Criptografía y Métodos de Cifrado

Héctor Corrales Sánchez

Carlos Cilleruelo Rodríguez

Alejandro Cuevas Notario

Índice:

1. ¿Qué es la criptografía?

Antes de zambullirnos en el mundo de la criptografía creemos necesario aclarar en qué cosiste la criptografía.

Según la RAE:

> Criptografía: Arte de escribir con clave secreta o de un modo enigmático.

Aportando una visión más específica, la criptografía es la creación de técnicas para el cifrado de datos. Teniendo como objetivo conseguir la confidencialidad de los mensajes. Si la criptografía es la creación de mecanismos para cifrar datos, el criptoanálisis son los métodos para "romper" estos mecanismos y obtener la información. Una vez que nuestros datos han pasado un proceso criptográfico decimos que la información se encuentra cifrada.

Cabe destacar el uso incorrecto del termino encriptar, que proviene de una mala traducción del inglés encrypt. La palabra encriptar no está reconocida por la RAE y el término correcto es cifrar. La interpretación del término encriptar sería introducir cuerpos en una cripta.

2. Historia de la criptografía. Enigma

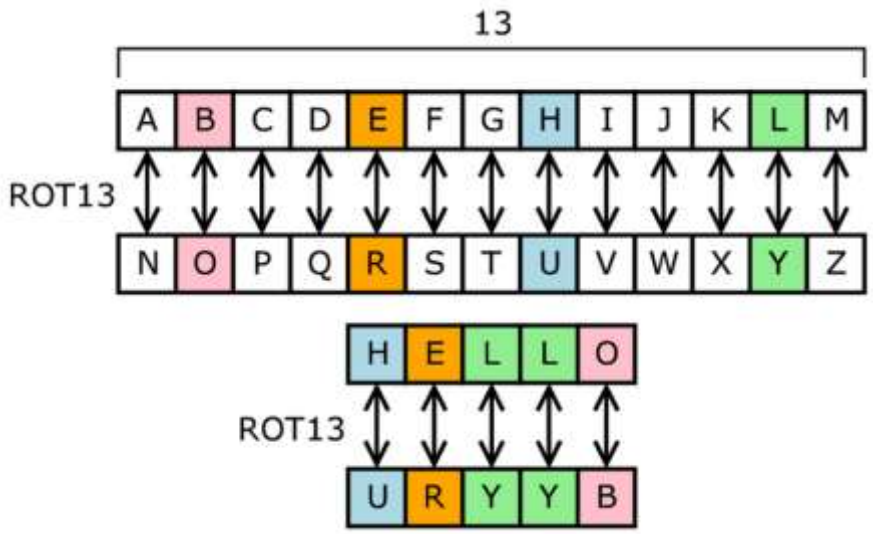
El primer sistema criptográfico del que se tiene constancia es la Escítala. Este sistema data del siglo V a.c. y era usado en Esparta.

El sistema consistía en dos varas del mismo grosor, una en poder del emisor y la otra del receptor. Cuando el emisor quería enviar un mensaje, este, enrollaba una cinta en su vara y escribía el mensaje. De este modo al desenrollar la cinta el mensaje era ilegible. Al recibir el mensaje, el receptor enrollaba la cinta en su vara, y de este modo podía leer el mensaje.

Los primeros sistemas de cifrado estuvieron ligados a campañas militares dada la necesidad de evitar que el enemigo obtuviese los movimientos de las tropas al interceptar mensajes.

Otro método de cifrado clásico es el conocido cifrado de Cesar. Su nombre viene de la supuesta utilización por parte de Julio de César de este sistema.

El cifrado de César es un cifrado de sustitución monoalfabética. Este sistema consiste en desplazar el alfabeto una cantidad determinada de posiciones y alinearlo con el alfabeto sin desplazar. De esta forma se obtiene una relación entre las letras.



Ya en el siglo XV es inventado un sistema de sustitución polialfabética por Leon Battista Alberti. Este sistema es conocido como cifrado Vigenere, al haber sido atribuido por error a Blaise de Vigeniere. Con este sistema cada letra tiene una correspondencia única, haciendo más difícil el descifrado.

En el siglo XX, a consecuencia de las dos guerras mundiales, la criptografía sufre un gran avance. En el año 1920 comenzó a usarse la máquina enigma. Su fama se debe a su uso por parte del ejército Alemán. Enigma hacía uso de partes mecánicas y eléctricas, era un mecanismo de cifrado rotatorio. La facilidad para cifrar, descifrar mensajes y la supuesta seguridad de este cifrado convierten a la máquina enigma en una pieza clave de la segunda guerra mundial.

Los esfuerzos por romper Enigma impulsaron la criptografía y el criptoanálisis de una forma inimaginable. Durante la segunda guerra mundial los aliados finalmente consiguen descifrar Enigma, aunque este hecho se mantiene oculto hasta finales de los años 60.

3. ¿Por qué es necesaria la criptografía?

La criptografía siempre había estado vinculada al ámbito militar. ¿Por qué se hizo necesaria para el resto de la gente?

Aunque el uso de comunicaciones seguras ha sido siempre una prioridad militar, la privacidad es requerida en otros sectores. Las empresas necesitan mantener unas comunicaciones seguras para proteger su información. Por esta razón el gobierno de EEUU y la NSA se ven obligados a crear DES.

Aparte de a las empresas, se hace necesario otorgar al ciudadano de privacidad y seguridad. Con el nacimiento de internet y la progresiva oferta de servicios telemáticos como acceso al banco, citas médicas y un sinfín de posibilidades se tiene que ofrecer confidencialidad y seguridad a estos servicios.

Por estas razones es necesaria la criptografía. Para otorgar privacidad, confidencialidad y seguridad a nuestras transacciones telemáticas.

4. Usos de la criptografía

La criptografía cuenta con 3 usos: Cifrar, autenticar y firmar.

Cifrar:
Como ya hemos dicho, siempre hay cierta información que no queremos que sea conocida más que por las personas que nosotros queramos. En esto nos ayuda el cifrado. Cifrando un mensaje hacemos que este no pueda ser leído por terceras personas consiguiendo así la tan deseada privacidad.

