dejado vacío, esto no hubiera sido posible, ya que cada volumen tiene un conjunto de datos iniciales llamado vector de inicialización (IV), que precisamente evita este tipo de ataques. Al copiar el volumen tal cual, también copió el IV.

Como pueden ver, el autor está mostrándonos lo que cualquier criptólogo mínimamente competente ya conoce. La información fue recogida en diversas webs especializadas[4]. Queda preguntarnos por el motivo. Es sencillo: el "artículo" (llamémosle así, aunque no hay constancia de que haya sido publicado nunca) fue, esencialmente, un ejercicio de publicidad enfocado a presentar un problema y, al final de todo, presentar una solución: un programa llamado TurboCrypt. Por cierto, el autor del artículo trabaja para una empresa llamada PMC Ciphers. ¿Se sorprenderán ustedes si les digo que esa misma empresa fabrica y vende TurboCrypt?

La verdad es que todo el tema de PMC y TurboCrypt resulta hilarante. Cuando leí el artículo por primera vez, me sonó un poco raro. Reconozco que me perdí en el artículo, en parte por el rollo de código fuente que incluyen (innecesariamente, creo yo), en parte por la sensación de estar leyendo cómo inventaban la rueda. Decidí

hacerle una consulta al criptógrafo y experto en seguridad Schneier, y su respuesta fue tajante: “Sólo están notando patrones cuando alguien cifra en modo EBC. Nada nuevo… y usar ECB es una bobada”. Al día siguiente, publicó el asunto en su blog[5], donde lo calificaba de "intento descarado de atraerse publicidad". Ya en 2003, Bruce habló de PMC en estos términos:

"*PMC Ciphers. La descripción de la teoría está tan llena de pseudo-criptografía que es divertida de leer. Las hipótesis se presentan como conclusiones. La investigación actual se especifica mal o se ignora. El primer enlace es un artículo técnico con cuatro referencias, tres de ellas escritas antes de 1975. ¿Quién necesita treinta años de investigación criptográfica cuando tienes teoría de cifrado polimórfico?*"

¿Cifrado polimórfico? TurboCrypt huele profundamente a “aceite de serpiente” (*snake oil*). Se trata de un divertido término que utilizan los criptólogos para referirse a productos de cifrado que, con palabras rimbombantes y exageradas afirmaciones de robustez, ocultan un interior débil y vulnerable. La expresión proviene del Lejano Oeste, donde ocasionalmente llegaba el típico buhonero en su carreta. A falta de farmacias, el viajante presentaba a los aldeanos su preparado milagroso que curaba todo tipo de dolencias, achaques y enfermedades. Por supuesto, su eficacia era nula, pero para cuando los incautos compradores se daban cuenta nuestro vendedor de aceite de serpiente ya se encontraba muy lejos.

Algo parecido parece suceder con TurboCrypt. Nada más abrir la web correspondiente, nos recibe una pelirroja muy sugerentemente escotada (la misma del artículo) junto con la lapidaria afirmación de “*La herramienta definitiva… ninguna agencia gubernamental de este mundo podrá jamás reventar TurboCrypt*”[6]. Hasta aquí, nada que no podamos achacar a un exceso de entusiasmo por parte del equipo de relaciones públicas.

Lo sorprendente viene cuando comenzamos a leer las interioridades de su Cifrado Polimórfico (PolyMorphic Cipher, o PMC). Algunos detalles son para echarse a reír, o cuando menos, a uno se le queda cara de bobo. El primero es la afirmación de que el cifrado polimórfico era un secreto de estado en Alemania hasta 1999, y que no existe ataque posible contra este tipo de sistemas. A continuación pasan a decir que su cifrado polimórfico, de 1024 bits, es tan estupendo que uno puede usar en su lugar AES “*si el usuario tolera una seguridad menor*”. Según ellos, TurboCrypt puede usarse con AES o con PMC, y la empresa afirma que los usuarios prefieren PMC en un 80% frente a AES; una afirmación tan arrogante como asegurar que el coche que usted quiere venderme resiste cualquier golpe, pero que si lo deseo puedo conformarse con las menores prestaciones de un carro de combate Leopard 2.

En cuanto al algoritmo en sí, no se detallan sus características, pero la información que proporcionan es de lo más extraña: según dicen, el propio algoritmo

de cifrado es variable. Se supone que, de entrada, el modo de cifrar está sin definir, y el algoritmo en sí viene determinado por la clave. Según los creadores de PMC, "*la suposición clave para un criptoanálisis con éxito es el conocimiento detallado del algoritmo de cifrado… pero el algoritmo del Método Polinómico es DESCONOCIDO*". Si hemos de creer a lo que nos dicen, parte de la clave se utilizar para compilar el código máquina del algoritmo en sí. El resultado final es un flujo de datos pseudoaleatorio, que al sumarlos (xor) con el texto llano nos da el texto cifrado, algo así como lo que hacen los principales algoritmos de cifra en flujo.

Todo esto es, en mi humilde opinión, un sinsentido sin pies ni cabeza tan grande que no sé ni por dónde empezar. En primer lugar, las técnicas de ingeniería inversa permiten obtener el modo de funcionamiento de un algoritmo, aun cuando éste sea secreto. William Friedman consiguió reproducir el funcionamiento de la máquina japonesa de cifrar Púrpura, e incluso consiguió criptoanalizarla sin éxito, a pesar de no haberla visto nunca. El criptoanálisis no se basa en conocer el algoritmo (aunque indudablemente, ayuda), sino en descubrir vulnerabilidades matemáticas, correlaciones no siempre obvias entre texto llano y texto cifrado. Los trabajadores del ramo se basan en los Principios de Kerchhoffs, enunciado en el siglo XIX, uno de los cuales afirma que la seguridad del sistema ha de recaer tan sólo en la clave. Cualquier proceso cuya seguridad se base en mantener detalles del algoritmo en secreto acabará siendo vulnerado.

En segundo lugar, si la clave de entrada es aleatoria, es de suponer que el algoritmo también lo será. Pero precisamente los detalles del algoritmo son clave a la hora de determinar si un sistema de cifra es seguro o no. Cojamos un algoritmo de cifra, como por ejemplo AES, y destripémoslo. Cada permutación, cada tabla, cada inversión, cada proceso están cuidadosamente diseñados para evitar el criptoanálisis. Incluso pequeños cambios, aparentemente inofensivos, pueden dar al traste con la seguridad del algoritmo. Si el diablo está en los detalles, los algoritmos de cifra deben de ser su lugar de vacaciones favorito. Afirmar que un código informático basado en bloques aleatorios produzca la misma seguridad que un algoritmo clásico es como intentar diseñar una ciudad a base de montar piezas de Lego a ciegas.

Incluso si todo ello fuese cierto, el proceso de compilación del código máquina sería lento (y diferente para cada clave) y difícil de asegurar contra mirones. Es posible que las agencias de inteligencia utilicen procedimientos similares, pero de hacerlo les habrá costado mucho esfuerzo, y quedaría por ver qué ventaja les proporcionaría frente a los algoritmos probados conocidos en la actualidad. La industria civil ciertamente no los utiliza, o habríamos visto ejemplos de otras empresas.

Finalmente, las referencias del tipo "hay más claves posibles que átomos en nuestro planeta" sugieren poca seriedad. Una clave de 128 bits o más proporciona

seguridad frente a cualquier ataque de fuerza bruta. Si a estas alturas tienen que convencer al usuario de la formidable ventaja de seguridad que obtendrán utilizando claves (de cifrado simétrico) de 1024 bits, deberían mejorar sus técnicas de venta. Que es probablemente lo que hicieron con el artículo de Roellgen anteriormente mencionado.

En la web de TurboCrypt se incluye un desafío criptográfico que le permitirá a usted, lector, ganar diez mil dólares si consigue romper su sistema de cifra[7]; lleva el atrevido título de "*Intente hacerlo mejor que Bruce Schneier. Rompa el Cifrado Polimórfico*". A los creadores del desafió no se les escapó el hecho de que Schneier es una especie de Chuck Norris de la seguridad informática, así que publicaron una carta para Schneier, en el que le desafiaban formalmente a atacar su sistema PMC, y tomaron su ausencia de respuesta como una especie de victoria[8].

Por supuesto, si yo desafiara a Chuck Norris y no obtuviera respuesta, eso no significa que me tenga miedo. Lo más probable es que él tenga mejores cosas que hacer que viajar hasta España para partirme la cara. Algo así pienso que le habrá pasado a Schneier, cuyo único comentario público al respecto fue "*tiene gracia*".

Frente a la indiferencia de Schneier, y probablemente en medio de un coro de risas por parte de la comunidad criptográfica, PMC Ciphers llegó a afirmar que la NSA usa su sistema de cifra, así que ha de ser bueno. Llegaron a apoyar los algoritmos propietarios (esto es, secretos) y a recelar incluso del algoritmo de cifra AES. ¿El motivo? La NSA tiene algoritmos propietarios, así que han de ser buenos. Y aluden incluso a las fuerzas del mercado:

"*¿Cuántos individuos y empresas van a invertir el dinero, tiempo y esfuerzo necesario para desarrollar nueva tecnología de cifrado, si luego lo regalan? La verdad pura y dura es que el capitalismo, el potencial para hacer beneficios, es una de las mayores fuerzas que impulsan el desarrollo de nuevas tecnologías. Eliminar esta fuerza del desarrollo del cifrado sólo nos hará más débiles a largo plazo*"."

A la vista de los fracasos obtenidos por los sistemas de cifra creados bajo los auspicios de la seguridad mediante oscuridad, creo que la afirmación de PMC cae por su propio peso. Los mejores sistemas de cifra han sido siempre los creados a la luz del día, con código abierto y revisable por cualquiera. La única ventaja efectiva de TurboCrypt es esta: representa un buen ejemplo de aceite de serpiente.

3) BÚSQUEDAS RAZONABLES

En todas las películas y series norteamericanas donde aparecen policías o abogados, una de las cosas que aparecen son las órdenes de registro. Llamadas genéricamente “warrants”, autorizan a la policía a registrar lugares, personas u objetos en busca de pruebas. Los seguidores de CSI están habituados a ver órdenes para ver zapatos, registrar dobladillos de pantalones o examinar ruedas de repuesto de automóviles.

El engorro en especificar qué se quiere registrar, cómo y por qué motivo, evita que los abogados defensores puedan invalidar una búsqueda por ser demasiado amplia. Uno de los principios de la ley norteamericana consiste en exigir lo que se necesita para la investigación, y nada más. Viene impuesto nada menos que por la Constitución de los Estados Unidos, cuya Cuarta Enmienda protege a los ciudadanos contra registros de tipo arbitrario (*unreasonable search*): “*No se emitirán órdenes [warrants] salvo mediante causa probable… describiendo particularmente el lugar que será registrado, y las personas o cosas que serán registradas*”.

Indudablemente, la Cuarta Enmienda se convierte en una limitación al trabajo de fiscales y policías, pero asimismo se convierte en una herramienta poderosa para la protección de los derechos y libertades individuales. Eso hace que, conforme avanza la tecnología y se complican las relaciones entre personas, se entablen batallas jurídicas apasionantes. ¿Una empresa, como entidad jurídica, es igual que una persona física a efectos de privacidad? ¿Se puede registrar la basura que ha tirado una persona? ¿Le pertenecen a una persona las huellas dactilares del vaso que ha tocado? ¿Se puede registrar una casa desde el exterior mediante medios no intrusivos, como examinar la radiación infrarroja (calor) que desprende, o con un radar?

La frontera es más difícil de establecer de lo que parece indicar el sentido común. Hay muchos procesos automáticos que difícilmente pueden considerarse registros o búsquedas; pero, por otro lado, ¿un registro se define por sus métodos, por sus resultados, o por el efecto obtenido? ¿Es registro una acción llevada a cabo fuera de la propiedad de una persona? Hasta 1967, los pinchazos telefónicos en los Estados Unidos se consideraban legales si no se invadía el domicilio del abonado; tuvo que ser el Tribunal Supremo el que dictaminara que una interceptación telefónica sí era un registro, y por tanto requería una orden judicial.

Puede uno imaginarse ejemplos actuales. Supongamos que usamos una cámara termográfica para medir el calor generado por una casa. Si lo hacemos desde el exterior, puede argumentarse que lo único que hacemos es recoger radiación infrarroja emitida por su dueño sin limitaciones. Pero si eso resulta una evidencia jurídica en un proceso judicial, la cosa se complica. Mi defendido, diría la defensa, estaba en su casa, nunca consintió que su calor fuese detectado, no fue informado de

que se recogería en la calle, y no tenía medios materiales para evitar que el calor se filtrase al exterior. Intenten ustedes resolver el intríngulis. Creo recordar que el Tribunal Supremo invalidó dicha búsqueda por violar la Cuarta Enmienda. Pero luego dijo que las fotografías que envía una persona para revelar pueden ser registradas por la policía sin necesidad de orden.

Por supuesto, este libro no se ha convertido en un boletín de información jurídica. Si he sacado el tema a colación es precisamente porque la frontera gris de la Cuarta Enmienda ha tocado un elemento que ya hemos tratado desde el punto de vista técnico: las funciones hash.

Un hash es una función que toma un archivo o texto y lo convierte en un "destilado" formado por unos cuantos bits. Es una forma cómoda de representar dicho archivo. Estamos acostumbrados a ver funciones hash dentro del esquema de la firma digital, ya que dicha firma no es más que el resultado de tomar un archivo, someterlo a una función hash y cifrar dicho hash con nuestra firma privada. Pero también sirve para identificar archivos. Si busco una imagen en Internet, me resultaría muy difícil. Digamos que quiero conocer el origen de una fotografía que una vez me pasaron. ¿Cómo describirla para que Google me la encuentre? Difícil, si se trata de una descripción verbal. Pero si en lugar de decir "una foto que vi una vez, que tenía flores blancas sobre un fondo de hierba, y un pequeño escarabajo a la derecha" indicamos el valor de su hash, el sistema puede buscar la foto cuyo valor de hash coincida con la nuestra. El hash vendría a ser algo así como nuestro número de DNI o de pasaporte.

Pueden imaginarse el valor que esto tiene para los que persiguen intercambios de música sin pagar, o sencillamente fotografías de contenido pederasta. Y aquí conectamos con el caso que nos ocupan. Recientemente, un tribunal norteamericano tuvo que dictaminar sobre si tomar un hash de un archivo constituye una búsqueda razonable o no. El caso tiene bastantes flecos interesantes, pero permítanme que nos centremos tan sólo en la parte criptográfica.

Antes, los antecedentes. El acusado, al que llamaremos Alan (los nombres reales son públicos, pero no quiero contribuir a airearlos), estaba de alquiler, y según parece no pagaba la renta, así que su casero Charlie contrata a Manny para que vacíe la vivienda. Manny, entre otras cosas, encuentra el ordenador de Alan y se lo da a su amigo Peter. Éste se dedicó a trastear en el ordenador y, entre otras cosas, descubrió dos videos con pornografía sexual explícita entre menores. Asustado, borró los vídeos, y unos días después se dirigió a la policía y les entregó el ordenador.

En este punto, la historia ya está bastante liada. Hay que tener en cuenta si el casero estuvo dentro de la ley al coger las posesiones del alquilado, quien por su parte denunció el hurto de su ordenador. En cualquier caso, el detective Grissom (lo siento, no he podido resistirme a usar ese alias) toma el ordenador y lleva a cabo un análisis

forense. Los pasos que siguió pueden ser de mucho interés para aquellos que se han preguntado cómo se puede, en un caso de este tipo, demostrar que la policía no ha alterado datos.

Lo primero que hizo Grissom es calcular un valor hash (usando el algoritmo MD5) de todo el disco duro. De este modo, más tarde se podrá detectar si ha sido cambiado siquiera un bit del disco. Dicho cálculo se efectuó mediante un programa en modo de sólo lectura (para evitar escrituras accidentales en el disco). A continuación, hizo un barrido antivirus, y después de ello creó una imagen, es decir, una copia exacta del disco duro. Usando los datos de la imagen, Grissom se dedicó a calcular valores hash de todos los archivos del disco duro, y más tarde los comparó con los valores hash existentes en una base de datos del *National Center for Missing and Exploited Children*. De ese modo, se puede verificar si algún archivo del disco duro es sospechoso de ser pornográfico sin siquiera ver dicho archivo. Según la denuncia, Grissom obtuvo 171 videos sospechosos, que tras una inspección ocular mostraron múltiples imágenes de pornografía infantil. Finalmente, examinó los registros del ordenador en busca de información sobre páginas web visitadas.