Autenticación:
Otra de las necesidades que surgen con la aparición de internet es la necesidad de demostrar que somos nosotros y que el emisor es quien dice ser. Un método de autenticación puede ser el propio cifrado. Si ciframos un mensaje con una clave solo conocida por nosotros, demostrando que somos quien decimos ser, el receptor podrá constatar nuestra identidad descifrándolo. Esto se puede conseguir mediante clave simétrica (el receptor tiene que estar en posesión de la clave empleada) o usando clave asimétrica en su modo de autenticación.

Firmar:
Dados los trámites que podemos realizar hoy en día a través de internet se hace necesaria la aparición de la firma digital. Igual que firmamos un documento, la firma digital nos ofrece la posibilidad de asociar una identidad a un mensaje.

Para la firma digital se utiliza clave asimétrica (dos claves una privada y otra pública). Lo que se cifra con la clave privada (que solo nosotros conocemos) sólo se puede descifrar con la pública. De esta forma al cifrar con nuestra clave privada demostramos que somos nosotros.

La firma digital tiene un problema. ¿Cómo sabe el receptor que la clave corresponde realmente con la persona o entidad que dice poseerla? De este modo surgen las entidades de certificación. Organismos de confianza que actúan como notarios.

Otro sistema existente la red de confianza. En esta red los usuarios certifican si los demás son quien dicen ser. De este modo podría decirse que cada usuario se constituye como entidad certificadora.

5. Criptografía simétrica: DES y AES

La criptografía Simétrica es un método criptográfico monoclave, esto quiere decir que se usa la misma clave para cifrar y descifrar. Esto supone un grave problema a la hora de realizar el intercambio entre el emisor y el receptor, dado que si una tercera persona estuviese escuchando el canal podría hacerse con la clave, siendo inútil el cifrado.

Es importante que la clave sea difícil de adivinar y el método de cifrado empleado el adecuado. Hoy en día, con la capacidad computacional disponible, si se emplean los algoritmos adecuados, dependiendo del método de cifrado empleado se puede obtener una clave en cuestión de minutos-horas.

DES:

El algoritmo DES (Data Encryption Standard) es un algoritmo de cifrado desarrollado por la NSA a petición del gobierno de EEUU bajo la presión de las empresas por la necesidad de un método para proteger sus comunicaciones. DES fue escogido como FIPS (Federal Information Processing Standard) en el año 1976 y su uso se extendió por todo el mundo. Hoy en día DES es considerado inseguro dada su clave de 56 bits, insuficiente frente al poder computacional actual. En su variante Triple DES el algoritmo se cree seguro.

DES es un algoritmo de cifrado por bloques. Se toma un bloque de una longitud fija de bits y lo transforma mediante una serie de operaciones básicas en otro bloque cifrado de la misma longitud. En el caso de DES el tamaño del bloque es de 64 bits. La clave también tiene 64 bits pero 8 de estos bits se emplean para comprobar la paridad, haciendo que la longitud efectiva de la clave sea de 56 bits.

DES se compone de 16 fases o rondas idénticas. Al comienzo y al final se realiza una permutación. Estas permutaciones no son significativas a nivel criptográfico, pues se incluyeron para facilitar la carga y descarga de los bloque en el hardware de los años 70. Antes de cada ronda el bloque se divide en dos mitades de 32 bits y se procesan alternativamente. Este proceso es conocido como esquema Feistel. El esquema Feistel nos proporciona un proceso de cifrado y descifrado casi iguales. La única diferencia es que las subclaves se aplican de forma inversa al descifrar.

Función F-Feistel:

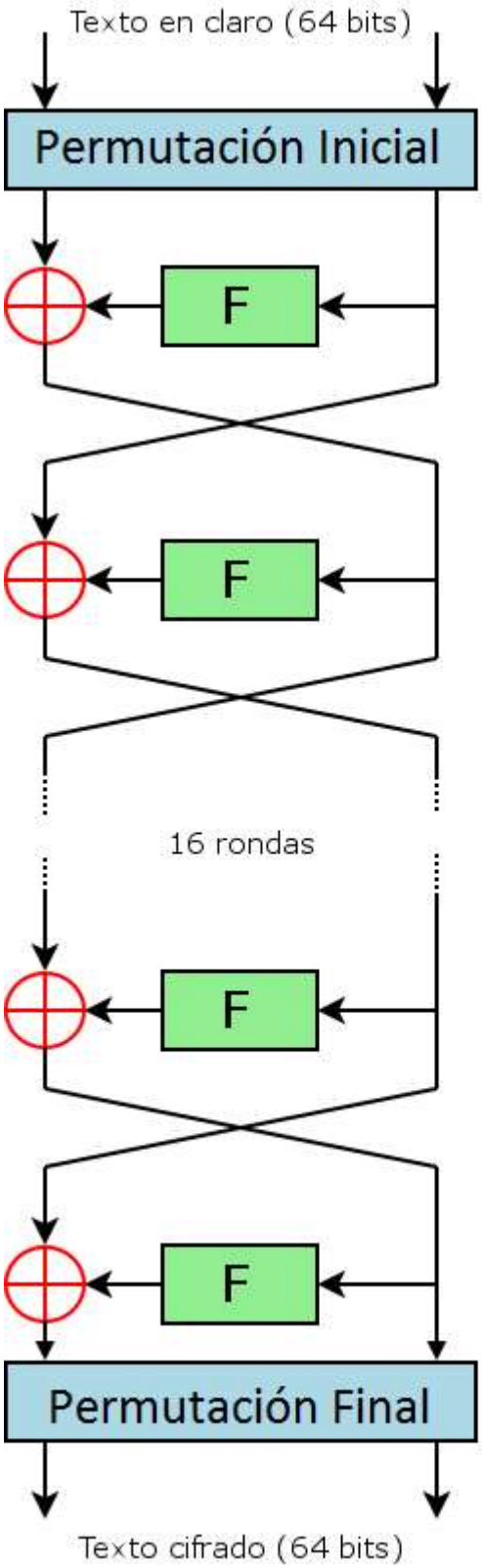
1. Expansión: se toma la mitad del bloque de 64 bits (32bits) que son expandidos a 48 bits mediante la permutación de expansión, denominada E en el diagrama, duplicando algunos de los bits.
2. Mezcla: el resultado se combina con una subclave utilizando una operación XOR. Dieciséis subclaves (una para cada ronda) se derivan de la clave inicial mediante la generación de subclaves.
3. Sustitución: tras la mezcla, el bloque es dividido en ocho trozos de 6 bits que se pasan a las cajas de sustitución. Cada una de las ocho S-cajas reemplaza sus seis bits de entrada con cuatro bits de salida, de acuerdo con una trasformación no lineal, especificada por una tabla. Las S-cajas constituyen el núcleo de la seguridad de DES, sin ellas, el cifrado sería lineal y fácil de romper.
4. Permutación: finalmente, los 32bits salientes de las S-cajas se reordenan de acuerdo a una permutación fija.

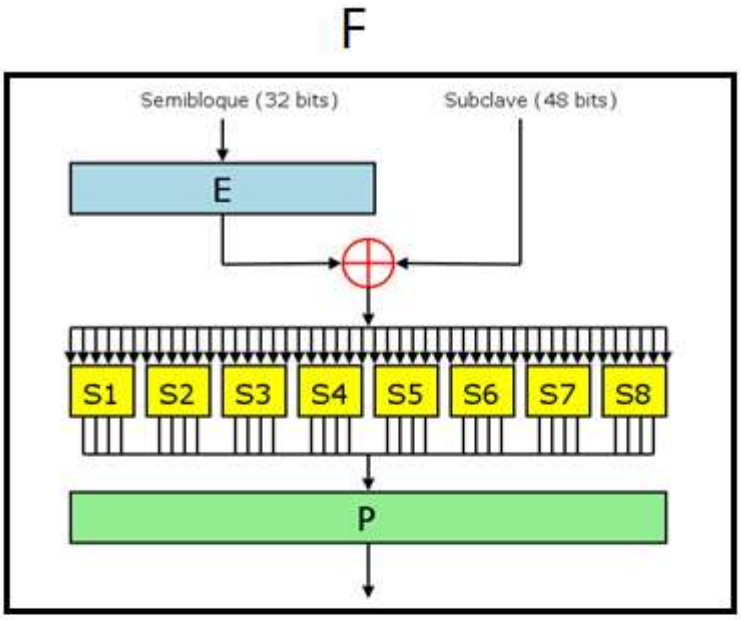
Generación de claves:

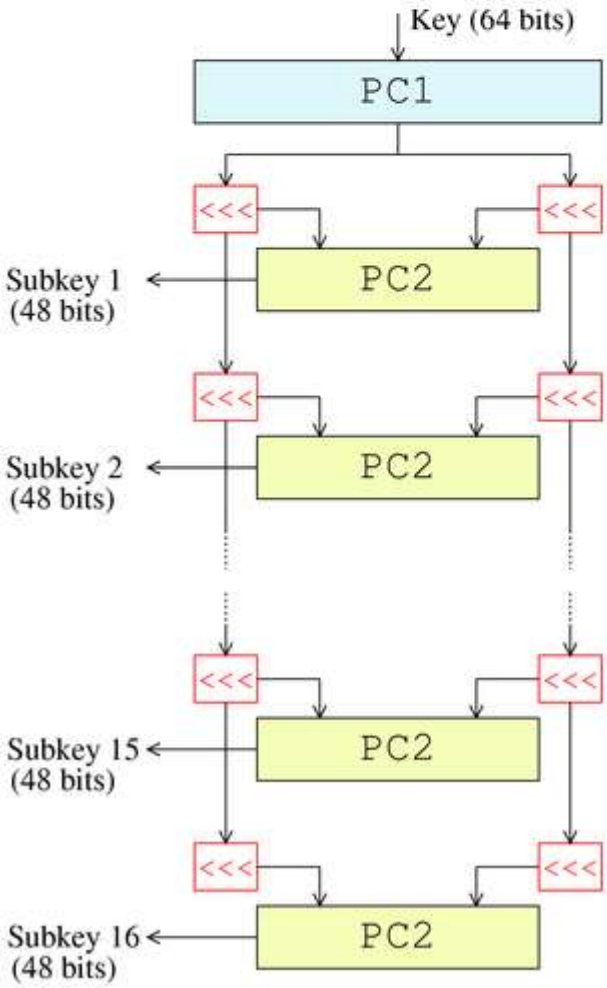
De los 64 bits iniciales se toman 56 con la Elección Permutada 1 (PC-1). Estos 56 bits son divididos en dos mitades de 28 bits que seran tratadas de forma independiente. En rondas sucesivas se desplazan los bits de ambas mitades 1 o 2 bits a la derecha. Tras el desplazamiento se toman 48 bits (24+24) mediante la

Elección Permutada 2 (PC-2). Al realizar un desplazamiento en cada ronda cada subclave estará empleando un conjunto diferente de bits.

La generación de claves para descifrado es similar, la única varaición es que se deben generar en orden inverso.







AES:

El algoritmo AES (Advanced Encryption Standard) tambien conocido como Rijndael fue el ganador del concurso convocado en el año 1997 por el NIST (Intituto Nacional de Normas y Tecnología) con objetivo de escoger un nuevo algoritmo de cifrado. En 2001 fue tomado como FIPS y en 2002 se transformó en un estándar efectivo. Desde el año 2006 es el algoritmo más popular empleado en criptografía simétrica.

AES opera sobre una matriz de 4x4 bytes. Mediante un algoritmo se reordenan los distintos bytes de la matriz. El cifrado es de clave simétrica, por lo que la misma clave aplicada en el cibrado se aplica para el descifrado.

Basado en El algoritmo Rijndael, Al contrario que su predecesor DES, Rijndael es una red de sustitución-permutación, no una red de Feistel. AES es rápido tanto en software como en hardware, es relativamente fácil de implementar, y requiere poca memoria.