El lector debe fijarse en el uso de los valores hash como identificadores de archivo. El investigador no accedió directamente a los archivos sospechosos hasta que obtuvo indicios razonables basados en la comparación de valores hash con los de otros archivos. Hasta que obtuvo ese indicio, nadie visualizó los videos, y ni siquiera se había alterado un solo bit de la propiedad del acusado. Por otro lado, esos datos podían ser usados para incriminarles, y el proceso de obtención de las pruebas fue cuando menos cuestionable. Por ello, una de las solicitudes de la defensa fue la supresión del ordenador como prueba. Argumentaban violación de la Cuarta Enmienda.

En la resolución judicial, se indicaba que, en efecto, un registro sin orden judicial es inaceptable, salvo casos muy concretos. Ello no obstante, se supone que cuando no hay expectativas razonables de privacidad la Cuarta Enmienda no te protege. Es decir, si un oficial de policía efectúa una búsqueda que no comprometa la privacidad del interesado, dicha búsqueda no está afectada por la Enmienda. Por poner un ejemplo tonto, si una persona está hablando en público por teléfono a grandes gritos y cualquier persona escucharle a varios metros de distancia, no debe quejarse de violación de la privacidad.

La acusación decía que se daba el caso de “no expectativa razonable de la privacidad”. Peter ya había accedido al ordenador de Alan, y el posterior registro policial del ordenador se llevó a cabo en condiciones mucho más restrictivas (doctrina de registro privado, en el que no se aplica la Enmienda). De hecho, Grissom ni siquiera “accedió al ordenador”, sino que se limitó a calcular valores hash. Resulta algo cuestionable, a menos de que por "acceso" quiera referirse a exámenes

audiovisuales o alteraciones de datos. Sólo tras comparar los hashes con valores de videos pornográficos ya conocidos resultó legal el registro (examen visual) del contenido del ordenador.

Por contra, la defensa argumentaba que el hecho de obtener valores hash constituyó un registro más intrusivo que el examen visual de Peter, y que la protección de la Cuarta Enmienda debería aplicarse sobre sus intereses de privacidad en el ordenador. El tribunal estimó adecuado que Peter se chivase a la policía, ya que la interpretación legal de la Cuarta Enmienda permite que un tercero a quien el acusado haya comunicado datos se los pase a la policía. Según esto, si los investigadores policiales no registran el ordenador de forma más invasiva que Peter, los resultados obtenidos no se consideran registro irrazonable.

Este es un punto importante: ¿fue el examen de Grissom más profundo, o menos, que el de Peter? O dicho en otras palabras: ¿es más intrusivo un examen visual de un video, o una compilación de valores hash? La acusación afirmó que, puesto que los agentes no miraron ningún archivo, no hubo registro. El tribunal rechaza esta argumentación y decreta que la obtención de hashes sin orden judicial fue una violación de la Cuarta Enmienda.

"*Para obtener los valores hash del ordenador de [Antonio], el gobierno [la parte acusadora] retiró físicamente el disco duro del ordenador… o aplicó el programa EnCase a cada compartimento, disco, archivo, carpeta y bit. Al someter al todo el ordenador a un análisis de valores hash, cada archivo, historia de internet, fotografía y "lista de colegas" se hicieron disponibles para revisión por parte del Gobierno. Un examen tal constituye un registro*".

El tribunal razonó que la búsqueda de Peter fue muy diferente que la de Grissom, y por tanto no es aplicable la doctrina de "Peter lo hizo primero". Que Alan perdiese las expectativas de privacidad con respecto a los dos videos visualizados por Peter no significaba que las perdiese con respecto al resto de sus archivos. La búsqueda con valores hash no cambiaba la cuestión, ya que permanecía en pie el hecho de que los investigadores de la acusación obtuvieron con ella mucha más información que la proporcionada por Peter.

Con todo, el modo de argumentar es algo extraño. Se citó un precedente, según el cual no es legítimo examinar todos los disquetes o CDs de la casa de un sospechoso sólo porque en uno de ellos un particular hubiese encontrado información sospechosa. El argumento de la acusación de que el disco duro es un sólo disco no era aplicable, según el juez porque un disco duro está compuesto de diversas placas metálicas denominadas "platters" que aunque funcionalmente funcionasen como un solo disco podían considerarse como "contenedores de datos" individuales, y por tanto, entidades físicas que podían considerarse como si fuesen disquetes separados.

Incluso el examen visual de los videos sospechosos, efectuado después de la

comparación de valores hash, se consideraba según el juez protegido por la Cuarta Enmienda, y por lo tanto necesitado de orden judicial. Su conclusión fue tajante: "*Los funcionarios policiales, sin exigencia y sin autorización, llevaron a cabo una investigación no limitada, sin orden judicial del ordenador de una persona, a pesar del hecho de que una orden podía haberse obtenido con facilidad*"."

De todos modos, Alan no se las vio muy felices. Aunque no se pudiese usar la evidencia obtenida del su ordenador, la policía también hizo sus deberes a la antigua usanza e interrogó al sospechoso en su casa; Alan terminó confesando que fue él quien introdujo esos videos en su ordenador. Su abogado intentó invalidar la confesión, pero el juez estimó que la hizo de modo voluntario y sin coerción.

La decisión de que los datos obtenidos del ordenador no sean admitidos como prueba va más allá del interés en situaciones similares. Como método de investigación legal, el análisis mediante comparación de valores hash se considera en iguales términos que el examen visual. Esto me parece adecuado, ya que permitiría obtener pruebas, sin orden judicial, que podrían condenar a un acusado o cuando menos influir fuertemente en un proceso. El cálculo de valores hash no puede apelar a su carácter no intrusivo para eximirse de protecciones tipo Cuarta Enmienda. Un valor hash puede ser usado para determinar el carácter legal o ilegal de un archivo (no de forma perfecta, pero puede ayudar en ocasiones), y por tanto un acusado debería estar protegido mediante las mismas salvaguardas legales que se otorgan a registros domiciliarios o interceptaciones telefónicas.

Por supuesto, lo que sucede en un país no es de aplicación inmediata en otro, y una sentencia del Tribunal Constitucional español nos lo recuerda. A finales de 2007, una persona envió a reparar la grabadora de discos de su ordenador, y cuando el técnico abrió un par de archivos para comprobar el correcto funcionamiento de las piezas se encontró con una gran cantidad de archivos pornográficos de adolescentes. El técnico denunció el caso a la policía, quien procedió al registro del ordenador. En mayo de 2008, la Audiencia Provincial de Sevilla halló culpable al dueño del ordenador de un delito de corrupción de menores.

La particularidad en este caso radicó en que la policía no se molestó en pedir una orden judicial para registrar el contenido del ordenador. Los motivos aducido por el juez para permitir tal cosa fueron dos. El primero, que al entregar el ordenador, el dueño indicó que no tenía contraseña de acceso, sin establecerle limitación en el uso o acceso de ficheros:

"*En consecuencia, pese a conocer que el técnico accedería al disco duro del ordenador (pues para ello le solicitó la contraseña), el acusado consintió en ello sin objetar nada ni realizar ninguna otra prevención o reserva que permita concluir que pretendía mantener al margen del conocimiento ajeno determinada información, datos o archivos*"

Y el segundo, que el causado tenía configurado el programa de intercambio p2p eMule, lo que ponía archivos a disposición de cualquier usuario, y por tanto:

"*difícilmente puede invocarse el derecho a la intimidad cuando los propios actos del acusado indican paladinamente que no tenía intención ni voluntad alguna de preservar para su esfera íntima, exclusiva y personal ninguno de los ficheros que conservaba en su ordenador, pues a ellos tenía acceso cualquier persona que se conectara en Internet a la misma red de intercambio*".

El Tribunal Supremo confirmó la sentencia, considerando que con sus acciones el acusado "*no había dispuesto un ámbito de privacidad respecto al contenido pornográfico infantil del ordenador*" El acusado recurrió ante el Constitucional por entender que la policía debió haber obtenido autorización del juez, que no había motivos de urgencia que legitimaran una actuación inmediata, que no había entregado el ordenador más que para que le reparasen la grabadora, y porque el uso de eMule se descubrió solamente tras el registro.

El Tribunal Constitucional, aun reconociendo el derecho a la protección constitucional de los datos del acusado (sean del tipo que sean), deniega el recurso en virtud a que la policía estaba persiguiendo un delito grave, y en consecuencia las medidas tomadas fueron proporcionadas y necesarias. Respecto a por qué no esperaron a obtener una orden judicial, el dictamen del alto tribunal, en mi humilde opinión, peca de simplista, o bien de ignorante en lo que a la tecnología actual se refiere[9]:

"*… adquiere especial relevancia en este caso la función que se encomienda a la Policía Judicial de asegurar las pruebas incriminatorias… en este supuesto, hay que tener en cuenta que la persona denunciada no estaba detenida cuando se practica la intervención, por lo que tampoco aparece como irrazonable intentar evitar la eventualidad de que mediante una conexión a distancia desde otra ubicación se procediese al borrado de los ficheros ilícitos de ese ordenador o que pudiera tener en la «nube» de Internet*".

¿Por qué digo que el TC muestra aquí ignorancia tecnológica? Pues porque hay que ser bastante ignorante para creer que una persona puede borrar datos de un ordenador en poder de la policía que está desconectado de Internet y además puede apagarse en cualquier momento. No lo digo yo sólo. Una magistrada del tribunal emitió un voto particular en el que discrepaba del fallo de sus colegas:

"*no alcanzo a entender por qué, estando el ordenador físicamente en poder de la Policía, las diligencias de investigación no podían esperar a que su realización contara con autorización judicial… el acceso a archivos de Internet (como los que incriminaban al recurrente) sólo puede realizarse si el terminal en cuestión está conectado a la red, por lo que en nada se hubiera puesto en riesgo la labor investigadora de la Policía si, estando dicho terminal en su poder, se mantiene*

apagado hasta lograr la preceptiva autorización judicial… no concurriendo la urgente necesidad de realizar esa intervención de manera inmediata, la misma debió llevarse a cabo con la autorización previa y el control de su ejecución por parte de la autoridad judicial, ya que durante el tiempo necesario para su obtención no existía riesgo de destrucción de las pruebas incriminatorias, ni se ponía en peligro la investigación policial".

Sí, es un delito asqueroso, todos lo sabemos. Pero las cosas se hacen bien o no se hacen. En este caso, el TC ha mantenido la condena a un personaje asqueroso, pero a cambio ha creado jurisprudencia a favor de irrumpir en la esfera privada de un ciudadano sin tutela judicial alguna. Una de dos: o el máximo tribunal español ha escogido tapar una falta de procedimiento por parte de la policía antes que anular una condena por pornografía infantil; o bien, sencillamente, no tienen ni idea de cómo funciona el siglo XXI. No sé cuál de las dos alternativas me da más miedo.

4) CRIPTOGRAFÍA NIKON

Dentro de un grupo de personas que comparten una afición, a veces se da una especie de polarización en torno a dos marcas o equipos, ambos muy buenos y con prestaciones similares. Los astrónomos aficionados llevan décadas divididos entre las marcas de telescopios Celestron y Meade. Los conductores dudan entre BMW y Mercedes, o bien entre Ferrari y Porsche. La ciencia ficción está polarizada entre los universos de Star Wars y Star Trek. En el campo de la informática, las opciones predominantes son PC y Mac (con permiso de los amigos de Linux). Y los entendidos en fotografía se debaten entre Canon y Nikon.

Lo cierto es que ambas marcas producen excelentes modelos de cámaras fotográficas, tanto para aficionados como para profesionales. Por supuesto, todo experto sabe que el elemento más relevante para hacer una buena fotografía es el propio fotógrafo, pero los aficionados que se dan de expertos, en lugar de hablar sobre exposiciones, encuadre y enfoque, se pasan interminables horas enzarzados en discusiones bizantinas sobre qué cámara tiene mejor calidad de imagen.

Un elemento fundamental en esta discusión es el formato en que se toman las fotos. Seguro que usted tiene una cámara fotográfica (la del móvil también vale), y habrá comprobado que habitualmente los archivos de imagen tienen formatos como TIFF, JPG o PNG. Tales formatos permiten una gran compresión de la información, y son por ello muy populares en Internet. Pero la compresión tiene un precio: se pierde información. Por dicho motivo, los profesionales utilizan los llamados formatos “crudos” (*raw*), de mucho mayor tamaño pero que conservan toda la información original de la fotografía.

Para desgracia tanto de aficionados como de profesionales, no existe un formato universalmente aceptado para las fotografías en formato crudo. Cada fabricante de cámaras tiene el suyo, y suele ser un formato “propietario,” protegido por patentes. Los fabricantes se ven aquí atrapados en un dilema. Por una parte, les conviene que ningún fabricante de software para procesar imágenes pueda usar su formato, ya que así pueden vender el programa propio; pero si el formato es demasiado cerrado, los usuarios podrían verse tentados a escoger otro tipo de cámara.

El formato usado por las cámaras Nikon se denomina NEF (*Nikon Electronic Format*). Como otros parecidos, incluye tanto la información de la propia fotografía (básicamente, la intensidad de los píxeles) como los llamados **metadatos**, que proporcionan información asociada a la fotografía. Los metadatos pueden incluir las condiciones de iluminación, la velocidad ISO, el uso de flash, la distancia focal, la apertura, incluso la situación geográfica si la cámara dispone de dispositivo GPS.

En 2003, Nikon introdujo la cámara profesional D2H, y a finales de 2004 su sustituta, la D2X. En el segmento de cámaras de consumo, la estrella era la D50,

presentada a finales de 2005. Todas ellas llevaban incorporada una nueva versión del formato NEF, distinto de los anteriores en un punto crucial: parte de la información estaba cifrada. Un programa cualquiera podría leer y procesar ese formato, pero para modificar el balance de blancos hacía falta un sistema de descifrado que solamente se encontraba en el programa de tratamiento de imágenes de Nikon (Nikon Capture, valorado entonces en $100).

La noticia se conoció en abril de 2005, cuando Thomas Knoll, co-creador del famoso programa de retoque fotográfico PhotoShop, acusó a Nikon de cifrar la información relativa al balance de blancos en sus nuevas cámaras[10]. El problema era muy grave, ya que entraba de lleno en un punto muy delicado: ¿a quién corresponde el control último de una fotografía, al fotógrafo o al creador del software que la modifica? Muchos temieron que Nikon comenzara a ejercer sus prerrogativas como propietario del formato para acabar diciendo a los usuarios cómo debían usarlo y con qué software se les permitiría modificar sus fotos.

Nikon tuvo que emitir una nota de prensa en la que intentó explicar sus motivos. La empresa proporcionaría un equipo de desarrollo de software o SDK (*Software Development Kit*) para trabajar con el formato NEF, incluido el balance de blancos. Pero dejaba bien claro que el SDK solamente se ponía a disposición de los desarrolladores de software autorizados, y la autorización es un proceso que controla Nikon. En lo que respecta al usuario final, ni una palabra[11].

Es decir, Nikon controla quién tiene permiso para hacer programas de software que puedan alterar el balance de blancos. Cualquier desarrollador "no autorizado" que intentase romper el sistema de cifrado podría verse ante una demanda judicial. A tenor de lo visto, romper el cifrado en cuestión era técnicamente viable, y además fácil. En pocos días, la empresa Bibble Labs anunció que su programa de tratamiento de imágenes Bibble ya llevaba soporte para la Nikon D2X[12]. Simultáneamente, Dave Coffin modificó su programa Dccraw con el mismo fin, e último incluso publicó el código fuente de su programa[13].

El propio Thomas Knoll advirtió que, aunque el cifrado fuese fácil de romper, "*Nikon podría considerar una ruptura del cifrado del balance de blancos como una violación de [la ley] DMCA [Digital Millenium Copyright Act], y demandar a Adobe [propietaria de PhotoShop]*" No sabemos cómo fueron las negociaciones entre Adobe y Nikon, pero sí el resultado: hubo acuerdo. Adobe consiguió desarrollar un complemento (*plug-in*) que, entre otros, era compatible con los modelos D2H, D2X y D50 de Nikon[14]; y en septiembre de 2005 un comunicado conjunto de ambas empresas, cuidadosamente redactado, informó que ambas empresas trabajaban conjuntamente para "*asegurar que nuestros clientes comunes tengan una experiencia excelente cuando usen cámaras Nikon con software Adobe*"[15].

No hay mucha información sobre el sistema de cifrado de blancos de NEF, pero

los SDK contienen los datos necesarios, y a pesar de los acuerdos de no revelación que se firman en estos caso, inevitablemente se han acabado filtrando los detalles en diversos foros especializados. Según el fotógrafo Thom Hogan[16], las claves usadas para cifrar serían una combinación del número de serie de la cámara y el número de la fotografía. Análisis posteriores determinaron el modo de cifrado: sencillamente, se trata de una tabla de valores (*look-up table*) prefijada que, para cada valor de entrada, proporciona un valor de salida. En este caso, el cifrado es de lo más arcaico, y sabiendo dónde se encuentra esa tabla, el secreto está revelado.