El algoritmos AES funciona mediante una serie de bucles que se repiten. 10 ciclos para claves de 128 bits, 12 para 192 y 14 para 256.

Supongamos que tenemos 2 matrices: matriz a y matriz k.

$$\begin{matrix} a00 & a01 & a02 & a03 \\ a10 & a11 & a12 & a13 \\ a20 & a21 & a22 & a23 \\ a30 & a31 & a32 & a33 \end{matrix}$$

$$\begin{matrix} k00 & k01 & k02 & k03 \\ k10 & k11 & k12 & k13 \\ k20 & k21 & k22 & k23 \\ k30 & k31 & k32 & k33 \end{matrix}$$

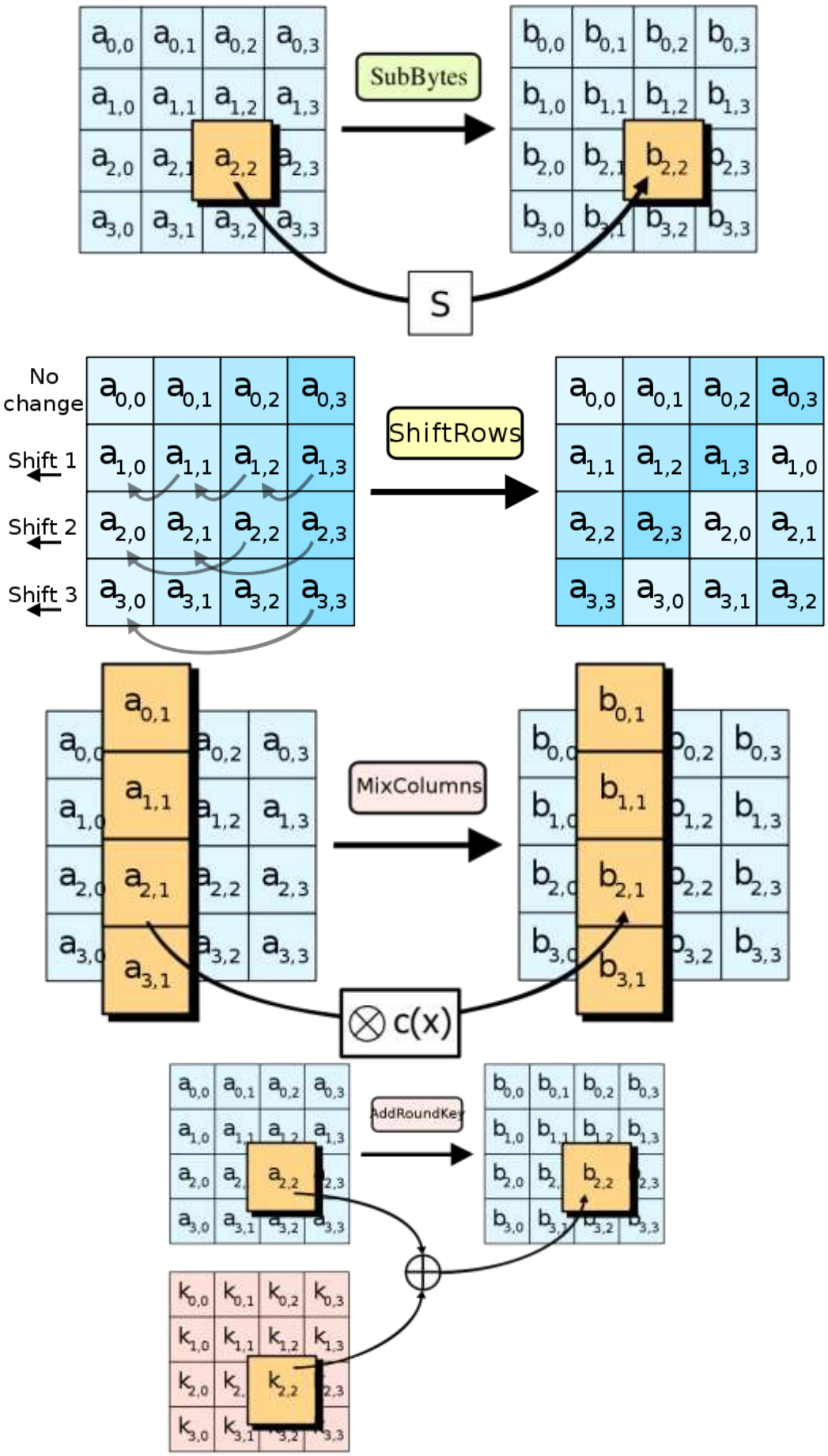
En la matriz a tenemos nuestra información y en la matriz k tenemos una subclave generada a partir de la principal.

El algoritmo de cifrado es el siguiente:

1. Expansion de la clave. Mediante una serie de operaciones se obtienen n+1subclaves a partir de la clave principal (n es el numero de ciclos).
2. Ciclo inicial. AddRoundKey. Se aplica una XOR byte byte entre la matriz a y la matriz k.
3. Ciclos intermedios.
   3.1. SubBytes. Tomando como referencia una tabla especificada cada byte es sustituido por otro en función de la tabla.
   3.2. ShiftRows. Cada byte de cada fila es desplazada n-1 huecos a la izquierda (siendo n el numero de fila).
   3.3. MixColumns. Los 4 bytes de una columna se combinan entre si para obtener 4 bytes diferentes. Este proceso se logra multiplicando la columna por una matriz dada.

$$\begin{matrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{matrix}$$

   3.4. AddRoundKey.
4. Ciclo final.
   4.1. SubBytes.
   4.2. ShiftRows
   4.3. AddRoundKey


SubBytes
S
No change
Shift 1
Shift 2
Shift 3
ShiftRows
MixColumns
c(x)
AddRoundKey

6. One-time pad

One-time pad es un tipo de algoritmo de cifrado por el que el texto en claro se combina con una clave aleatoria o «libreta» de la misma longitud y que sólo se utiliza una vez. Fue inventado en 1917. Si la clave es verdaderamente aleatoria, nunca se reutiliza y, por supuesto, se mantiene en secreto, se puede demostrar que el método de la libreta de un solo uso es irrompible.