Lo extraño de todo este asunto es que la información cifrada por Nikon es escasa y poco relevante. Un fotógrafo profesional puede trabajar sin la información exacta sobre el balance de blancos, y como hemos visto el descifrado es fácil de descubrir. Hubo en su momento una gran polémica sobre si ese sería el primer paso para tomar el control de las fotografías de los usuarios, disponiendo qué se puede hacer con ellas y qué software habría que usar. Sin embargo, un episodio más reciente puede arrojar algo de luz al respecto.

El lector quizá recordará haber visto una fotografía del presidente George Bush, leyendo un cuento infantil a un grupo de niños durante los atentados del 11-S. En algunas copias de dicha fotografía, el cuento aparecía boca abajo. En realidad, se trataba de una manipulación informática poco caritativa hacia el entonces inquilino de la Casa Blanca. Saber si una fotografía ha sido alterada es algo muy útil en campos como la fotografía profesional o la prensa, y resulta esencial a la hora de presentar imágenes como evidencias forenses.

En junio de 2006, Nikon anunció el lanzamiento de un programa para autenticación de imágenes (IAS, *Image Authentication Software*), diseñado para la cámara D2X[17]. El sistema firma digitalmente la fotografía, y el resultado se guarda en una sección del archivo de metadatos en la que habitualmente el fotógrafo puede incluir información sobre… el balance de colores. Es, por tanto, plausible que en 2005 Nikon pretendiese tan sólo experimentar con una versión rudimentaria de firmado digital. Quizá no quisiese hacerlo público en su momento para no dar pistas a sus competidores.

En cualquier caso, el sistema IAS de Nikon es interesante. Nos resultará muy útil como ejemplo, porque nos permitirá exponer dos elementos criptográficos muy útiles en el proceso de firma digital: las funciones hash y la criptografía de clave pública. Una firma digital consiste, en esencia, en cifrar un archivo. Tomamos el algoritmo de cifrado, utilizamos nuestra clave y ya tenemos la firma. El problema es que, si un tercero quiere verificar la firma, tendremos que darle la clave que hemos usado, y nada le impedirá usarla para firmar otros archivos.

Para evitar este problema, nada mejor que la criptografía de clave pública (PKC) o criptografía asimétrica. Este sistema, descubierto en los años setenta, utiliza dos

claves complementarias. Una de ellas se mantiene en secreto (clave privada), en tanto que la otra puede diseminarse libremente (clave pública). La clave pública sirve para que cualquiera pueda cifrar un mensaje, pero solamente quien controle la clave privada podrá descifrarlo. También puede utilizarse en forma inversa para la firma digital: la clave privada puede utilizarse para firmar un archivo, y cualquiera puede utilizar la clave pública para verificar dicha firma.

La PKC tiene un problema: los archivos cifrados son de un tamaño enorme, miles de veces mayores que el original. Por eso, y por su lentitud, es mejor tomar el mensaje, condensarlo en un pequeño archivo, cifrar dicho condensado con la clave pública, y el resultado es la firma digital. Eso requiere la invención de una función de condensación o destilado, lo que se denomina "función hash".

Supongamos que *M* es nuestro mensaje, *H* es la función hash y *h=H(M)* es el hash del mensaje. Para que la función hash *H* sea criptográficamente útil, ha de cumplir las siguientes condiciones:

1) Comodidad. Calcular *H(M)* ha de ser un proceso fácil y rápido, y no requerir mucha memoria o capacidad de cálculo.

2) Confidencialidad (o resistencia de preimagen). La función hash no ha de revelar nada de información sobre el mensaje. Por ello, dado un hash *h* dado, ha de ser imposible encontrar un mensaje *M* que cumpla que *h=H(M)*.

3) Resistencia (débil) a las colisiones (o resistencia de segunda preimagen). Dado un mensaje *M* determinado, ha de resultar inviable encontrar otro mensaje *M'* que tenga el mismo valor hash. Esto es, no se debe cumplir que *H(M)=H(M')*

4) Resistencia a las colisiones ("resistencia de cumpleaños"). Ha de ser muy difícil encontrar dos mensajes que den el mismo valor de hash. Esta propiedad se parece a la anterior, pero en este caso *M* no es un mensaje determinado, sino un mensaje cualquiera.

Encontrar una función hash que cumpla esas condiciones es difícil. Idealmente, un cambio de un solo bit en el mensaje debería conllevar la alteración de al menos la mitad de los bits del hash. Las funciones hash, en cierto modo, representan al mensaje, lo que las hace útiles en diversas situaciones como firmas digitales, justo lo que estamos examinando aquí.

El resultado de salida de una función hash ha de ser una cadena con un número de bits lo bastante largo. Una función cuyo hash tuviese solamente dos bits no sería útil, ya que al haber solamente cuatro posibles valores (00, 01, 10, 11) es poco resistente a las colisiones. Un ejemplo es nuestro Documento Nacional de Identidad, que consta de ocho números y una letra. La letra no es independiente, sino que depende del número. Para obtener la letra, basta dividir el número del DNI por 23; el resto se hace corresponder con una letra de acuerdo con la secuencia siguiente:

```
0 1 2 3 4 5 6 7 8 9 10 11 12
```

```
T R W A G M Y F P D  X  B  N

13 14 15 16 17 18 19 20 21 22
 J  Z  S  Q  V  H  L  C  K  E
```

Lo que en un principio se conoció como “la letra del NIF” fue introducido para poder detectar rápidamente si alguien estaba inventándose un DNI sobre la marcha. La letra del DNI no es una función hash válida, ya que solamente tiene 23 posibles valores, lo que le hace altamente vulnerable a colisiones. En realidad, nunca se diseñó como función hash sino como una especie de código de detección de errores.

Una función hash de *n* bits tiene un total de *2^n* (dos elevado a *n*) posibles valores. Eso hace que para altos valores de *n*, una buena función hash tenga resistencia débil a las colisiones (propiedad 3), ya que la probabilidad de encontrar un segundo mensaje con el mismo valor hash sería de uno entre *2^n*. Eso hace que, en principio, valgan valores de *n* del orden de 128 para proporcionar seguridad contra las colisiones débiles. Sin embargo, la resistencia a las colisiones (propiedad 4) es más difícil de conseguir, ya que no estamos hablando de un mensaje determinado, sino de un texto cualquiera.

Para entenderlo, digamos que estoy en una sala con una sola persona. ¿Cuál es la probabilidad de que esa persona cumpla años el mismo día que yo? Si obviamos las soluciones extrañas (nada de gemelos, y queda prohibido cumplir años el 29 de febrero), esa probabilidad es de una entre 365, lo que podemos poner como 1/365 o bien como (1-364/365). Si hay *n* personas, puede demostrarse que la probabilidad viene dada como 1-(364/365)^*n*. Para que esa probabilidad supere el 50%, ha de haber un número de personas *n* superior a 253.

Sin embargo, supongamos que nos interesa la probabilidad de que dos personas cumplan años el mismo día. Parece lo mismo, pero no lo es, porque ahora el día en cuestión no es necesariamente el de mi cumpleaños, sino el de cualquiera. Ahora, la probabilidad se calcula de modo distinto: P=1–365!/[(365-n)!*365^n]. Para una probabilidad del 50%, el número *n* es solamente 23. Este número parece demasiado pequeño, pero es correcto. A este resultado se le llama **paradoja de cumpleaños**.

En realidad, ambos casos esconden una igualdad: para alcanzar una probabilidad del 50%, hemos de comprobar un total de 253 parejas. La diferencia es que, en el primer caso, las parejas eran del tipo “una persona cualquiera y yo mismo,” en tanto que en el segundo caso tenemos parejas tipo “una persona cualquiera con otra persona cualquiera”.

La propiedad 3 de una función hash (resistencia débil) es similar al primer ejemplo de cumpleaños: tenemos un mensaje determinado, y buscamos un segundo mensaje; la probabilidad de encontrarlo es de una entre 2^n. La propiedad 4 (resistencia), sin embargo, es similar al segundo ejemplo, porque ahora tenemos dos

mensajes cualesquiera. En tal caso, la probabilidad de encontrar dos mensajes con igual valor de hash es de una entre *2^(n/2)*.

Eso significa que las funciones hash de 128 son seguras contra las colisiones débiles, pero no contra las normales, porque estas últimas suceden una vez cada 2^64, y ese número, aunque grande en términos absolutos, está al alcance de los recursos informáticos de un atacante, o de grupos de computación distribuida. Por dicho motivo, las funciones hash con 128 bits de salida se consideran obsoletas, y han sido sustituidas por otras como SHA-1, de 160 bits. Esto nos daría una probabilidad de resistencia a las colisiones de 1 entre 2^80. Eso ya es un número muy alto, aunque no absolutamente fuera de límites. Por ese y por otros motivos, una variante llamada SHA-2 tiene una salida de entre 224 y 512 bits.

Lo anterior es un límite a la fortaleza, y por tanto a la utilidad, de las funciones hash. Eso presupone que dichas funciones son criptográficamente sólidas, de modo que nadie puede "hacer trampa" y obtener colisiones con menos esfuerzo. En esto, las funciones hash comparten con los algoritmos de cifrado rasgos comunes: son difíciles de obtener. Por ejemplo, la función MD4 fue diseñada en 1990 por Ronald Rivest, toda una autoridad en la materia, a pesar de lo cual se descubrió una serie de ataques a partir de 1995. El autor creó un segundo algoritmo hash, llamado MD5, que aguantó algo más, pero finalmente sucumbió ante diversos ataques. En la actualidad, las mejores técnicas permiten obtener colisiones en MD5 con una dificultad de 2^21. Es decir, la resistencia frente a las "colisiones de cumpleaños" es análoga a la de un algoritmo hash perfecto de 42 bits; teniendo en cuenta que MD5 tiene salida de 128 bits, está claro que algo va mal.

En 2004 se puso en marcha el proyecto MD5CRK para encontrar colisiones débiles en MD5, cosa que se logró en pocos meses después[18]. Lo que en teoría debió requerir 2^128 cálculos de hash se consiguió en apenas 2^64, casi como una colisión de cumpleaños. Poco después, un grupo de investigadores consiguió obtener una pareja de certificados X.509 (usados en muchas aplicaciones de criptografía de clave pública) que utilizaban dos claves distintas pero tenían el mismo valor hash MD5[19], lo que demostraba que el uso de esa función en certificados digitales era insegura.

Los ataques contra MD5 fueron mejorando en potencia y rapidez. Y, lo que es peor, en seriedad. En la actualidad, los diseñadores de productos criptográficos tienden a huir de MD5. Sin embargo, hasta hace bien poco algunas aplicaciones importantes permitían el uso de esa función hash, con consecuencias potenciales catastróficas.

Veamos dos de ellas. La primera concierne al protocolo SSL, utilizado en la navegación web segura. Este sistema de seguridad actúa en las páginas de bancos y tiendas online, y utiliza un sistema de certificados digitales. Por supuesto, tener un

certificado digital no significa nada en sí mismo, del igual modo que una firma manuscrita no es más que un garabato en un papel. Para que sean aceptados, los certificados digitales suelen ir firmados por lo que se denominan Autoridades de Certificación (AC), que actúan como una especie de notarios digitales.

En 2008, un grupo de investigadores de Suiza, Holanda y Estados Unidos consiguió falsificar el certificado digital de una Autoridad de Certificación, algo así como duplicar el sello del notario[20]. Para conseguirlo, montaron un superordenador casero consistente en una bancada de 200 consolas Playstation 3. En palabras de uno de los autores, "*podríamos hacernos pasar por Amazon.com y usted no se daría cuenta*"[21]. En efecto, el atacante podría montar una web falsa y asegurarla con un certificado firmado por una AC falsa. Todo parecería normal: la cabecera *https*, el candado cerrado en una esquina del navegador.

El ataque suizo pudo llevarse a cabo porque una autoridad de certificación llamada AC RapidSSL todavía utilizaba MD5 en sus certificados, a pesar de que se conocían los fallos de esa función hash. Casi de inmediato, el propietario de RapidSSL, el gigante Verisign, anunció la retirada de los certificados MD5 y anunció que dejaría de usarlos en el futuro. A pesar de ello, en agosto de 2012 —cuatro años después del ataque— todavía quedaban 1300 páginas web que seguían utilizando certificados digitales basados en MD5, una lista en la que hay bancos, servicios de correo webmail, universidades e incluso un vendedor de certificados SSL[22].

El segundo ejemplo tuvo repercusiones políticas internacionales. Entre 2010 y 2012, los expertos en seguridad informática descubrieron un conjunto de programas maliciosos (*malware*) de una sofisticación sin precedentes. Al contrario que los virus y troyanos habituales, que atacan indiscriminadamente, estos nuevos elementos de software estaban diseñados para espiar o dañar sistemas de control industrial, con especial atención a los países de Oriente Próximo. La prensa norteamericana apuntó como responsables de este cibersabotaje a los servicios de inteligencia de Israel y Estados Unidos, con el fin aparente de frenar el programa nuclear iraní[23].

Uno de estos códigos maliciosos, llamado Flame, se caracteriza por el modo en que se transmitió de un sistema a otro sin ser detectado por ningún sistema antivirus. Al parecer, sus diseñadores se aprovecharon de una vulnerabilidad de MD5 llamada "colisión de prefijo escogido," descubierta en 2009 por Marc Stevens, Arjen Lenstra y Benne de Weger[24]. Con ello consiguieron falsificar un certificado de Microsoft, con el que firmaban el código de Flame para posteriormente enviarlo de forma que pareciese una actualización de Windows. Por supuesto, el sistema operativo comprobaba la firma digital, pero la falsificación daba el pego y Flame entraba sin ser detectado.

La sofisticación del ataque quedó patente (si es que había dudas a esas alturas) cuando los descubridores del ataque por colisión de prefijo escogido anunciaron que

Flame utilizaba una colisión similar pero en una variante distinta, desconocida hasta entonces para la comunidad criptográfica. Marc Stevens afirmó que el diseño de esta nueva variante precisó de "criptoanálisis de talla mundial"[25]. No se trata de un par de hackers aburridos, o de un grupo de investigación universitario. Estamos hablando de actores nacionales.

Microsoft, socio involuntario en este ataque sofisticado, dio la alerta en un aviso de seguridad que incluía pocos detalles[26]. Posteriormente proporcionó más información, incluyendo referencias a Flame[27], y procedió rápidamente a revocar los certificados digitales en cuestión[28], así como a reforzar el sistema de actualizaciones para evitar, o cuando menos dificultar, ataques similares en el futuro[29]. En un intento para tranquilizar a sus clientes, advertía que "*es importante que mantenga usted su PC al día con las últimas actualizaciones para que funcione bien y con seguridad*;" un aviso que, habida cuenta de cómo se propagó Flame, no deja de tener su punto de ironía.

Volvamos al sistema de autenticación de imágenes de Nikon. A juzgar por cómo le fue con el ensayo de 2005, no cabría esperar mucha diligencia por su parte. Por el contrario, la empresa de fotografía hizo bien los deberes. La PKI que escogió se basa en el algoritmo RSA, con clave de 1024 bits. Aunque la mayoría de los expertos consideran pequeño dicho tamaño, y se decantan por claves de 2048 bits[30], lo cierto es que nadie ha conseguido "romper" una clave RSA de 1024 bits; aun cuando sea factible, la potencia de cálculo necesaria sería asombrosamente alta.

En cuanto a la función hash, Nikon utilizó el llamado SHA-1 (*Secure Hash Algorithm*). Se trata de un algoritmo usado, y recomendado, por el gobierno federal de Estados Unidos, y que se utiliza para aplicaciones civiles en todo el mundo. El algoritmo ha mostrado recientemente cierta vulnerabilidad a la resistencia frente a colisiones (la propiedad 4 de antes). Aun así, se sigue considerando a SHA-1 como una de las mejores funciones hash existentes hoy día.

En conclusión, Nikon escogió buenas funciones para su firma digital. Cada imagen digital es procesada mediante el hash SHA-1, dando como resultado un valor de 160 bits de longitud, que al ser sometido al algoritmo RSA produce una firma digital de 128 bits. La misma operación se lleva a cabo una segunda vez, para firmar el conjunto de metadatos de la imagen. Ambas firmas se guardan en una zona de la sección de los propios metadatos.

Un proceso impecable, al menos sobre el papel, pero tiene un punto débil. Para que la cámara pueda firmar las fotografías, en sus entrañas electrónicas ha de guardarse una copia de la clave privada RSA, y un atacante hábil podría obtenerla examinando cuidadosamente el firmware interno de la cámara. Esto es lo que hizo la empresa Elcomsoft en abril de 2011. No dio muchos detalles técnicos, más allá de una vaga referencia a que "*la clave criptográfica secreta se maneja de forma*

incorrecta, y puede ser extraída de la cámara" y una lista de las cámaras Nikon afectadas[31], pero sugirió que la vulnerabilidad era similar a una que habían hallado un año antes en cámaras Canon. En un toque de fino humor, Elcomsoft presentó fotografías firmadas "correctamente" según los estándares de Canon, y que incluían incongruencias tales como un astronauta levantando una bandera soviética en la Luna o una imagen de Stalin sosteniendo en alto un iPhone[32].

En ambos casos, el fallo no estaba en el sistema criptográfico en sí, sino en la forma en que se llevaba a cabo dentro de la cámara. Es algo así como mirar por encima del hombro a alguien que esté abriendo una caja fuerte: por muy fuerte que ésta sea, si podemos ver la combinación ya está todo hecho. En el caso de Canon (y puede que también de Nikon), la clave es la misma para todas las cámaras de un modelo dado, lo que aumenta el problema, ya que basta con obtener la clave de una sola cámara para que todas las demás del mismo modelo queden comprometidas.

En ninguno de ambos casos se han reconocido dichos ataques. Una nota de prensa de Nikon, fechada cinco meses después, se limita a afirmar que no ha hecho anuncio oficial alguno[33]; en cuanto a Canon, no hay respuesta oficial hasta la fecha. Ahora los fans de ambas marcas tienen un tema más de controversia: ¿cuál de las dos empresas será la primera en reconocer el fallo?

5) SONY Y LAS CLAVES DE LA PLAYSTATION 3

Uno de los puntos de fricción más activos en la eterna pugna entre usuarios contra fabricantes compete a las videoconsolas. Los grandes fabricantes parecen desear que los usuarios se limiten a comprar el aparato, comprar los juegos y callarse. Sin embargo, una Wii, una Xbox360 o una PS3 tienen tanta potencia de cómputo como un ordenador. Y hay por ahí mucha gente con conocimientos técnicos, que piensan que usar una consola solamente para jugar es un desperdicio. Puede sonar raro a algunos que alguien quiera instalar Linux en una consola de juegos, ya que no están diseñadas para eso. Pero si puede hacerse, ¿por qué no hacerlo?

La PlayStation3 ha sido la videoconsola más resistente a “usos no autorizados”, pero incluso ella sucumbió, y llegó el momento en que se le pudieron introducir chips modificados para poder realizar operaciones prohibidas, como por ejemplo jugar con juegos piratas o instalar Linux en ellas. El problema para los fabricantes es que, al menos en España, eso no es legal. La ley prohíbe la fabricación, venta o uso de medios técnicos diseñados específicamente para neutralizar protecciones informáticas, pero el énfasis está en la palabra “específicamente”. Los jueces han dictaminado en repetidas ocasiones que usar un dispositivo que tenga también otros fines legítimos es legal. Recomiendo a los interesados en el aspecto legal la excelente web de Bufet Almeida[34]. Lean las últimas sentencias sobre el asunto.

En el fondo, se trata del viejo asunto de quién controla la consola: ¿el fabricante o el usuario? El primero la vende para ganar pasta, y pretende que el segundo la use de forma limitada y controlada; pero el usuario no es tonto, y piensa que si la consola es suya, debería poder hacer con ella lo que quiera. Y no es solamente una tontería de cuatro frikis con demasiado tiempo libre. Una consola de juegos moderna es un ordenador por derecho propio, y si los usuarios pudiesen usarlo como ordenador, el propio mercado informático podría cambiar sustancialmente.

Ya en 2000, la prensa digital afirmó que Sadam Hussein había comprado hasta 4000 consolas PlayStation 2, con posibles aplicaciones militares como controlar aviones no tripulados cargados de armas químicas, calcular trayectorias para misiles balísticos o incluso diseñar armas nucleares[35]. Nunca se encontraron esas máquinas en Irak, así que o bien las consolas nunca llegaron a su destino, o bien la noticia era un bulo desde el principio; o acabaron en el mismo lugar que las misteriosas armas de destrucción masiva de Sadam.

Pero la idea era factible, y se aplicó en la siguiente generación de consolas de Sony.En 2007, un equipo de la Universidad de Massachusetts instaló una versión del sistema operativo Linux en ocho máquinas PlayStation 3, creando con ello un “cluster” con aplicaciones en astrofísica computacional a un precio muy inferior al de un superordenador de potencia equivalente[36]. La Fuerza Aérea de EEUU llevó esta

idea al límite, y compró nada menos que 1760 máquinas PS3. El resultado fue el Condor Cluster, un sistema con una velocidad de cálculo de 500 teraflops diseñado para procesar la información que envían los sistemas de vigilancia aérea.

Las posibilidades solamente estaban limitadas por la imaginación "… y por Sony. En este punto, y antes de continuar, es interesante resaltar que la PlayStation 3 tuvo en un principio la capacidad de ejecutar Linux y otros sistemas operativos desde su disco duro. La propia Sony se aprovechó de esta capacidad y la usó como argumento de ventas desde 2000, cuando la PS2 era su videoconsola estrella. Se convertía así en la reina de las consolas, un instrumento tan potente que podía ser usado como ordenador de gran capacidad.

En algún momento la historia de amor entre Sony y la comunidad hacker se rompió. Probablemente todo comenzó cuando Sony insertó programas *rootkit* en sus CDs musicales para intentar atajar la piratería allá por 2005 (los lectores encontrarán información sobre el tema en el capítulo "Descifrando a Nemo"). Un *rootkit* es un programa que tiene amplios privilegios para hacer prácticamente cualquier cosa en un ordenador, y colárselo a los usuarios sin autorización ni preaviso es el tipo de cosas por el que la gente va a la cárcel en algunos países; no es, ciertamente, la conducta que se espera de un caballero, ni mucho menos lo que corresponde a una empresa que pretenda pasar por ejemplo "cool" frente a una comunidad de tecnófilos.

La PS3 fue durante años objetivo de todo tipo de hackers. Tres elementos contribuyeron a ello: era una máquina potente, tenía reputación de ser más segura que sus rivales, y podía albergar sistemas operativos como Linux. Toda una tentación. Un hacker llamado George Hotz ("Geohot") cantó victoria a comienzos de 2010: había hackeado la PS3 para lograr acceso a la memoria de lectura y escritura. Con esa hazaña, se abría la puerta a programar la consola a voluntad, pudiendo ejecutar software hecho por los propios usuarios. Como opción adicional, se podrían ejecutar juegos pirateados, o incluso los juegos de la antigua PS2[37].

La acción de Hotz probablemente fue la gota que colmó el vaso. Sony decidió que había llegado el momento de acabar con tanto uso no autorizado. A partir de la revisión 3.21 del firmware (1 de abril de 2010), las PlayStation3 dejarían de ser capaces de ejecutar Linux. Los usuarios que no instalasen esta nueva actualización perderían, entre otras cosas, la posibilidad de usar el modo multijugador de PlayStation Network[38]; algo que Sony se olvidó de advertir a sus clientes españoles, por cierto[39].

Se acabó eso de usar consolas de juegos para hacer simulaciones informáticas, ejecutar Linux o cualquier otra cosa que no fuese jugar y callar. Eso fue un duro golpe para muchos proyectos de supercomputación. Los sistemas ya existentes, como el Condor Cluster de la USAF, podían seguir en funcionamiento, a condición de no volver a actualizar el firmware de las máquinas. Para los nuevos, se acabó la

posibilidad de utilizar las PS3 como una alternativa barata a un gran ordenador de cálculo científico.

Esta acción fue vista por muchos como una traición por parte de Sony y, a partir de ese momento, se les rompió el amor. Los amantes despechados se lanzaron sobre la PS3 para reclamar la posibilidad de instalar Linux y volver a convertirla esta máquina en un aparato interactivo más allá de lo que sus fabricantes impusieron. Sony, el colega guay de la comunidad hacker, pasó a convertirse en el enemigo.

En una reunión del Chaos Computer Club de Alemania celebrada a comienzos de 2011, un grupo de expertos mostró cómo la seguridad de la supuestamente invencible consola caía trozo a trozo. No voy a repetir aquí todos sus descubrimientos, pero sí aprovecharemos para mostrar un “epic fail” del gigante japonés relacionado con una mala elección de claves.

Uno de los procedimientos que tiene la PlayStation 3 para evitar la ejecución de programas no autorizados es un sistema de firma digital. En algún lugar de la Sony Corporation hay una cámara acorazada con una clave criptográfica guardada. Cuando hay que “firmar” un juego, usan la clave y luego le dan el resultado al programador del juego para que la incluya en el disco. Al insertar el disco en la consola, ésta verifica la firma digital. Si no coincide con lo que tiene que dar, una de dos: o se ha firmado con otra clave distinta no autorizada, o bien el juego ha sido modificado. En cualquiera de los dos casos, rechaza ejecutar el juego. De ese modo, la firma digital permite a la consola determinar qué programas de pueden ejecutar en ella.

Las firmas digitales de la PS3 se basan en la llamada *criptografía de curva elíptica* (CCE), un sistema que utiliza un problema computacionalmente difícil para garantizar la seguridad. En concreto, el mecanismo de firma incorporado se denomina ECDSA (*Elliptic Curve Digital Signature Algorithm*). No nos interesan aquí los detalles, sino tan sólo el hecho de que esta firma tiene como elementos secretos una clave k y un número aleatorio m. A partir de ahí se obtienen dos elementos R, S que forman la firma digital.

Hay una condición imprescindible para que este sistema funcione adecuadamente: cada firma tiene que basarse en un número aleatorio m diferente. Si dos firmas compartiesen el mismo valor de m, el parámetro secreto k podría ser recuperado. Por supuesto, Sony sabrá eso y seguro que sus ingenieros habrán configurado un buen generador de números aleatorios para obtener diferentes valores de m, ¿no? ¡Pues no! Por algún motivo que desconocemos, las PS3 usan siempre el mismo valor de m. Imagínese usted una operadora de telefonía móvil que diese a todos sus clientes el mismo PIN; pues más o menos lo mismo.

Los hackers alemanes aprovecharon esa vulnerabilidad (que no es un fallo criptográfico, sino de implementación) y sus conocimientos técnicos sobre la consola (que Sony, en un habitual ejercicio de arrogancia, pensaba que nadie más sabría) para

obtener la clave *k*. Si quieren acudir a la fuente original, les recomiendo la presentación de la charla[40], y sobre todo el video[41]. Personalmente, me quedo con el momento en que el hacker dice: *por alguna razón, Sony usa el mismo número todo el tiempo* (parte 3, minuto 7:09). ¡El auditorio estalló en risas y aplausos! Era el momento en que los asistentes descubrían el *epic fail*.

¿Y qué se puede hacer con la clave *k*? Pues firmar cosas. Y con ello se desmonta toda la seguridad de la PlayStation 3. Todas las capas de algoritmos, protocolos y cifrados que Sony había instalado en la PS3 se vienen abajo, igual que un castillo de cartas. Epic fail en estado puro. En toda la boca. Geohot publicó la clave en su página web, y aunque fue obligado a retirarla bajo amenaza de una demanda judicial, puede encontrarse fácilmente[42]. Aquí la tiene usted:

```
BA 90 55 91 68 61 B9 77 ED CB ED 92 00 50 92 F6 6C 7A 3D 8D
```

Conociendo la clave, cualquiera puede publicar sus propias actualizaciones. La contramedida escogida por Sony fue una combinación de modificaciones en el firmware de las PS3. Las variantes de firmware 3.56 y 3.60, publicadas en enero y marzo de 2011 respectivamente, incluyeron medidas para evitar la creación de firmware "a medida," y la red PlayStation Network fue configurada para bloquear a los usuarios cuyas máquinas no estuviesen actualizadas; oficialmente, al menos en España, la función de estas actualizaciones eran añadir un parche de seguridad[43] y permitir el almacenamiento online de los datos de software, es decir, guardar partidas en la nube[44].

La modificación más importante fue un cambio en la "cadena de confianza" del sistema. En sistemas como el de la PS3, la seguridad se conforma como una cebolla: a capas. La nueva "capa de cebolla 3.60" se basaría en una clave criptográfica llamada *lv0*. Este sería el nuevo gran secreto, la llave maestra. Sorpresa, sorpresa, el nuevo secreto acabó siendo revelado. Se ignora la fecha del descubrimiento, pero la noticia saltó a la prensa en octubre de 2012: un grupo llamado "los tres mosqueteros" consiguió de algún modo acceder a la clave y la compartieron publicándola en Internet[45].

El 13 de noviembre, Sony respondió con una nota de prensa en la que amenazaba con bloquear el acceso a la red mutijugador PlayStation Network a todos aquellos que ejecuten software no autorizado[46]. La dureza de la amenaza no debe distraer del hecho de que Sony no hizo ningún anuncio sobre nuevas actualizaciones de firmware o modificaciones técnicas para contrarrestar esta amenaza. En mi opinión, les va a costar arreglar este problema. La clave *lv0* se integra en un chip durante su fabricación, y no es algo diseñado para ser modificado mediante una actualización; por no hablar de que, con la clave nueva bajo dominio público, cualquier modificación hecha por Sony puede ser deshecha con facilidad.

Tengo mucha curiosidad por ver cómo hará Sony para salir de este agujero. Mientras tanto, pueden ustedes entretenerse con las claves de la PS3 en[47]. Feliz jailbreaking a todos.

Anexo – Claves de Texas Instruments

Este Anexo contiene las claves RSA (512 bits) de las calculadoras de Texas Instruments, tal y como fueron publicadas tras su factorización a finales de 2009. Se denotan con *p,q* los números primos, y con *n* su producto. La clave privada es *d*. En todos los casos, el exponente usado es *e*=17. Todos los números están indicados en notación hexadecimal.

PRIMERA PARTE: Claves para firma digital del sistema operativo

1) Modelo TI-92+

n=AD49CA3CFEF1F2DE400B5D3790813BF3822CB0BD83E3F565CE81B3A6CEF36F
p=331792FFBB24450379CD2FA4F562961625E0EF737006A375CB9ABEA2C9D4E2
q=3644570912C38CD2D25322B5C2074DC9C40B774873F4BCEF8E1D2526237DE5
d=7020B00959AB9D2665AD0014E50853F7EAD19F89AFB19EC9678119E467CAB1

2) Modelo TI-73

n=F3FA1D8F06918D7CAA2A3D1EE76563E96F9FD0D6068647A7C17CFE427F8B0B
p=1ECCBA67FE2BFB6A29EFF138C2B55224FAE7D9ADBBAC2FE93422AB5745FDA6
q=7EBE11E729ADCBEE93031F5EE347E414F064E225169B9D389F3B499DC04BE0
d=396806F47A04214A82644A9DDC17DB45FC259A8CB63DB681D32C780FA58A20

3) Modelo TI-89

n=8976D4B5045A8988FB2BBAF8BADAFAFA4C5F8ABD5A9453D46790B33A03F6C2
p=4EEC590226B160EB0C00C1A5FE84011BC04947EDB01EB434C3581CC2D90122
q=1BDE307D27AD9ED6CF7ABB0D8F16F6E42175446D065B478CB248726E6C7F5F
d=7134AF2BA93B8052B0BA99FA034AECCE20C726F64A984509463AEDF38ACB36

4) Modelos TI-83+ / TI-83+ Silver Edition

n=82EF4009ED7CAC2A5EE12B5F8E8AD9A0AB9CC9F4F3E44B7E8BF2D57A2F2BEA
p=B709D3A0CD2FEC08EAFCCF540D8A100BB38E5E091D646ADB7B14D021096FFC
q=B7207BD184E0B5A0B89832AA68849B29EDFB03FBA2E8917B176504F08A9624
d=4D0534BA8BB2BFA0740BFB6562E843C7EC7A58AE351CE11D43438CA239DD99

5) Modelo Voyage 200

n=8307B022CEC848E14CA5D57C0C148A4803FEB19F7EEEC4493C860DF8959425

p=8FAEE8D84AB6F0AE8FCED849C52A5E5E63366D2484CE172685BADE4D908EE7
q=E974B04EBBCA3F5AF86576CEEF637470F2AA78B84BE3784613861349DB70F4
d=2689CA64972BD93334A93ECA21ABB0334C78161FDA09FD7EF3AEF50CE0B319

6) Modelo TI-89 Titanium

n=D65139FA0ADA452B80CD35C0F9ACA3604EFE1915F0D3A4232C2C3B1FFEDDF2
p=F39D6276648A571322729F44E84C895EF33AF37FB70FD498588CC6B414639C
q=E13689E94702FEAE752C61F9F793739B1C64E13AFF7B1D526A68118A517575
d=97486528F89A12B54BDC25F1A12E917128B35D006DC291FAB5C4DE70F02432