Proviene del cifrado de Vernam. El sistema de Vernam es un cifrado que combina un mensaje con una clave que se lee de un bucle de cinta de papel. En su forma original, el sistema de Vernam era vulnerable porque la clave se podía reutilizar. El uso único vino un poco después, cuando Joseph Mauborgne reconoció que si la cinta de la clave era completamente aleatoria, se incrementaría la dificultad criptoanalítica.

¿Por qué es inviolable este cifrado?

De acuerdo con Alfred Menezes en su libro, Handbook of Applied Cryptography (Manual de criptografía aplicada), se puede decir que un sistema es totalmente secreto, o incondicionalmente seguro, cuando el texto cifrado que se observa no proporciona información complementaria acerca de la cadena de texto en claro original. Si suponemos que L es el número de bits en la cadena de texto en claro, i va de 1 a L en las siguientes definiciones:

- pi = el bit n° i en la cadena de texto en claro
- ci = el bit n° i en la cadena de texto cifrado
- ki = el bit n° i en la cadena de clave
- P(pi) = la probabilidad de que pi se ha enviado
- P(pi | ci) = la probabilidad de que pi se ha enviado dado que ci se ha observado

Se puede decir que un sistema es perfectamente secreto cuando P(pi) = P(pi | ci). En los sistemas de cifrado de flujo tradicionales, el método más común de mezclar bits de datos de texto en claro con bits de clave es a través de la operación XOR en los bits correspondientes.
El hecho de que la clave sea completamente aleatoria nos lleva a varias conclusiones. La probabilidad de observar un bit de la clave es la misma que la de cualquier otro bit y el hecho de conocer valores previos de la clave no nos aporta nada sobre valores futuros.
Enunciando otra definición,

- P(ki=1) = P(ki=0) = 1/2 para todos los i.

Dicho de otra forma, el valor de un bit cualquiera es equiprovablemente 0 o 1.

Conocidos dos elementos cualquiera de {pi , ci , ki } podemos determinar el tercero. Asimismo, conociendo uno entre {pi , ci , ki }, se puede escribir un segundo en relación con el tercero.
Por ejemplo, P(ci=1 | ki=0) = P(pi=1); dicho de otra manera, si sabemos que el bit clave es 0, entonces el texto en claro y el texto cifrado deben ser iguales.
Para mostrar que P(pi | ci) = P(pi), primero hay que mostrar que P(ci) = P(ci | pi).

Algunos incovenientes:

- Requiere libretas de un solo uso perfectamente aleatorias.
- La generación e intercambio de las libretas de un solo uso tiene que ser segura, y la libreta tiene que ser al menos tan larga como el mensaje.
- Hace falta un tratamiento cuidadoso para asegurarse de que siempre permanecerán en secreto para cualquier adversario, y es necesario deshacerse de ellas correctamente para evitar cualquier reutilización parcial o completa —de ahí el «un solo uso».

7. Cifradores de flujo y cifradores de Bloque: WEP (RC4) y ECB

Los cifradores de bloque trabajan con clave simétrica sobre grupos de bits de una determinada longitud fija (los bloques). El cifrador de bloque toma en la entrada un bloque de texto plano y produce un bloque de texto cifrado de igual longitud. Esta transformación de texto plano a cifrado se controla mediante una clave secreta. Para el camino inverso (texto cifrado -> texto plano) se opera de la misma manera.

Los cifradores de bloques tienen un inconveniente, hay ciertas aplicaciones que no pueden hacer uso de esta utilidad por estar formadas por un flujo constante de bits. Tenemos por ejemplo un enlace de radio, telefonía, etc. Para estas aplicaciones surgen los cifradores de flujo.

Un cifrador de flujo consiste en un algoritmo que convierte el texto claro en texto cifrado pero trabajando bit a bit. El funcionamiento es similar al de bloque, en la entrada tenemos el flujo de datos. Se genera un "flujo de clave" y la salida es una XOR bit a bit del flujo de datos y el de clave.

Vamos a tratar 2 algoritmos: WEP-RC4 (flujo) y ECB (bloque)

WEP:

El algoritmo WEP (Wired Equivalent Privacy) es el sistema de cifrado incluido en el estandar IEEE 802.11 (estandar que define las redes inalámbricas). WEP está basado en el algoritmo de cifrado RC4, teniendo dos variantes, una que usa clave de 64 bits (40 bits más 24 bits de vector de iniciación) y otra que usa clave de 128 bits (104 bits y 24 bits de vector de iniciación). En 2003 la Wi-Fi Alliance anunció la sustitución de WEP por WPA y en 2004 se ratificó el estándar 802.11i (WPA2) declarando WEP-40 y WEP-104 como inseguros. A pesar de ello todavía hoy en día se sigue utilizando.

Como hemos dicho WEP utiliza el algoritmo de cifrado RC4. El funcionamiento de RC4 es el siguiente:

Se expande una semilla (seed) para generar una secuencia de números pseudoaleatorios de mayor tamaño. Posteriormente, se "unifican" el mensaje y la secuencia pseudoaleatoria mediante una función XOR. El problema que presenta este método es que si se utiliza la misma semilla para cifrar dos mensajes diferentes obtener la clave a partir de los dos textos cifrados sería trivial (por trivial entendemos sencillo desde el punto de vista matemático). En un intento de evitar esto, en WEP se incluyó un vector de iniciación de 24 bits que se modifica regularmente y se concatena a la contraseña para generar la semilla.

El principal defecto de WEP se encuentra precisamente en este vector de iniciación. El tamaño del vector de iniciación es constante (24 bits), esto nos da un número limitado de vectores (2^24=16.777.216). El problema es que la cantidad de tramas que pasan a través de un punto de acceso son muy grandes y es fácil encontrar 2 mensajes con el mismo vector haciendo relativamente fácil obtener la clave. Se puede aumentar el tamaño de la clave, pero esto solo incrementará el tiempo necesario para romper el cifrado.

ECB:

ECB (electronic codebook) es uno de los múltiples algoritmos de cifrado de bloque y uno de los más sencillos.

En ECB la información se divide en bloques y cada uno de ellos es cifrado por separado aplicando una clave común. Este sistema implica un problema, con bloques idénticos tendremos bloques cifrados idénticos pudiendose reconocer así un patrón que facilitará la obtención del mensaje original.

Una mejora de este algoritmo es CBC (cipher-block chaining). Esta mejora consiste en hacer una XOR del bloque antes del cifrado con el anterior cifrado, y posteriormente cifrarlo. Otras evoluciones como CFB y OFB hacen que el cifrado pase a operar como un cifrador de fllujo.

8. Cifrado asimétrico: RSA y Diffie Hellman

La criptografía asimétrica (también conocida como de clave pública) es un sistema que emplea una pareja de claves. Esta pareja de claves pertenecen a la misma persona. Una es de dominio público y cualquiera puede tenerla y la otra es privada. El funcionamiento de este sistema es el siguiente:

El remitente usa la clave pública del destinatario y sólo con la clave privada se podrá descifrar el mensaje. De esta forma se consigue que sólo el destinatario pueda acceder a la información.

De la misma forma si el propietario usa su clave privada para cifrar un mensaje sólo se podrá descifrar con la clave pública. Pero, si todo el mundo puede tener acceso a la clave pública ¿Que utilidad tiene esto? Precisamente por esto es interesante el sistema. Usando tu clave privada estas demostrando tu identidad, pues, en teoria, solo tu eres poseedor de esa clave privada.

La mayor ventaja de este sistema es que la distribución de claves es más fácil y segura que usando clave simétrica.

Sin embargo este sistema tiene varias desventajas:

- Mayor tiempo de proceso en mismas condiciones respecto a clave simétrica.
- Claves más grandes en en sistemas simétricos.
- El mensaje cifrado es más grande que el original.

Por estas razones el principal uso del cifrado asimétrico es solventar los problemas a la hora de intercambiar las claves del cifrado simétrico. De hecho, normalmente lo que se hace es compartir las claves simétricas mediante cifrado asimétrico para posteriormente pasar a un cifrado simétrico más rápido y menos costoso.

Hablaremos de 2 algoritmos: RSA y Diffie Hellman

RSA:
RSA (Rivest, Shamir y Adleman) es un algoritmo de cifrado asimétrico desarrollado en el año 1977 por los anteriormente citados.
Este algoritmo se basa en escoger 2 números primos grandes elegidos de forma aleatoria y mantenidos en secreto. La principal ventaja de este algoritmo desde el punto de vista de seguridad radica en la dificultad a la hora de factorizar números grandes. RSA es seguro hasta la fecha.
La idea del algoritmo es la siguiente:
Tenemos un mensaje M. Empleando un protocolo reversible conocido como patrón de relleno convertimos el mensaje M en un número m menor que otro número dado n.
Se genera el mensaje cifrado c:

$$c \equiv m^e \pmod{n}$$

Se obtiene m descifrando el mensaje cifrado c:

$$m \equiv c^d \pmod{n}$$

Generación de claves:

1. Tomamos 2 números primos p y q. Estos tienen que ser aleatorios e impredecibles. Importante impredecibles, porque un proceso puede ser perfectamente aleatorio, pero si se conoce, se puede predecir los valores, y por tanto, resultaría en una baja seguridad.
2. Calculamos n=p*q.
3. Calculamos ϕ(n). La función ϕ de Euler se define como el número de enteros positivos menores o iguales a n y coprimos con n (dos números son coprimos si no tienen ningún divisor común distinto 1 o -1). La función tiene las siguientes propiedades:
   - ϕ(1)=1.
   - ϕ(p)= p-1 si p es primo.
   - ϕ(p^k)= (p-1)*p^(k-1) si p es primo y k un número natural.
   - ϕ(m*n)=ϕ(m)ϕ(n) si m y n son primos.

   De esta forma nos queda que para nuestro n=p*q:

   ϕ(n)=(p-1)*(q-1)

   4.-Escogemos un número e menor que ϕ(n) y que sea coprimo con ϕ(n).
   Este número será dado a conocer como exponente de la clave pública.
   5.-Obtenemos un número d mediante aritmética modular tal que d = e^(-1) mod(ϕ(n)) o lo que es lo mismo (d*e)-1 tiene que ser divisible por ϕ(n).

De esta forma tenemos la clave pública formada por (n,e) y la privada formada por (n,d).

p = 61	1º nº primo privado
q = 53	2º nº primo privado
n = p*q = 3233	producto p×q
e = 17	exponente público
d = 2753	exponente privado

La clave pública (e, n). La clave privada es (d, n). La función de cifrado e

$$\mathrm{encrypt}(m) = m^e \pmod n = m^{17} \pmod{3233}$$

Donde m es el texto sin cifrar. La función de descifrado es:

$$\mathrm{decrypt}(\mathrm{c}) = c^d \pmod n = c^{2753} \pmod{3233}$$

Donde c es el texto cifrado. Para cifrar el valor del texto sin cifrar 123, nosotros calculamos:

$$\mathrm{encrypt}(123) = 123^{17} \pmod{3233} = 855$$

Para descifrar el valor del texto cifrado, nosotros calculamos:

$$\mathrm{decrypt}(855) = 855^{2753} \pmod{3233} = 123$$

Como hemos mencionado anteriormente, RSA requiere de un esquema de rellenado dado que sino M puede conducirnos a textos cifrados inseguros. Hay múltiples algoritmos de rellenado, entre ellos podemos destacar OAEP (Optimal Asymetric Encription Padding) o SAEP (Simplified Asymetric Encryption Padding).

Diffie Hellman:

El protocolo de cifrado Diffie-Hellman (recibe el nombre de sus creadores) es un sistema de intercambio de claves entre partes, que no han contactado previamente, a través de un canal inseguro y sin autenticación.

Este protocolo se utiliza principalmente para intercambiar claves simétricas de forma segura para posteriormente pasar a utilizar un cifrado simétrico, menos costoso que el asimétrico.

Se parte de la idea de que dos interlocutores pueden generar de forma conjunta una clave sin que esta sea comprometida.