7) Modelos TI-84+ / TI-84 Silver Edition

n=EF5FEF0B0AB6E22731C17539658B2E91E53A59BF8E00FCC81D05758F26C179
p=94489014C63CC9E1E1ADB192DBBDD1F78F90A630DA9C86EFC4CBCA44E5B4D5
q=19D431AF2794229620B884E3750D622D1C74F2E4569DC15486FC8D5A3BCDFE
d=2A3E1B2010F318D9BD7C7E19300980B055A0E2A9554B77E7142E23CDF7C7CA

8) TI-92+ FlashApp

n=BC747C4065E96E3B79B9BCC1A441BC3692E264CF681C9962B763C19824D84F
p=C5F79C13DFAC64548AFFDAF9106D5495C7D1562E7E070B8CD11D94740DCF1F
q=F3B309023180214F8872DD036434BACAD21D6DBF7CD656D4F10044B85800A3
d=376D8DF4D2AE115CC972DD29E504466A676FC34C0F8FF0E0CC86C0780AD635

9) TI-73 FlashApp

n=FCBE6045900704759799BE325EA9B0E74C6541FDB9BEE21A55A8D2C85D370E
p=548E4172D99E319EC8EC3B97D23AE6F3954F1648604EE17F77786D27D3CB07
q=2FD344E1A66F486766C39065E5ADD604DFFCA71BE4098558CA0A398525196F
d=DF0254F215ABD6C21C5A7AA4EA1D41BD072C2B2B2B6C30EA0F58B9FC160367

10) TI-89 FlashApp

n=916BDE593CC9F21B07F72033A92D6DC6DCCE8622705E9F7B4C4235A00B0A0D
p=55418FD1803B562C7E0ED13B7A774F0A1B9794F626691A22C963117013C816
q=1B4A8C75455E0F8CDF2EE1E5E4627304521EAD549FF1DAE5944E524230E162
d=33533F6ACA292845C69374C6F06A62FAE485204863E5293A9362A983C7A932

11) TI-83+ FlashApp

n=878E894D2CBA39ED8191EFB30A0DF25B4DC3E5E585A80D8AEDBCD73B74167C
p=1CC2C1433A79A5D734F9F5F1FF1BC43F3F87D378142693CE26FEC1B5E9542E
q=04B697D56EA14013042B11939BBAA1ED3BAB09496DBF208785739B07947B70
d=7F9535EE4836CD1BC53E0EC6A00D2055EED67E0532800CBEFDDEE8B06D4257

12) Voyage 200 FlashApp

n=B53225EE518E9EAE0239DE47B9C3BB7F1D2647A3BB95AC6BA3E2B0FB21116B
p=49306D3448E68EDCD746D258BBD11B5E1FF5B3A56E99C9320A9A4E1A5A936B
q=279C8CB2099364B22B6CB7402FDA38EAD5C6018574DEB37C775577D430D7D6
d=8A8FC2A72F4EF1D05C0E22731595AD7F5286AF40F8DBDE343207B483CDFE43

13) TI-89 Titanium FlashApp

n=85421ED0805812E8255F7F8565D86CE20F35C3D6676797C9D73EB7CF1FF03A
p=1DC6E97D025CABDC33F94A63FB4E7A08093C788C68DF9F9E9431F4157165E0
q=479A7429046095EB8C679D13A21E90268813AC8A76FBDB46B5BB51603A3A04
d=468C6AAA9E4CBEB722D83473CC81A30E4449A3E9FA82232E9F2134225C33E2

14): TI-84+ / TI-84 Silver FlashApp

n=8F44CF7BA748D305139C11560ED3CF4D80212FA135AA5B32B7FE142EDD3B17
p=29D8D93667BB609DDA0E1C9F43774BFC1AE31E8D1FD3A7E897E53E226EDCE8
q=36C72E64900AF24D617F2C6FD68BAD1A4200E07789C34D2F7796811E18E126
d=6521836657F72B8B1CE6A2D355C2B072F1085DDB34F0B8D881E086B7AB38C5

SEGUNDA PARTE: Clave de firma para fechado temporal de los modelos:

TI-73 / Explorer
TI-83 Plus / TI-83 Silver Edition
TI-84 Plus / TI-84 Silver Edition
TI-89 / TI-89 Titanium
TI-92 Plus
Voyage 200

n=A3E337A7BB1A47198D79FC393AB0A7898FFD714E1FC80314FB61CE71481B8B
p=3D7316BFF85539DAE08FAF040631F952EB7DB77EC824F52613ECDB523FD474
q=2AAC2314B2992A3DAE35CB3106001D972C134E4F08FCEF53E1BCFAD84200C8
d=60678A266E0F751E16FC763FC82BADD872D151B57C1B4D1B66B200F75797BB

ARTURO QUIRANTES SIERRA es Profesor Titular de Física en la Universidad de Granada. Desde 1997 escribe sobre temas de criptografía y seguridad informática en el *Taller de Criptografía* (www.cripto.es). Sus aficiones incluyen la divulgación científica, que incluyen el proyecto Física de Película en una triple vertiente:

—un Proyecto de Innovación Docente para la Universidad de Granada

—un blog de física

—la creación de material docente de nivel universitario.

En la actualidad, escribe para Naukas, y ha participado recientemente en el programa de divulgación Con-ciencia de Canal Sur Televisión.

REFERENCIAS

Las referencias que se incluyen a continuación han sido comprobadas a fecha diciembre de 2012. En ocasiones, el enlace original ha desaparecido de Internet, y ha tenido que ser sustituido por una copia del repositorio archive.org. Para facilitarle la decisión sobre si seguir el enlace o no, he insertado el tamaño del archivo cuando éste supera 1 MB de tamaño. Los archivos con formato especial (PDF, PS) vienen asimismo indicados.

Notas

[1] Como ejemplo, vea el sistema de sustitución con homófonos utilizado en la primera Cifra General de Felipe II de 1556. <<

[2] “340 Cipher” *sinleb.com*. <<

[3] “Z 408 Zodiac Killer Cipher” *zodiologists.com*. <<

[4] ”Zodiac cipher” *wikipedia.com*. <<

[5] “Efficient attacks on homophonic substitution ciphers” Amrapali Dhavare. Proyecto de Master, San José State University, 2011. <<

[6] “Heuristic Search Cryptanalysis of the Zodiac 340 Cipher” Pallavi Kanagalakatte Basavaraju. Proyecto de Master, San José State University, 2009. <<

[7] “Analysis of the Zodiac 340-cipher” Thang Dao. Tesis, San José State University, 2008. <<

[8] "Tewksbury Native: I've Cracked The Code Of The Zodiac Killer" *Tewksbury Patch*, 21/07/2011. <<

[9] "Meet the man who claims he has CRACKED the code of the Zodiac Killer (and his identity)" *Daily Mail*, 22/07/2011. <<

[10] “Zodiac Cipher cracked” Bruce Schneier. *CryptoGram*, 5/08/2011. <<

[11] http://oranchak.com/zodiac/corey/hoax.html.<<

Notas

[1] “The U.S. suffered its worst airpower loss since Vietnam last week and no one really noticed” *The Atlantic Wire*, 21/12/2012. <<

[2] "RockYou hack exposes names, passwords of 30M accounts" *Computerworld*, 15/12/2009. <<

[3] “RockYou hack: from bad to worse” *TechCrunch*, 14/12/2009. <<

[4] “Imperva releases detailed analysis of 32 million breached consumer passwords” Nota de prensa de Imperva, 21/01/2010. <<

[5] “Consumer password worst practices (PDF)” *imperva.com* <<

[6] “Microsoft India store hacked, user database exposed” *PCWorld*, 131/02/2012. <<

[7] Vea un ejemplo en esta página <<

[8] Register_user_sample.png <<

[9] “YouPorn data NOT exposed” *blog.youporn.com*, 22/02/2012 <<

[10] Datos procesados por Anders Nilsson (@nilssonanders), así como esta infografía <<

[11] Cuenta de Twitter de Kevin Mitnick 11/07/2012. <<

[12] “Yahoo! takes immediate action after hacker incident” *ycorpblog.com*, 13/07/2012. <<

[13] La web original https://d33ds.co/archive/yahoo-disclosure.txt ha sido desactivada. Existen copias en torrent, por ejemplo aquí <<

[14] "Statistics about Yahoo leak of 450 000 plain-text accounts" *blog.eset.se*, 12/07/2012. <<

[15] “Web de Yahoo! Voice comprometida: publicadas más de 450,000 cuentas” José A. Guasch. *securitybydefault.com*, 12/07/2012. <<

[16] “Important notice - security breach” *androidforums.com*, 10/07/2012. <<

[17] “More user passwords dumped, this time from alleged Blllabong.com hack” *Ars Technica*, 13/07/2012. <<

[17b] “More user passwords dumped, this time from alleged Blllabong.com hack” *Ars Technica*, 13/07/2012. <<

[18] nvidia.com - forums 12/07/2012. <<

[19] “Nvidia forums suspended after large-scale hack, 390,000 accounts at risk” *The Verge*, 13/07/2012. <<

[20] “Pinterest hacked. Hundreds of thousands of users are unknowing posting spam pins” *llsocial.com*, 17/03/2012. <<

[21] "Eight million email addresses and passwords spilled from gaming site Gamigo months after hacker breach" *Forbes*, 23/07/2012. <<

[22] “8.24 million Gamigo passwords leaked after hack” *ZDNet*, 23/07/2012. <<

[23] “Months later, Gamigo hacker takes dozy dump, exposes 8 million” *The Register*, 24/07/2012. <<

[24] “Philips databases pillaged and leaked SECOND time in a month” *The Register*, 31/08/2012. <<

[25] "Hackers roban más de 30,000 bytes de datos de AMD" *bsecure.com.mx*, 20/08/2012. <<

[26] “Hackers collect significant account details from Blizzard servers” *Ars Technica*, 10/08/2012. <<

[27] “Hacker claims breach of 50,000 accounts from Wall Street IT recruiting firm” *Computerworld*, 18/07/2012. <<

[28] “Hackers Leak Thousands of Passwords From Large Private BitTorrent Tracker” *TorrentFreak*, 19/09/2012. <<

[29] “Peru Domains Registrar hacked and 207116 Domain panel credentials leaked” *The Hacker News*, 20/10/2012. <<

[30] http://pastebin.com/yXN7uc6r 2/11/2012. <<

[31] "Adobe breach reportedly spills easy-to-crack password hashes" *Ars Technica*, 10/08/2012. <<

[32] “IInyección SQL – Descripción” *Wikipedia.com* <<

[33] “PlayStation Network security update” *blog.us.playstation.com*, 2/05/2011. <<

[34] http://thepiratebay.se/torrent/6443601 <<

[35] “An important message from Sony’s chief information security officer” *blog.us.playstation.com*, 211/10/2011. <<

[36] “John the Ripper password cracker” *openwall.com* <<

[37] “Ataque de contraseñas, password guessing - 1 de 2” Alejandro Ramos. *securitybydefault.com*, 8/03/2010. <<

[38] “Ataque de contraseñas, password cracking - 2 de 2” Alejandro Ramos. *securitybydefault.com*, 26/03/2010. <<

[39] "John the Ripper cracks slow hashes on GPU" *Slashdot*, 4/07/2012. <<

[40] About us *Stratfor Global Intelligence* <<

[41] "Stratfor is a joke and so is Wikileaks for taking it seriously" *The Atlantic*, 27/02/2012. <<

[42] http://wikileaks.org/the-gifiles.html. El diario Público se encargó de su publicación en España: http://www.publico.es/internacional/wikileaks <<

[43] “Battlefield Heroes data compromised by Lulzsec” *battlefield4online.com*, 26/06/2011. <<

[44] "LulzSec says goodbye, dumping NATO, AT&T, Gamer data" *Forbes*, 25/06/2011. <<

[45] “LinkedIn passwords leaked by hackers” *BBC News*, 6/06/2012. <<

[46] “6.5 million LinkedIn password hashes leaked” *Hacker News*, 6/6/2012. <<

[47] “Updating your password on LinkedIn and other account security best practices” *blog.linkedin.com*, 6/06/2012. <<

[48] "Update on compromised passwords" *eharmony news*, 6/06/2012. <<

[49] “Actualización sobre la seguridad de las contraseñas en Last.fm” *lastfm.es*, 7/06/2012. <<

[50] “Gcrack descifrando hashes con Google” Marc Rivero López. *Caminando entre bits*, 3/09/2012. <<

[51] “Taking steps to protect our members” *blog.linkedin.com*, 7/06/2012. <<

[52] “An update on taking steps to protect our members” *blog.linkedin.com*, 9/06/2012. <<

[53] La base de datos con los valores hash de los clientes de eHarmony está disponible en http://hacktalk.net/eharmony.txt (49,1 MB). <<

[54] "Brief analysis of the Gawker password dump" *The Duo Bulletin*, 12/12/2019 <<

[55] “Gawker top-250” *duosecurity.com* <<

[56] “The only secure password is the one you can’t remember” Troy Hunt, 21/03/2011. <<

[57] "Formspring springs a leak: 28 MILLION passwords reset after raid" *The Register*, 11/07/2012. <<

[58] “Modern password hashing for your software and your servers” *openwall.com* <<

[59] "A cryptanalytic time-memory trade-off (PDF)" Martin E. Hellman. *IEEE Transactions on Information Theory* 26, 401-406 (1980). <<

[60] "How rainbow tables work" *Kestas Kuliukas*, 11/12/2006. <<

[61] Tables ophcrack.sourceforge.net <<

[62] List of Rainbow Tables RainbowCrack Project <<

[63] Free rainbow tables - distributed rainbow table project <<

[64] Rainbow tables pwcrack.com <<

[65] https://www.cryptohaze.com <<

[66] “154 billion NTLM/sec on 10 hashes” *Cryptohaze blog*, 15/07/2012. <<

[67] "A brief Sony password analysis" *Troy Hunt*, 6/06/2011. <<

[68] "Report: analysis of the Stratfor password list" *The Tech Herald*, 2/02/2012. <<

[69] “10 (or so) of the worst passwords exposed by the LinkedIn hacks *Ars Technica*, 6/06/2012. <<

[70] "Real-World passwords" Bruce Schneier. *CryptoGram*, 14/12/2006. <<

[71] "The science of guessing: analyzing an anonymized corpus of 70 million passwords (PDF)" Joseph Bonneau. *2012 IEEE Symposium on Security and Privacy* <<

[72] "Using Twitter to build password cracking wordlist" *7 habits of highly effective hackers*, 31/05/2012. <<

[73] "How I collect passwords" *xato.net*, 13/06/2011. <<

[74] "Passwords" *skullsecurity.com* 21/09/2011. <<

[75] “Florida man arrested in ‘Operation Hackerazzi’ for targeting celebrities with computer intrusion, wiretapping, and identity theft” Nota de prensa del FBI, 12/10/2011. <<

[76] "PIN analysis" *datagenetics.com*, 3/09/2012. <<

[77] "Trade group exposes 100,000 passwords for Google, Apple engineers" *Ars Technica*, 25/09/2012. <<

[78] "Important note regarding a change in your password" *IEEE log*, 25/09/2012. <<

[79] “Contraseña para sistema online de suministro de agua potable: '0-0-0-0' ss” *DiarioTI.com, 9/09/2011* <<
>

[80] "Insurgents hack U.S. drones" *The Wall Street Journal*, 17/12/2009. <<

[81] "The great brazilian sat-hack crackdown" *Wired*, 20/04/2009. <<

[82] "WhatsApp al descubierto" *securitybydefault.com*, 5/01/2012. <<

[83] “Are my messages secure?” *WhatsApp Support*, 15/08/2012. <<

[84] "WhatsApp is using IEMI numbers as passwords" *samgranger.com*, 5/09/2012. <<

[85] “Lo que no te cuenta WhatsApp” Yago Jesús. *securitybydefault.com*, 9/06/2011. <<

[86] "Descifrando el fichero msgstore.db.crypt de WhatsApp" Alejandro Ramos. *securitybydefault.com*, 7/05/2012. <<

[87] “Finding Your iPhone’s Unique Identifier (UDID)” *innerfence.com* <<

[88] “Antisec leaks 1,000,001 UIUDs from a trove of 12 million allegedly stolen from an FBI laptop” *TechCrunch*, 4/09/2012. <<

[89] “Statement from BlueToad regarding the cyber attack suffered in the recent case of stolen Apple UDIDs” *blog.bluetoad.com*, 10/09/2012. <<

[90] “Apple Legacy filevault barn door… “ David I. Emery. *cryptome.org*, mayo 2012. <<