1. Se escoge un número primo p y un generador g que será coprimo de p. Estos 2 números son públicos.
2. Escogemos un número a menor que p y calculamos A = g^a mod p. Enviamos A, p y g al otro interlocutor.
3. El otro interlocutor escoge un número b menor que p y calcula B = g^b mod p. Nos envia B.

Ahora, ambos podemos calcular K= g^(a-b) mod p.

Para nosotros B^a mod p = K y para nuestro interlocutor A^b mod p =K. Usamos K como calve.

Al ser p y g públicos cualquier atacante puede conocerlos. Esto no supone una vulnerabilidad. Aunque el atacante conociera estos dos número y capturase A y B, le resultaría computacionalmente imposible obtener a y/o b y consecuentemente K.

Tomamos p=23 y g=5. Elegimos a=6 y b=15. A=8 y B=19.

(g^a mod p) = 8 → (5^a mod 23) = 8

(g^b mod p) = 19 → (5^b mod 23) = 19

Partiendo de estas ecuaciones obtener a y b es un problema conocido como logaritmo discreto.

a = log discg (g^a mod p) = log disc5 (8)

b = log discg (g^b mod p) = log disc5 (19)

En este caso si podríamos obtenerlos, pues sabiendo que p=23 y que a y b son menores que p solo tendríamos que probar 22 números diferentes. En la realidad se utilizan números primos del orden de 10^200 haciendo computacionalmente imposible la resolución.

Este protocolo es vulnerable a ataque man-in-the-middle. Estos ataques consisten en que un tercero se coloca en medio del canal y hace creer a ambos que es el otro. De esta forma se podría acordar una clave con cada parte y servir de “enlace” entre los dos participantes. Para que este ataque sea funcional se necesita saber que método de cifrado simétrico se va a emplear. Ocultar el algoritmo de cifrado no cumple con el principio de Kerkckhoffs de que la efectividad de un sistema no debe depender de que su diseño permanezca en secreto por lo que conocer el sistema que se va a emplear se hace trivial.

Para evitar esto se puede emplear un protocolo de autenticación de las partes mediante por ejemplo TLS.

9. Funciones Hash criptográficas

Una función hash criptográfica es aquella función hash que se emplea en el área de la criptografía.
Una función Hash es un algoritmo matemático que, con una entrada A, nos da una salida B.
Una función hash puede ser cualquier algoritmo matemático pero tiene que cumplir una serie de propiedades.

1. Para una misma función hash, sea cual sea la longitud de la entrada A la salida B tiene que ser de un tamaño fijo.
2. Para cada A, B tiene que ser única.
3. Tiene que ser rápido y fácil de calcular.
4. No se puede volver a A desde B.
5. No puede presentar colisiones. Esto quiere decir que para dos A diferentes no se puede dar un mismo B. Observando las condiciones 1 y 2 vemos que esto es imposible. La función MD5 nos da como resultado un hash de 128 bits. Esto quiere decir que como máximo hay solo 2^128 textos diferentes, lo cual, es falso. Se hace por tanto muy importante que las colisiones sean mínimas y que encontrarlas sea muy difícil.

En el caso de funciones hash criptográficas se requiere de forma adicional que sean uniformes (para una A elegida aleatoriamente todos los valores hash son equiprobables) y con efecto avalancha (un cambio de un único bit en A supone una B completamente diferente).

Podemos distinguir dos grupos de funciones: las que tienen como objetivo mantener la integridad de los mensajes (detección de modificaciones) y las que tienen como objetivo verificar el origen del mensaje(autenticación).

MD5:

MD5 es un algoritmo de reducción criptográfico de 128 bits ampliamente usado. El algoritmo consta de 5 pasos:

1. Adición de bits. El mensaje es extendido de forma que su longitud menos 448 sea múltiplo de 512. Este paso se realiza aunque la longitud ya sea congruente con 448 modulo 512.
2. Longitud del mensaje. Un entero de 64 bits que represente la longitud del mensaje antes de la adición se concatena al final del resultado del paso anterior. Si la longitud del mensaje es de más de 64 bits se usan sólo los 64 primeros bits. Tras este paso el mensaje es múltiplo exacto de 512. A su vez, el mensaje es múltiplo de 16.
3. Iniciar el búfer MD. Un búfer de cuatro palabras se inicia con unos valores determinados. Cada palabra A,B,C y D tiene 32 bits. Las palabras tienen los siguientes valores:

   A: 01 23 45 67 B: 89 ab cd ef C: fe dc ba 98 D: 76 54 32 10
4. Procesado el mensaje en bloques de 16 palabras. Tomamos cuatro funciones auxiliares cuya entrada es de tres palabra y su salida es una:

$$F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$$
$$G(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$$
$$H(B,C,D) = B \oplus C \oplus D$$
$$I(B,C,D) = C \oplus (B \vee \neg D)$$

$\oplus, \wedge, \vee, \neg$ representan las operaciones XOR, AND, OR y NOT.

5. Tras realizar 16 ciclos en los que se efectuan una serie de operaciones con las funciones anteriores sobre B, C y D obtenemos la salida ABCD.

*Hemos decidido no comentar el código completo del algoritmo MD5 por resultar demasiado extenso.

10. Epic Fails criptográficos

Sony:

En Diciembre de 2010 saltó la noticia, un grupo de hackers conocimos como fail0verflow afirmaba que había conseguido hallar las claves privadas de la PlayStation 3. Con esta clave privada cualquiera podría firmar digitalmente cualquier aplicación y ejecutarla. De esta forma la PlayStation 3 ,videoconsola nacida en 2006, pasaba a ser pirateable.

¿Cómo pudo ser posible que fail0verflow consiguiera hallar estas claves? Sony en su Playstation 3 incorporó el algoritmo criptográfico ECDSA, Elliptic Curve Digital Signature Algorithm. ECDSA es una modificación del algoritmo DSA que utiliza puntos de curvas elípticas. Lo más importante de este hecho es que ECDSA es un algoritmo fuerte que no se encuentra roto.

El problema, como en muchos otros casos,fue la implementación del algoritmo.

$$r = mG$$

$$s = \frac{e + kr}{m}$$

El proceso de firma de ECDSA necesita una pareja de valores r y s. S requiere la generación de un número aleatorio, m, para generar la clave. Es importante que m sea un valor distinto para cada firma. Sin embargo, para Sony m fue un número estático.

Sony evitó la generación de números aleatorios y puso el mismo número en todas sus videoconsolas. De esta forma si se tienen dos firmas generadas con el mismo número aleatorio se obtiene la clave. Este descubrimiento permitió a George Hotz "Geohot" publicar la clave privada de la PlayStation 3 poco después, ya que fail0verflow no hizo pública esta clave.

Debian y SSL:

En mayo de 2008 se descubrió que el generador de números utilizado por Debian OpenSSL era predecible. Esto supuso que las claves generadas con él ya no sean realmente fiables o verdaderamente seguras.

OpenSSL es un cifrado que no se encuentra roto, pero al igual que en el caso de Sony, no generar correctamente un número aleatorio volvió a la implementación del algoritmo vulnerable.

Los sistemas de cifrado basados en clave asimétrica necesitan calcular numeros aleatorios para generar claves de sesión, tanto públicas como privadas.

El fallo fue producido al eliminarse líneas de código que limitan el generador a producir sólo 2^18 claves (solamente 262.144), en vez de poder elegir claves de, por ejemplo 2^1.024 posibilidades. El fallo fue introducido en la versión OpenSSL 0.9.8c-1 de septiembre de 2006 de esta forma desde 2006 OpenSSL de Debian era vulnerable.

La líneas que se eliminaron hacía uso de memoria no iniciada. Esto sería un fallo en cualquier aplicación, en la primera clase de programación con punteros siempre te dicen que trabajar con un puntero a memoria no iniciada es un grave error. El problema viene dado en que esa memoria no iniciada se utilizaba para conseguir "basura" o datos aleatorios aumentando de ese modo la entropía.

De esta forma todas las claves creadas entre 2006 y 2008 en Debian para usarse en comunicaciones asimétricas pasan a ser vulnerables.

Adobe:

Adobe es una de las grandes empresas de software presente en internet. Recientemente Adobe reconoció el robo de código fuente de varios de sus productos y la información personal de casi 3 millones de usuarios y clientes.

Los datos que se robaron alcanzaron casi los 10 Gb y supusieron 130 millones de credenciales robadas. La obtención de esas 130 millones de credenciales viene dado por la mala politica de Adobe a la hora de guardarlas.

Adobe no optó por almacenar contraseñas en forma de hash con una función segura como es la práctica recomendada.

En su lugar Adobe eligió usar TripleDES como algoritmo de cifrado. TripleDES es una evolución del algoritmo DES, del que hablamos previamente. TripleDES es el resultado de encadenar tres veces el algoritmo DES en cada bloque de datos. Sin embargo TripleDES cuenta con ataques conocidos que tienden a reducir su seguridad.Además, Adobe uso TripleDES con método ECB, que tiene como problema que dos textos claros idénticos dan el mismo resultado cifrados.

Esto permite muchas ventajas al atacante, tenemos que el mismo texto claro siempre va a tener el mismo resultado. Si conseguimos una contraseña podemos ver si esta se encuentra repetida en el resto sin ningún coste. En internet se encuentran disponibles listas con las 100 contraseñas más usadas en adobe.

http://stricture-group.com/files/adobe-top100.txt

11. Fuentes

- http://www.rae.es/recursos/diccionarios/drae
- https://es.wikipedia.org/wiki/Esc%C3%ADtala
- https://es.wikipedia.org/wiki/Enigma_(m%C3%A1quina)
- http://www.lcc.uma.es/~pastrana/EP/trabajos/57.pdf
- http://www.ecured.cu/index.php/Cifrado#Usos_del_cifrado
- http://www.seguridadenlared.org/es/index25esp.html
- http://laurel.datsi.fi.upm.es/proyectos/teldatsi/teldatsi/protocolos_de_comunicaciones/algritmos_de_cifrado
- https://en.wikipedia.org/wiki/Data_Encryption_Standard
- http://www.ehowenespanol.com/funciona-aes-info_215975/
- https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
- http://www.mils.com/uploads/media/Cifrado_One_Time_Pad.pdf
- https://en.wikipedia.org/wiki/One-time_pad
- https://en.wikipedia.org/wiki/Block_cipher
- https://en.wikipedia.org/wiki/Stream_cipher
- https://en.wikipedia.org/wiki/Wired_Equivalent_Privacy
- https://en.wikipedia.org/wiki/RC4
- https://es.wikipedia.org/wiki/Criptograf%C3%ADa_asim%C3%A9trica
- https://en.wikipedia.org/wiki/RSA_(cryptosystem)
- https://es.wikipedia.org/wiki/Funci%C3%B3n_%CF%86_de_Euler
- https://es.wikipedia.org/wiki/N%C3%BAmeros_primos_entre_s%C3%AD
- https://es.wikipedia.org/wiki/Diffie-Hellman
- http://www.javiercampos.es/blog/2011/07/22/el-algoritmo-de-diffie-hellman/
- http://foro.elhacker.net/criptografia/funciones_de_hash-t100025.0.html
- https://es.wikipedia.org/wiki/Funci%C3%B3n_hash_criptogr%C3%A1fica
- https://en.wikipedia.org/wiki/MD5
- http://www.engadget.com/2010/12/29/hackers-obtain-ps3-private-cryptography-key-due-to-epic-programm/
- http://www.tomsguide.com/us/PlayStation-Console-Private-Cryptography-Key-fail0verflow-linux,news-9542.html
- http://www.theguardian.com/technology/gamesblog/2011/jan/07/playstation-3-hack-ps3
- https://en.wikipedia.org/wiki/Elliptic_Curve_DSA
- http://unaaldia.hispasec.com/2013/11/adobe-la-tormenta-despues-de-la-tormenta.html
- http://www.securitybydefault.com/2013/11/primeras-impresiones-acerca-del-top-100.html
- Libro: Information Security Principles and Practice Mark Stamp Segunda Edición