[91] “OS X Lion v10.7.3: Las contraseñas de las cuentas de usuario aparecen en los archivos de registro de FileVault Original y/o en los directorios de inicio de red” *support.apple.com* <<

[92] “Yahoo Axis Chrome extension leaks private certificate file” Nik Cubrilovic. *nikcub.com*, 24/05/2012. <<

[93] “Extensión Yahoo Axis para Chrome publica su clave privada” *INTECO*, 27/05/2012. <<

[94] “Outlook webmail passwords restricted to 16 chars - how does that compare with Yahoo and Gmail?” *Naked Security*, 2/08/2012. <<

[95] "Secret Microsoft policy limited Hotmail passwords to 16 characters" *Ars Technica*, 24/0972012. <<

[96] "Microsoft gives away Windows 8 Pro to pirates by accident" *ZDNet*, 21/11/2012. <<

[97] “De contraseñas demasiado complejas” Alejandro Ramos. *securitybydefault.com*, 22/08/2012. <<

[98] Datos del cliente: Registro, modificación, política de seguridad y confidencialidad *elcorteingles.es* <<

[99] "Millions of Virgin Mobile accounts at risk of password attacks" *Ars Technica*, 19/09/2012. <<

[100] “Virgin Mobile fails web security 101, leaves six million subscriber accounts wide open” Kevin Burke. *kev.inburke.com*, 17/09/2012. <<

[101] "Prince William photos slip-up forces MoD to change passwords" *The Guardian*, 20/11/2012. <<

[102] “A working day in the life of Flight Lieutenant Wales” Web oficial de los Duques de Cambridge, 20/11/2012. <<

[103] "Prince William photos accidentally reveal RAF password" *Naked Security*, 21/11/2012. <<

[104] “Security tip: When being interviewed on TV, make sure passwords aren’t written behind you” *Naked Security*, 10/05/2012. <<

[105] “Security tip: Before being interviewed on TV, wipe passwords off whiteboard” *Naked Security*, 24/08/2012. <<

[106] “Anonymous hacks Syrian President’s email. The password: 12345” *Mashable*, 7/02/2012. <<

[107] http://pastebin.com/uaYDfCz0 <<

[108] Greek Ministry of Finance credentials *anompaste.me*, 31/10/2012. <<

[109] “Keeping presidents in the nuclear dark (Episode #1:The case of the missing ‘Permissive Action Links’ ” [Archive.org - 29/06/2011] *Bruce Blair’s Nuclear Column*, 11/02/2004. <<

[110] "Russian military forces have 'safe busting' sledgehammer" *RiaNovosti*, 6/06/2012. <<

[111] http://cryptome.org/isp-spy/bios-spy.pdf <<

[112] http://www.accounttech.us/bios.htm<<

[113] http://www.pwcrack.com/bios.shtml <<

[114] Password Safe <<

Notas

[1] “Baigent vs. The Random House Group Ltd (PDF)” [Archive.org - 5/12/2006] HC04C03092, High Court of Justice, Chancery Division. <<

[2] "How judge's secret Da Vinci code was cracked" *The Guardian*, 28/04/2006. <<

[3] "Baigent vs. The Random House Group Ltd" Caso nº A3 2006/0971, Supreme Court of Appeals (Civil Division). <<

[4] “Entrada ‘Código’, RAE”, acepciones 7ª y 8ª. <<

[5] “The da Vinci Code web quest” *randomhouse.com* <<

[6] El concurso estaba restringido a Estados Unidos, Reino Unido y Australia. La web oficial del concurso es esta, y los acertijos resueltos están disponibles en http://davinciquest.blogspot.com/ <<

[7] Si quiere usted regalarme uno por mi cumpleaños: http://www.cryptex.org/ <<

[8] http://www.criptex.es/ <<

[9] http://www.schneier.com/ No dejen de visitar su magnífico blog. <<

[10] http://www.schneierfacts.com/ <<

[11] "*Los códigos secretos*" Simon Singh, capítulo 1. Círculo de Lectores SA, 2000. <<

[12] “Origins of cryptography: the arab contributions” Ibrahim A. Al-Kadi. *Cryptologia* 16, 97-126 (1992) Resumen <<

[13] “Charting Arabic cryptology’s evolution” (Kathryn A. Schwartz) *Cryptologia* 33, 297-394 doi: 10.1080/01611190903030904 (2009) Resumen <<

[14] Autor norteamericano experto en los servicios de inteligencia norteamericano. Su libro de 1982 *The Puzzle Palace* fue pionero en describir el funcionamiento de la NSA. <<

[15] “Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance (PDF)” Varios autores. *Nature* Vol.414, 20/27 diciembre 2001, pp. 883-887. <<

[16] "D-Wave Systems sells its first Quantum Computing System to Lockheed Martin Corporation" Nota de prensa de *D-Wave Systems*, 25/05/2011. <<

[17] “Quantum annealing with manufactured spins” Varios autores. *Nature* Vol.473, pp. 194-198 (12/05/2011) Resumen <<

[18] "D-Wave quantum computer solves protein folding problem" *Nature newsblog*, 12/08/2012. <<

[19] "D-Wave Systems, Inc., the World's First Commercial Quantum Computing Company, Secures $30 Million in a New Equity Round From Investors Including Bezos Expeditions and In-Q-Tel" Nota de prensa de *IN-Q-TEL*, 20/09/2012. <<

[20] “Intelligence community partners” *iqt.com* <<

[21] “The Nobel Prize in Physics 2012” *nobelprize.org* <<

[22] "Interview with David J. Wineland" *nobelprize.org*, 9/10/2012. <<

[23] La propia NSA ha publicado la historia sobre la operación VENONA y puede consultarse online: “The Venona Story” Robert L. Benson. <<

[24] “La alternativa ‘key escrow’ ” I - debate abierto (28/11/1999), II - posturas nacionales (28/11/1999), III - España es así (16/12/1999). Arturo Quirantes Sierra. *Taller de Criptografía* (Informes 16 – 18). <<

[25] "Protocol failure in the Escrowed Encryption Standard (PDF)" Matt Blaze. *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, pp. 59-67 (1994). <<

[26] http://cripto.es/aquirantes_cripto_es.asc Clave PGP del autor. <<

[27] Véase, por ejemplo, algunos ejemplares encontrados en el Cuartel General del Ejército español, en 2009: http://www.cripto.es/museo/enigma-esp-fotos.htm <<

[28] La web de Dan Brown ha cambiado desde entonces, pero puede verse todavía una copia de la foto en esta dirección [Archive.org - 7/12/2003] <<

[29] "'Digital Fortress:' un libro prescindible" Fernando Acero. *Kriptopolis.org*, 20/03/2006. <<

Notas

[1] “Criptografía: entrevista con Arturo Quirantes” *Más Allá de la Ciencia*, nº 236. <<

[2] http://www.randi.org <<

[3] “Challenge info” *randi.org* <<

[4] “My life as an international arms courier” Matt Blaze. Existe traducción en castellano <<

[5] “James Randi’s Swift – 17/12/2007” [Archive.org - 7/06/2011]. <<

[6] “James Randi owes me a million dollars” <<

[7] "How to explain zero-knowledge protocols to your children (PDF)" Varios autores. *CRYPTO'89 Proceedings on Advances in Cryptology*, pp. 628-631 (1989). <<

[8] “Final examination Contracts I, Section 1 (PDF)” Professor Jimenez, Spring Semester 2007, Stetson University of College Law. <<

[9] “James Randi’s Swift – 2/02/2007” [Archive.org - 8/01/2011]. <<

Notas

[1] “Recalls and safety alerts involving pacemakers and implantable cardioverter-defibrillator” (W.H. Maisel, M.O. Sweeny, W.G. Stevenson, K.E. Ellison, L.M. Epstein). *JAMA* 286, 793-797 (2001). Resumen disponible aquí <<

[2] VeriChip fue retirado en 2010 por problemas relacionados con su seguridad y privacidad. <<

[3] “Una discoteca catalana implantará un chip bajo la piel a personajes famosos” *El Mundo*, 17/03/2004. <<

[4] "MicroCHIPS announces clinical results for first successful human trial of implantable, wireless microchip drug delivery device" Nota de prensa de *MicroCHIPS*, 16/02/2012. <<

[5] Se recomienda el artículo "Security and privacy for implantable medical devices (PDF 1,2 MB)" Daniel Halperin, Thomas S. Heydt-Benjamin, Kefin Fu, Tadayohsi Kohno, William H. Maisel. *IEEE Pervasive Computing* 8, 30-39 (2008) <<

[6] “Insulin Pumps Vulnerable to Hacking” *Fox News*, 4/07/2011 <<

[7] “Improving the security and privacy of implantable medical devices” (William H Maisel, Tadayoshi Kohno) *The New England Journal of Medicine* 362, 1164-6 (2010). Resumen disponible aquí <<

[8] “Hooligans Attack Epilepsy Patients During Epilepsy Awareness Month” *pr.com*, 19/11/2007. <<

[9] “Hackers Assault Epilepsy Patients via Computer” *Wired*, 28/3/2008. <<

[10] “Spooks want to go fishing in Oyster database” *The Register*, 17/03/2008. <<

[11] "Student Expelled for Refusing Location Tracking RFID Badge" *Infowars.com*, 19/11/2012. <<

[12] “Texas school district’s RFID tracking of students goes to court” *Ars Technica*, 23/11/2012. <<

[13] “Mifare: little security despite obscurity” *24th Chaos Communication Congress*, 28/12/2007. <<

[14] "Reverse-engineering a cryptographic RFID tag (PDF)" Karsten Nohl, David Evans, Starbug, Henryk Plötz. *17th Usenix Security Symposium* (2008) <<

[15] “Microscope-wielding boffins crack Tube smartcard” *The Register*, 12/03/2008. <<

[16] “Security flaw in MIFARE Classic (PDF)” Varios autores. <<

[17] "A practical attack on the MIFARE Classic (PDF)" Gerhard de Koning Gans, Jaap-Henk Hoepman, Flavio D. García. *Lecture Notes in Computer Science* 5189, 267-282 (2008) <<

[18] "Dismantling MIFARE Classic (PDF)" Varios autores. *Lecture Notes in Computer Science* 5283, 97-114 (2008) <<

[19] "Wirelessly Pickpocketing a Mifare Classic Card (PDF)" Flavio D. García, Peter van Rossum, Roel Verdult, Ronny Wichers Scheur. *30th IEEE Symposium on Security and Privacy* pp 3-15 (2009) <<

[20] “Anatomy of a subway hack (PDF 4,2 MB)” Russell Ryan, Zack Anderson, Alessandro Chiesa. *DEFCON 2008* <<

[21] "Black Hat organizers punt totally hackable RFID badges" *The Register*, 8/08/2008. <<

[22] “Free rides: the story of the smartphone subway hackers” *Mashable*, 10/10/2012. <<

[23] UltraCardTester (Google Play) <<

[24] “Android NFC hack enables travelers to ride subways for free, researchers say” *Computerworld*, 27/07/2012. <<

[25] "NXP responds to NFC transit security hack" Nota de prensa de *NXP*, 24/09/2012. <<

[26] “Pacemakers and implantable cardiac defibrillators: software radio attacks and zero-power defenses (PDF 2,1 MB)” Varios autores. *IEEE Symposium on Security and Privacy*, mayo 2008. <<

[27] “They can hear your heartbeats: non-invasive security for implantable medical devices (PDF)” Shyamnath Gollakota, Haitham Hassanieh, Benjamin Ransford, Dina Katabi, Kevin Fu. *ACM Special Interest Group on Data Communication (SIGCOMM11)*, agosto 2011. <<

[28] "Activity-aware ECG-based patient authentication for remote health monitoring (PDF)" Janani Sriram, Minho Shin, Tanzeem Choudury, David Kotz. *International Conference on Multimodal Interfaces (ICMI09)*, noviembre 2009. <<

[29] “Patients, pacemakers, and implantable defibrillators: human values and security for wireless implantable medical devices (PDF)” Varios autores. *ACM Conference on Human Factors in Computing Systems*, Abril 2010. <<

[30] “Security that is meant to be deep skin: using ultraviolet micropigmentation to store emergency-access keys for implantable medical devices (PDF)” Stuart Schechter, 8/04/2010. <<

Notas

[1] "The 15 Biggest Tech Disappointments of 2007" *PCWorld*, 26/12/2007. <<

[2] “Analysis of the MediaMax CD3 Copy-Prevention System” J. Alex Halderman, 06/10/2003. <<

[3] "Shift key breaks CD copy locks" *Cnet*, 07/10/2003. <<

[4] “SunnComm to sue ‘Shift key’ student for $10m” *The Register*, 09/10/2003. <<

[5] “SunnComm Backs Down” EFF, 14/10/2003. <<

[6] "Threat of lawsuit passes for student" *The Daily Princetonian*, 10/10/2003. <<

[7] “SunnComm Responds” Ed Felten. *Freedom to Tinker*, 07/10/2003. <<

[8] “Sony CD copy protection installs a rootkit on users PC’s” Mark Russinovich, 30/10/2005. El artículo original de Russinovich no se encuentra en la Red, pero hay copias en otras páginas web <<

[9] "Study of Sony Anti-Piracy Software Triggers Uproar" *The Washington Post*, 02/11/2005. <<

[10] La web con la actualización http://updates.xcp-aurora.com/ ya no se encuentra activa. <<

[11] “Sony Music CDs Under Fire from Privacy Advocates” *npr.org*, 4/11/2005. <<

[12] “SonyBMG and First4Internet Release Mysterious Software Update” Ed Felten. *Freedom to Tinker*, 03/11/2005. <<

[13] "World of Warcraft hackers using Sony BMG rootkit" *Security Focus*, 03/11/2005. <<

[14] "Sony's Rootkit and the DMCA" *Emergent Chaos*, 17/11/2005. <<

[15] Spyware Detail - XCP.Sony.Rootkit [Archive.org - 08/07/2011] *Computer Associates*, 05/11/2005. <<

[16] “CA targets Sony DRM as spyware” *ZDNet*, 08/11/2005. <<

[17] "XCP" *McAfee* (sin fecha) <<

[18] "Sony DRM Rootkit" *TechNet Blogs*, 12/11/2005. <<

[19] “Sony to stop making rootkit’ CDs” Security Focus, 11/11/2005. <<

[20] "Sony to pull controversial CDs, offer swap" *USA Today*, 14/11/2005. <<

[21] "Sony Numbers Add Up to Trouble" *Wired*, 15/11/2005. <<

[22] "Sony's Web-Based Uninstaller Opens a Big Security Hole; Sony to Recall Discs" Ed Felten. *Freedom to Tinker*, 15/11/2005. <<

[23] "Sony Shipping Spyware from SunnComm, Too" Ed Felten. *Freedom to Tinker*, 11/11/2005. <<

[24] "Media Max access control vulnerability" *iSEC Partners*, 29/11/2005. <<

[25] “MediaMax Bug Found; Patch Issued; Patch Suffers from Same Bug” Ed Felten. *Freedom to Tinker*, 07/12/2005. <<

[26] "EMI Music launches DRM-free superior sound quality downloads across its entire digital repertoire" Nota de prensa de *EMI*, 02/04/2007. <<

[27] "Universal and Rhapsody launch DRM-free partnership 'test' " *Engadget*, 21/08/2007. <<

[28] "Three down, one to go: Warner Music Group drops DRM" *Ars Technica*, 27/12/2007. <<

[29] “Sony BMG Plans to Drop DRM” *Businessweek*, 04/01/2008. <<

[30] www.dvdcca.org <<

[31] DVD CCA Frequently asked questions and andswers dvdcca.org <<

[32] http://www.pigdog.org/decss/ <<

[33] "Why the DVD Hack Was a Cinch" *Wired*, 02/11/1999 <<

[34] “International Traffic in Arms Regulations” EPIC <<

[35] “DVD Encryption Broken” Bruce Schneier. *CryptoGram*, 15/11/1999. <<

[36] “Netscape obtiene aprobación federal para exportar Netscape Communicator con cifrado seguro de 128 bits” [Archive.org - 29/01/1999] Nota de prensa de *Netscape Communications Corporation*. <<

[37] “Cryptanalysis of Contents Scrambling System” Frank A. Stevenson. *insecure.org*, 08/11/1999. <<

[38] http://www.dvddecrypter.org.uk/ <<

[39] “Request for expressions of interest (PDF)” DVD Copy Control Association, 26/10/2005. <<

[40] “Sony copy protection taking heat again: now DVDs won’t play” *Engadget*, 16/04/2007. <<

[41] AACS specifications aacsla.com <<

[42] “Copy protection hole in Blu-ray and HD DVD movies” *h-online.com* Un mes después, Corel Corporation anunció su intención de adquirir Intervideo, lo que llevaron a cabo en diciembre. El programa WinDVD sigue vendiéndose en la actualidad. <<

[43] "BackupHDDVD, a tool to decrypt AACS protected movies" *forum.doom9.org*, 27/12/2006. <<

[44] "The Saga of decrypting an AACS protected movie, by Muslix64" *forum.doom9.org*, 27/12/2006. <<

[45] "Setting the record straigth" *forum.doom9.org*, 02/01/2007. El programa fue retirado por orden legal, pero existen copias. <<

[46] “Here it is, alpha version of BackupBluRay V0.21!” *forum.doom9.org*, 20/01/2007. Hay una copia del programa disponible en esta dirección. <<

[47] www.aacskeys.com [Archive.org - 6/02/2007] <<

[48] www.hdkeys.com [Archive.org - 27/02/2007] <<

[49] "Post HD DVD Volume Unique Keys here - post questions in the forum" *forum.doom9.org*, 13/01/2007 a 25/07/2010. <<

[50] 2010-01-03_KEYDB_HD.zip *forum.doom9.org* <<

[51] 2010-06-15_KEYDB_BD.zip *forum.doom9.org* <<

[52] “A letter to customers from Corel CEO David Dobson” [Archive.org - 02/01/2007] Nota a clientes, *Corel.com* <<

[53] "Processing Key, Media Key and Volume ID found!!!" *forum.doom9.org*, 05/02/2007. <<

[54] (sin título) *forum.doom9.org*, 02/02/2007. *Serenity* parece haber sido la primera película de alta definición en haber sido pirateada. Según *DailyTech*, ya había copias en bittorrent a fecha 17 de enero de 2007. <<

[55] (sin título) *forum.doom9.org*, 11/02/2007. <<

[56] (sin título) *forum.doom9.org*, 11/02/2007. <<

[57] Programa disponible en esta dirección. <<

[58] "WinDVD 8 Device Key Found!" *forum.doom9.org*, 24/02/2007. <<

[59] “PowerDVD private key” *forum.doom9.org*, 04/03/2007. <<

[60] AnyDVD History Versiones 6.1.2.3 y 6.1.3.0, respectivamente. <<

[61] AACS key update *Corel.com* <<

[62] CyberLink Blu-ray Disc and HD DVD Update Center [Archive.org - 13/05/2007]
<<

[63] “Free speech flag” *badmouth.com* <<

[64] "Digg This: 09-f9-11-02-9d-74-e3-5b-d8-41-56-c5-63-56-88-c0" [Archive.org - 11/02/2010] *digg.com*, 1/05/2007. <<

[65] “DVD DRM row sparks user rebellion” *BBC News*, 2/05/2007. <<

[66] “The infamous hexadecimal code T-shirt” *nerdyshirts.com* <<

[67] "Oh Nine, Eff Nine" *youtube.com*, 1/05/2007. <<

[68] “Photoshop rebels rib great HD DVD clampdown” *Wired*, 5/03/2007. <<

[69] “09-f9-11-02-9d-74-e3-5b-d8-41-56-c5-63-56-88-c(…)” *photoree.com*, 3/05/2007. <<

[70] "Software HD-DVD/Blu-ray Players Updated" Ed Felten. *Freedom to Tinker*, 13/04/2007. <<

[71] "You Can Own an Integer Too - Get Yours Here" Ed Felten. *Freedom to Tinker*, 7/05/2007, comentario de fecha 23/05/2007. <<

[72] "New Processing Key found!! (MKB v3 is now open)" *forum.doom9.org*, 30/05/2007. <<

[73] AnyDVD History Versión 6.1.5.4. <<

[74] Versión 6.1.9.3AnyDVD History <<

[75] AnyDVD History Versión 6.2.0.1. <<

[76] “Toshiba Announces Discontinuation of HD DVD Businesses” Nota de prensa de *Toshiba*, 19/02/2008. <<

[77] AnyDVD History Versión 6.4.0.0. <<

[78] “Protección” *es.dvdfab.com* <<

[79] “BD/DVD Tools” *pavtube.com* <<

[80] “Conditions for High Definition Labelling of Display Devices” *digitaleurope.org* <<

[81] “HDCP technologies” digital-cp.com <<

[82] “HDCP Encryption/Decryption Code” (Rob Johnson, Mikhail Rubnich) <<

[83] “A cryptanalysis of the High-bandwidth Digital Content Protection System (PostScript)” Scotr Crosby, Ian Goldberg, Robert Johnson, Dawn Song, David Wagner. 3/08/2001. <<

[84] “Censorship in action: why I don’t publish my HDCP results” [Archive.org - 16/07/2011]. Niels Ferguson. *mcfergus.com*, 15/08/2001. <<

[85] “Is the leaked HDCP master key real?” [Archive.org - 26/11/2010] *pastebin.com*, 13/09/2010. Existe copia de la clave, con instrucciones en español, aquí. <<

[85b] "Is the leaked HDCP master key real?" [Archive.org - 26/11/2010] *pastebin.com*, 13/09/2010. Existe copia de la clave, con instrucciones en español, aquí. <<

[86] "HDCP Master Key Confirmed; Blu-ray Content Vulnerable" *PCMag*, 16/09/2010. <<

[87] “Checkmate! Researchers outsmart Intel copy protection HDCP” phys.org, 28/11/2011. <<

[88] “Exemption to prohibition on circumvention of copyright protection systems for access control technologies” U.S. Copyright Office. <<

[89] Circular 1/2006 sobre los delitos contra la propiedad intelectual e industrial tras la reforma de la Ley Orgánica 15/2003 (PDF). <<

[90] “Cómo sobrevivir a la SGAE” Arturo Quirantes Sierra. <<

Notas

[1] “Decimalisation table attacks for PIN cracking (PDF)” (Mike Bond, Piotr Zieliński) Technical Report # 560, University of Cambridge, febrero 2003. <<

[2] "The man who invented the cash machine" *BBC News*, 25/06/2007. <<

[3] “Ross Anderson (web profesional)” Universidad de Cambridge. <<

[4] Faxes (Bowes & Turner, abogados). <<

[5] Cartas y documentación (Ross Anderson) *cryptome.org* <<

[5b] Cartas y documentación (Ross Anderson) *cryptome.org* <<

[6] Carta de Ross Anderson 19/02/2003. <<

[7] “Creditcard-holders ‘must pay’ ” [Archive.org - 29/10/2004] *news24.com*, 8/08/2005. <<

[8] "Frequently Asked Questions (FAQ) about the Electronic Frontier Foundation's 'DES Cracker' machine" *EFF*, 16/07/1998. <<

[9] "Tribunal de Distrito de Francfort del Meno, Número de documento 30 C 2119/97 hasta 45, 1/09/1008" [Archive.org - 8/05/2009, traducido del alemán mediante Google Translate] *www.ccc.de* <<

[10] "Crypto in Europe - Markets, Law and Policy (PDF)" Ross J. Anderson. *Cryptography: Policy and Algorithms*, Springer LNCS 1029, 75-89 (1995). <<

[11] Carta y documentación Ross Anderson. *doc.ic.ac.uk* <<

[12] "Card fraud and computer evidence" Ross Anderson. *The Risks Digest* 15, 17/02/1994. <<

[13] “Why Cryptosystems Fail” Ross Anderson. <<

[14] "The unbearable lightness of PIN cracking (PDF)" Omer Berkman, Odelia Moshe Ostrovsky. *Proceedings of the 11th International Conference on Financial cryptography and 1st International conference on Usable Security*, pp. 224-238 (2007). <<

[15] www.emvco.com Europay fue absorbida por MasterCard en 2002. JCB se unió a la asociación en 2004, y American Express lo hizo en 2009. <<

[16] “EMVCo Integrated Circuit Card Specifications for Payment Systems - Book 2: Security and key management” (Hacer clic en “*Agree*“). <<

[17] "Global EMV Adoption Continues to Grow" *EmvX blog*, 30/01/2012. <<

[18] "BBVA lanza una nueva familia de tarjetas que se adapta a las preferencias de pago de sus clientes y hace más sencillo su uso" Nota de prensa de *BBVA*, 23/10/2009. <<

[19] “La gran banca frena la implantación de la tecnología chip en las tarjetas” *Cinco Días*, 2/06/2009. <<

[20] "Millions in danger from chip and pin fraudsters" *Mail Online*, 5/06/2006. <<

[21] “Visa update for EMV Chip implementation in the U.S”. *EmvX blog*, 17/01/2012. <<

[22] "MasterCard aligns with Visa's U.S. EMV migration plans by publishing its own EMV implementation roadmap" *EmvX blog*, 1/02/2012. <<

[23] “APACS response to BBC Watchdog findings on chip and PIN” Nota de prensa de *APACS*, 6/02/2007. <<

[24] "Card fraud: banks now have to prove your guilt" *The Telegraph*, 15/10/2009. <<

[25] “Shell’s £1m chip and PIN fraud ‘an inside job’ ” *ZDNet*, 9/05/2006. <<

[26] “New card threat to bank customers” *BBC News*, 26/10/2009. <<

[27] "Egg acts over 'risky' customers" *BBB News*, 2/02/2008. <<

[28] Carta del Defensor del Cliente (*Financial Ombudman Service*) 18/01/2007. <<

[29] “Banking: the PIN and the ATM” Stephen Mason. *www.stephenmason.eu* <<

[30] “Fraud victims told: Go to the bank, NOT the police” *London Evening Standard*, 30/03/2007. <<

[31] “Growing epidemic of card cloning” Ross Anderson, 26/07/2006. <<

[32] “Chip & PIN terminal playing Tetris” *youtube.com*, 3/01/2007. <<

[33] “Chip & PIN (EMV) relay attacks” Saar Drimen, Steven J. Murdoch. <<

[34] “Thinking inside the box: system-level failures of tamper proofing (PDF 5,1 MB)” Saar Drimen, Steven J. Murdoch, Ross Anderson. Technical Report # 711, University of Cambridge, febrero 2008. <<

[35] Carta de B. Dunn (VISA) a Steven Murdoch (PDF) 26/02/2008. <<

[36] "How secure is Chip and PIN?" *BBC News*, 26/02/2008. <<

[37] “APACS response to clarification questions raised by Stephen Murdoch of Cambridge University (PDF)” <<

[38] "Fraud ring funnels data from cards to Pakistan" *The Wall Street Journal – European Edition*, 11/10/2008. <<

[39] “Understanding terminal manipulation at the point of sale (PDF)” Folleto de MasterCard. <<

[40] "Portable credit card terminals used in fraud" *thestar.com*, 31/05/2012. <<

[41] "2008 fraud figures announced by APACS" Nota de prensa de *UK Payments Administration*, 19/03/2009. <<

[42] “Optimised to fail: card readers for online banking (PDF 2 MB)” Saar Drimer, Steven J. Murdoch, Ross Anderson. *Financial Cryptography and Data Security '09, Barbados* 02/2009. <<

[43] "New card and banking fraud figures" Nota de prensa de *The UK Cards Association*, 10/03/2011. <<

[44] "DDA on EMV cards offers best-in-class security to reduce fraud" *Financial Services & Retail Newsletter*, Julio 2009. <<

[45] “DDA authentication in Europe - payment card security at its best” *gi-de.com* <<

[46] “Indicadores estadísticos de la migración a SEPA (PDF)” *Comisión de Seguimiento de la migración a SEPA (sepaesp.es)* <<

[47] “Axalto y Caixa Penedes son pioneros en la migración hacia la tecnología EMV en España” *Business Wire*, 18/04/2005. <<

[48] “Gemalto respalda la migración de Diners Club España al sistema EMV” Nota de prensa de *Gemalto*, 2/02/2012. <<

[49] "El sector de las tarjetas de pago en España (PDF)" Ahmad Rahnema. *IESE Business School (Navarra)*, Estudio nº 39, julio 2006. <<

[50] “Los bancos españoles pierden 198 millones de euros en 2011 por el fraude en tarjetas” *Expansión*, 16/05/2012. <<

[51] Directiva 2007/64/CE del Parlamento Europeo y del Consejo de 13/11/2007, sobre servicios de pago en el mercado interior (PDF). Diario Oficial de la Unión Europea L 319/1 a 319/36, 5/12/2007. <<

[52] Ley 16/2009, de 13 noviembre, de servicios de pago Boletín Oficial del Estado, Num.275, pp.96887-96918, 14/11/2009. <<

[53] "FSA begins new banking regulation to promote fairness for consumers" Nota de prensa de la *Financial Services Authority*, 28/10/2009. <<

[54] "Card fraud: banks now have to prove your guilt" *The Telegraph*, 15/10/2009. <<

[55] “Chip and PIN is broken (PDF 1,4 MB)” Stephen J. Murdoch, Saar Drimer, Ross Anderson, Mike Bond. *2010 IEEE Symposium on Security and Privacy pp. 433-446.* <<

[56] “EMV PIN verification ‘wedge’ vulnerability” Stephen J. Murdoch, Saar Drimer, Ross Anderson, Mike Bond. <<

[57] "New flaws in chip and pin system revealed" *BBC News*, 11/02/2010. <<

[58] "The UK Cards Association's Response to BBC Newsnight's item on Chip and PIN is broken" Nota de prensa de *The UK Cards Association*, 11/02/2010. <<

[59] “Chip and pin is broken” Ross Anderson, 11/02/2011. <<

[60] Id. [59], comentario 19 <<

[61] Id. [59], comentario 22<<

[62] IP Information for 193.128.116.71 *whois.domaintools.com* <<

[63] "Banking industry worker faces cosh over anonymous rant" *The Register*, 24/02/2010. <<

[64] http://code.google.com/p/smartcarddetective/ <<

[65] “Smart Card Detective” Smart Architects. <<

[66] http://www.smartcarddetective.com/ <<

[67] “The Smart Card Detective: a hand-held EMV interceptor (PDF 2,7 MB)” Omar S. Choudary. Tesis. Universidad de Cambridge, junio 2010. <<

[68] “Responsible disclosure practice (PDF)” Carta de Melanie Johnson (UK Cards Association) a Stephen Jolly (Universidad de Cambridge), 1/12/2010. <<

[69] “Responsible disclosure and academic freedom (PDF)” Carta de Ross Anderson a Melanie Johnson, 24/12/2010. <<

[70] “Press Articles on UKCA request” web de Omar Choudary. <<

[71] “Publication of materials that assist crime (PDF)” Carta de Melanie Johnson a G. P. Allen (Universidad de Cambridge), 4/04/2011. <<

[72] “Responsible disclosure and academic freedom (PDF)” Carta de Ross Anderson a Melanie Johnson, 2/12/2011. <<

[73] “Worldwide EMV Deployment” *EMVCo* <<

[74] “New Cryptography Drafts” *EMVCo* <<

[75] Sentencia STS 8466/2009 (PDF) 16/12/2009. <<

Notas

[1] “Hiding information and signatures in trapdoor knapsacks (PDF)” Ralph C. Merkle, Martin E Hellman. *IEEE Transactions on Information Security* 24, 525-530 (1978). <<

[2] "A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem (PDF)" Adi Shamir. *IEEE Transactions on Information security* 30, 600-704 (1984). <<

[3] En 1998, Rivest, Shamir y Adleman vendieron su empresa por 200 millones de dólares. Los accionistas de RSA Security aprobaron en 2006 su venta a EMC Corporation por una cantidad que se mide en miles de millones de dólares. <<

[4] The RSA Factoring Challenge *RSA Laboratories*. <<

[5] “Ron was wrong, Whit is right (PDF)” Varios autores. <<

[6] "Actualización para la longitud mínima de clave de certificado" *Documento informativo sobre seguridad de Microsoft (2661254)*, 14/07/2012. <<

[7] "TI-83 Plus OS Signing Key Cracked" *ticalc.org*, 31/07/2009. <<

[8] Mensaje de Floppus Maximus *United TI forum*, 21/07/2009. <<

[9] Puede ver un ejemplo de aviso DMCA en http://brandonw.net/calcstuff/DMCA_notice.txt <<

[10] El contenido original ha sido sustituido por una nota que dice: "*Querida comunidad, se me ha pedido amablemente que retire el contenido de este mensaje*". <<

[11] RSA Lattice Siever. En la actualidad, este proyecto distribuido se ha fusionado en la iniciativa nfs@home y continúa factorizando números primos, aunque ya no están relacionados con Texas Instruments. <<

[12] http://wlstorage.net/file/ti-os-keys-dmca-2009.txt <<

[13] Carta a Texas Instruments (PDF 1,2 MB), Jennifer Granick, 31/11/2009. <<

[14] http://www.brandonw.net/ (entrada 29 agosto 2011). <<

[15] “Prohibido calcular: las claves” Arturo Quirantes Sierra. *Boletín ENIGMA* nº 71, 01/11/2009. <<

Notas

[1] “Sigsaly Story” Patrick D. Weadon. *Historical Publications, Center for Cryptologic History, National Security Agency*. <<

[2] “Sigsaly - The start of the digital revolution” J.V. Boone, R. R. Peterson. *Historical Publications, Center for Cryptologic History, National Security Agency*. <<

[3] “El proceso de implantación de la telefonía móvil en España (PDF)” (Antonio Pérez Yuste) *Revista Antena del COITT*, septiembre 2002. <<

[3b] “El proceso de implantación de la telefonía móvil en España (PDF)” (Antonio Pérez Yuste) *Revista Antena del COITT*, septiembre 2002. <<

[4] “Txiki Benegas se enzarza con Solchaga en una nueva polémica sobre el plan de viviendas del PSOE” *El País*, 23/04/1991. <<

[5] “Aquí, el problema es el ‘one’, no Solchaga” *El País*, 26/04/1991. <<

[6] “Transportes abre una investigación técnica sobre las escuchas a Benegas” *El País, 27/04/1991*. <<

[7] “Benegas: ‘Me grabó un profesional porque circulaba a 200 por hora’ (PDF)” *ABC*, 2/05/1991. <<

[8] “Corcuera confía en que la Guardia Civil sepa quién siguió el coche de Benegas” *El País*, 9/05/1991. <<

[9] “Las cintas se captaron por un procedimiento ‘inocente y casual’ ” *El País*, 28/04/1991. <<

[10] “Un informe oficial dice que las cintas de Benegas se grabaron desde otro coche” *El País*, 31/05/1991. <<

[11] “El juez del ‘caso Benegas’ afirma que la difusión de las cintas no es delictiva” *El País, 17/08/1991*. <<

[12] “El juez archiva la querella de Benegas contra la SER por difundir sus conversaciones” *El País*, 5/10/1991. <<

[13] “Primer juicio a un periodista por escuchas telefónicas en Almería” *El País*, 28/02/1992. <<

[14] “Telefónica echa el cierre a Moviline tres años antes de lo previsto” *Cinco Días*, 31/12/2003. <<

[15] "Cryptanalysis of ORYX (PDF)" Varios autores. *Fifth Annual Workshop on Selected Areas in Cryptography*, Springer Verlag, agosto 1998. <<

[16] "Cryptanalysis of the cellular authentication and voice encryption algorithm (PDF)" William Millan, Praveen Gauravaram. *IEICE Electronics Express* 1, 453-459 (2004). <<

[17] "Cryptanalysis of the Cellular Message Encryption Algorithm (PDF)" David Wagner, Bruce Schneier, John Kelsey. *17th Annual International Cryptology Conference*, pp.526-537 (1997). <<

[18] “CTIA: Encryption of Digital Wireless Phones” 20/03/1997. <<

[19] “Counterpane’s Reply to CTIA” 20/03/1997. <<

[20] “QUALCOMM Responds to Digital Security Issues” 20/03/1997. <<

[21] “Pacific Bell Mobile Services response statement to cellular cryptography research conducted by U.C. Berkeley” 20/03/1997. <<

[22] “Despite codebreakers, Omnipoint handsets remain 100% secure” 20/03/1997. <<

[23] “Powertel responds to wireless security issue” 20/03/1997. <<

[24] “Cryptanalysis of the Improved Cellular Message Encryption Algorithm (PDF)” Thomas Chardin, Raphaël Marinier. <<

[25] www.gsma.com <<

[26] “Specification numbering” *3gpp.org* <<

[27] “An implementation of the GSM A3A8 algorithm (Specifically, COMP128)” Marc Briceno, Ian Goldberg, David Wagner. 1998. <<

[28] “GSM cloning”. <<

[29] “COMP128 y los afectados” Traducido del alemán mediante Google Translate. <<

[30] ftp://ftp.ccc.de/pub/software/gsm/ <<

[31] “GSM Alliance clarifies false & misleading reports of digital phone” *Business Wire*, 20/04/1998. <<

[31b] "GSM Alliance clarifies false & misleading reports of digital phone" *Business Wire*, 20/04/1998. <<

[32] “Cryptographers announce break in authentication encryption for GSM phones” 13/04/1998. <<

[33] “Concerned telecom and network security specialists respond to claims of GSM cellular phone cloning” 13/04/1994. <<

[34] http://www.jcea.es/artic/gsm.htm <<

[35] “GSM cloning” (David Wagner). <<

[36] “COMP128: a birthday surprise (PDF)” Stuart Wray, 11/05/2003. <<

[37] “Partitioning attacks: or how to rapidly clone coms GSM cards (PostScript)” Varios autores. *2002 IEEE Symposium on Security and Privacy*. <<

[38] “A brief history on the withdrawal of the A5/2 ciphering algorithm in GSM” (Harald Welte) 12/11/2010. <<

[39] "Solutions to the GSM security weakness (PDF)" Mohsen Toorani, Ali A. Beheshti. *Proceedings of the 2008 The Second International Conference on Next Generation Mobile Applications, Services, and Technologies*, pp.576-581 (2008). <<

[40] http://www.jcea.es/artic/gsm.txt <<

[41] “A5 (Was: Hacking digital phones)” Ross Anderson, 17/06/1994. <<

[42] “A pedagogical implementation of A5/1” Marc Briceno, Ian Goldberg, David Wagner. <<

[43] “A pedagogical implementation of the GSM A5/1 and A5/2 ‘voice privacy’ encryption algorithms” Marc Briceno, Ian Goldberg, David Wagner. <<

[44] “Technical information - GSM System Security Group” Racal Research Ltd. 10/06/1998. <<

[45] “The (real-time) cryptanalysis of A5/2 (PostScript)” Ian Goldberg, David Wagner, Lucky Green. 26/08/1999. <<

[46] "Withdrawal of A5/2 algorithim support" *Osmocom Security* (11/11/2010). <<

[47] "Cryptanalysis of Alleged A5 Stream Cipher" Jovan Dj. Golic. *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pp.239-255 (1997). <<

[48] "Real Time Cryptanalysis of A5/1 on a PC" Alex Biryukov, Adi Shamir, David Wagner. *Proceedings of the 7th International Workshop on Fast Software Encryption* (2000). <<

[49] "Researchers claim to have broken privacy code for wireless phones" *The New York Times*, 7/12/1999. <<

[50] “European cellular encryption algorithms” Bruce Schneier. *CryptoGram*, 15/12/1999. <<

[51] “Security in the GSM network (PDF)” Marcin Olawski. <<

[52] “Instant ciphertext-only cryptanalysis of GSM encrypted communication (PDF)” Elad Barkan, Eli Biham, Nathan Keller. *Journal of Cryptology* 21, 392-429 (2008). <<

[53] “How to Break DES for € 8980 (PDF)” Varios autores, 2006. Una nota de los investigadores avisa al lector: “*sí, lo sabemos, la famosa playa de Rio de Janeiro se deletrea de forma algo diferente, Copacabana, pero llegar al nombre actual ya nos ha hecho penosamente conscientes de nuestras limitadas habilidades imaginativas*”. <<

[54] "Cryptanalysis with COPACOBANA (PDF 1,5 MB)" Tim Güneysu, Timo Kasper, Martin Novotny, Andy Rupp *IEEE Transactions on Computers* 57, 1498-1513 (2008). <<

[55] "A real-world attack breaking a5/1 within hours (PDF)" Timo Gendrullis, Martin Novotny, Andy Rupp. *Proceedings of the 10th International Workshop on Cryptographic Hardware and Embedded Systems*, pp.266-282 (2008). <<

[56] “New project uses distributed computing to break GSM crypto” *The Tech Herald*, 28/08/2009. <<

[57] "GSM Alliance downplays seriousness of GSM project" *The Tech Herald*, 28/08/2009. <<

[58] “GSM - SRSLY? (PDF)” Karsten Nohl, Chris Paget. *26th Chaos Communication Congress*, Berlin (2009). <<

[59] “GSMA statement on media reports relating to the breaking of GSM encryption” Nota de prensa de *GSM Association*, 30/12/2009. <<

[60] http://opensource.srlabs.de/projects/a51-decrypt/files <<

[61] “New ‘Kraken’ GSM-cracking software is released” *Computerworld*, 21/07/2010. <<

[62] “Welcome to Airprobe” ccc.de <<

[63] "STOA Report: Interception Capabilities 2000" Duncan Campbell. Sección 4, "*Comint and Law Enforcement*. "Ver también Informe nº 16 del Taller de Criptografía (apartado "Amistades tenebrosas") <<

[64] “The Hellenic Radio (ERA): News in english” 6/02/2002. <<

[65] “MS user manual (PDF 2,7 MB)” quintessenz.at <<

[66] "The Athens affair" *IEEE Spectrum*, julio 2007 <<

[67] “Greek Wiretapping Scandal” Bruce Schneier. *CryptoGram*, 22/07/2006 <<

[68] "Greece fines Ericsson Hellas in tapping case" *Reuters*, 6/09/2007. <<

[69] "Vodafone's 'tappers' named" *The Guardian*, 2/07/2006. <<

[70] “GPRS intercept: wardriving your country (PDF)” Karsten Nohl, Luca Melette. *Chaos Communication Camp 2011*. <<

[71] “Mobile phone eavesdropping made easy: hackers crack GPRS encryption” *Computerworld*, 10/08/2011. <<

[72] "Hackers crack crypto for GPRS mobile networks" *The Register*, 10/08/2011. <<

[73] "Don't trust satellite phones: a security analysis of two satphone standards (PDF)" Benedikt Driessen, Ralf Hund, Carsten Willems, Christof Paar, Thorsten Holz. *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, pp.128-142 (2012). <<

[74] “3gpp confidentiality and integrity algorithms”*3gpp.org* <<

[75] "ETSI to distribute openly 3GPP confidentiality and integrity algorithms" Nota de prensa de *ETSI*, 4/09/2000. <<

[76] “Specification of the 3GPP confidentiality and integrity algorithms. Document 2: Kasumi specification (PDF)” Technical Specification ETSI TS 135 202 (versión 7.0.0, junio 2007). <<

[77] “Specification of the A5/3 encryption algorithms for GSM and ECSD, and the GEA3 encryption algorithm for GPRS (PDF)” Technical Specification 3GPP TS 55.216 (versión 6.2.0, septiembre 2009). <<

[78] “GSM Association welcomes formation of open mobile alliance” [Archive.org - 9/05/2008) Nota de prensa de *GSM World*, 1/07/2002. <<

[79] “Specification of the 3GPP confidentiality and integrity algorithms. Document 1: f8 and f9 specification (PDF)” Technical Specification ETSI TS 135 201 (versión 7.0.0, junio 2007). <<

[80] "A related-key rectangle attack on the full KASUMI (PostScript)" Eli Biham, Orr Dunkelman, Nathan Keller. *Proceedings of the 11th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2005).* <<

[81] “A practical-time attack on the A5/3 cryptosystem used in third generation GSM telephony (PDF)” Orr Dunkelman, Nathan Keller, Adi Shamir. 2010. <<

[82] "Spy games turn real as eavesdropping technology spreads" *Radio Praga*, 16/08/2012. <<

Notas

[1] “Dilly Knox – A reminiscence of this pioneer Enigma Cryptanalyst” (Mavis Batey) *Cryptologia* 32:2, 104-130 (2008) Resumen <<

[2] “De cifras” (Miguel Gómez del Campillo) *Boletín de la Real Academia de la Historia*¸nº 129, pp-279-230 (1951). El descifrado formaba parte de un trabajo de compilación, el *Archivo Documental Español*, publicado por la Real Academia de la Historia, Tomo IV, Negociaciones con Francia - 1562; pp. 81-84 (1951). <<

[3] “Visualization of potential weakness of existing cipher engine implementations in commercial on-the-fly encryption software (PDF)” C. B. Roellgen, 15/08/2008. <<

[4] “Encrypted image backups open to new attack” *Techworld*, 3/10/2008. <<

[5] “ ‘New Attack’ Against Encrypted Images” Bruce Schneier. *CryptoGram*, 9/10/2008. <<

[6] PMC Ciphers <<

[7] “$10,000 Challenge: PMC Ciphers, Inc. will pay $10,000 to anyone who can break their cipher” *turbocrypt.com* <<

[8] “Bruce Schneier puts down Polymorphic Encryption” C. B. Roellgen, mayo 2007. <<

[9] Sentencia 173/2011 del Tribunal Constitucional 7/11/2011. <<

[10] “Nikon encrypts D2X white balance data” *Photoshop News*, 17/04/2005. <<

[11] Nikon advisory *Nikondigital.com*, 22/04/2005. <<

[12] “Bibble Labs releases Bibble 4.2.6 with support for Nikon D2Hs & D2X, Canon 350D” [Archive.org - 23/03/2006] *Bibble Labs*, 28/04/2004. <<

[13] http://www.cybercom.net/~dcoffin/dcraw/dcraw.c Página web de Dave Coffin. <<

[14] In depth: Camera Raw Adobe.com <<

[15] Adobe and Nikon Adobe.com <<

[16] Nikon D2x and D2xs review Página web de Thom Hogan, 3/05/2005. <<

[17] Nikon Image Authentication Software Nota de prensa de *Nikon*, 1/06/2006. <<

[18] "Musing on the Want et al. MD5 collision (PDF)" Philip Hawkes, Michael Paddon, Gregory G. Rose. 13/10/2004. <<

[19] “Colliding X.509 certificates (PDF)” Arjen Lenstra, Xiaoyun Wang, Benne de Weger. 01/03/2005. <<

[20] “MD5 considered harmful today” Varios autores. <<

[21] “Researchers use PlayStation cluster to forge a web skeleton key” *Wired*, 30/12/2008. <<

[22] “Governments and banks still using weak-MD5-signed ssl certificates” *Netcraft.com*, 31/08/2012. <<

[23] "U.S., Israel developed Flame computer virus to slow Iranian nuclear efforts, officials say" *The Washington Post*, 19/06/2012. <<

[24] “Chosen-prefix collisions for MD5 and colliding X.509 certificates for different identities (PDF)” Marc Stevens, Arjen Lenstra, Berne de Weger. 16/06/2009. <<

[25] "CWI cryptanalyst discovers new cryptographic attack variant in Flame spy malware" *Centrum Wiskunde & Informatica*, 07/06/2012. <<

[26] “Los certificados digitales no autorizados podrían permitir la suplantación de identidad” Documento informativo sobre seguridad de Microsoft (2718704), 03/06/2012. <<

[27] "Flame malware collision attack explained" *TechNet Blogs*, 06/06/2012. <<

[28] “Microsoft throws 'kill switch’ on own certificates after Flame hijack” *Computer World*, 04/06/2012. <<

[29] "Update to Windows Update, WSUS Coming This Week" *TechNet Blogs*, 06/06/2012. <<

[30] "El tamaño de mi clave" Arturo Quirantes Sierra, *Boletín ENIGMA* nº 62, 01/08/2008. <<

[31] "Nikon Image Authentication System: Compromised" *Advanced Password Cracking – Insight*, 28/04/2011. <<

[32] "Canon original data security systen vulnerability" *Elcomsoft.com*, 30/11/2010. <<

[33] “Comments on Media Reports about Nikon’s imaging product” Nota de prensa de *Nikon*, 09/09/2011. <<

[34] www.bufetalmeida.com <<

[35] “Why Iraq’s buying un Sony PlayStation 2s” *WND*, 19/12/2000. <<

[36] "Astrophysicist Replaces Supercomputer with Eight PlayStation 3s" *Wired*, 17/10/2007. <<

[37] “PlayStation 3 ‘hacked’ by iPhone cracker” *BBC World*, 25/01/2010. <<

[38] "PS3 Firmware (v3.21) Update" *blogs.us.playstation.com*, 28/03/2010. <<

[39] “Funciones de la actualización (v.3.21)” *es.playstation.com*, 1/04/2010. <<

[40] “Console Hacking 2010 - PS3 epic fail (PDF 8,7 MB)” *events.ccc.de* <<

[41] "Sony's PS3 security is epic fail - videos within" *psx-scene.com*, 29/12/2010. <<

[42] Página web de geohot [Archive.org - 4/01/2011] <<

[43] “Funciones de la actualización (v.3.56)” *es.playstation.com*, 27/01/2011. <<

[44] “Funciones de la actualización (v.3.60)” *es.playstation.com*, 10/03/2011. <<

[45] http://pastie.org/private/bevpt5jf9kdjg3vrrv05w <<

[46] “Sony responds to Playstation 3 LV0 bootloader keys leak; banning warnings issued” *Examiner*, 31/11/2012. <<

[47] http://www.ps3devwiki.com/wiki/Keys <<