***Historia de la criptografía(clase 1)***

* 1900 AC En el antiguo Egipto se usaron símbolos que no eran los normales.
* 1500 AC Los fenicios diseñaron un alfabeto.
* 1000 AC Se usaron otros símbolos distintos a los normales en la antigua Mesopotamia.
* 600 AC En Palestina se cifran textos usando un algoritmo simple de sustitución monoalfabética Atbash.
* 500 AC Los espartanos cifran mensajes utilizando Scytale (escítala).
* 400 AC El Kamasutra describe un algoritmo de cifrado por sustitución monoalfabética.
* 100-44 AC Julio César inventa un código para cifrar sus mensajes (el Código AC César). Este es el algoritmo de sustitución monoalfabética más conocido.
* 500-1400 DC La "edad oscura de la criptografía" empieza en Europa, se considera como magia negra: Durante este florece en Persia.
* 1400 DC En Italia se produce un boom de la criptografía debido un alto desarrollo de la vida diplomática.
* 1795 DC Thomas Jefferson diseña el primer dispositivo de cifrado cilíndrico, conocido como la "rueda de Jefferson".
* 1917 DC El americano Gilbert S. Vernam, empleado de AT&T, desarrolla la cinta aleatoria de un sólo uso, el único sistema criptográfico seguro.
* 1918 DC Arthur Scherbius y Richard Ritter inventan la primera Enigma. Al mismo tiempo, la máquina de rotores es inventada y patentada por Alexander Koch (Países Bajos) y Arvid Damm (Suecia).
* 1940-1945 DC - Alan Turing rompe Enigma con la idea de la Bomba de Turing que concibió basándose en el trabajo de Marian Rejewski.
* 1948-1949 DC Claude Shannon establece la bases matemáticas de la teoría de la información y public "Communication Theory of Secrecy Systems" en donde expone un algoritmo de cifrado teóricamente irrompible que debe satisfacer los requisitos de la cinta aleatoria de un solo uso.

***Elementos teóricos de la criptografía***

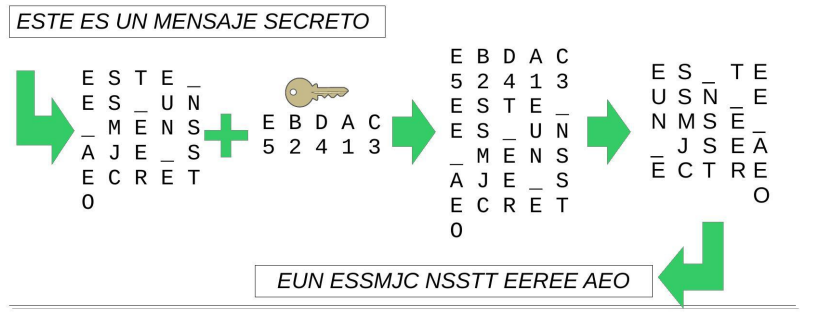
**D(k, E(k, m) ) = m**

* m representa el conjunto de todos los mensajes sin cifrar (lo que se denomina texto claro, o plaintext) que pueden ser enviados.
* C representa el conjunto de todos los posibles mensajes cifrados, o criptogramas. (Es el resultado de e, k, m)
* k representa el conjunto de claves que se pueden emplear en el criptosistema.
* E es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de M para obtener un elemento de C. Existe una transformación diferente E para cada valor posible de la clave k.
* D es el conjunto de transformaciones de descifrado, análogo a E. (Vendria a ser la función inversa)
* (Para que se entienda, para cifrar utilizo la función e, k, m; y para desifrar utilizo d, k, e, k, m)

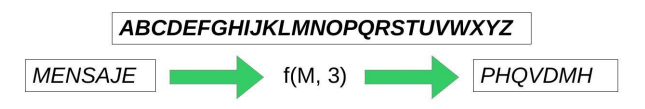


***Criptografía clásica***

Los **cifradores** **por transposición** utilizan la técnica de permutación de forma que los caracteres del texto se reordenan mediante un algoritmo específico. (osea los caracteres siempre van a ser los mismo, solo que se ordenan de forma diferente)



Los **cifradores** **por sustitución** utilizan la técnica de modificación de cada carácter del texto en claro por otro correspondiente al alfabeto de cifrado. Si el alfabeto de cifrado es el mismo que el del mensaje o bien el único, hablamos entonces de cifradores monoalfabéticos; es decir, existe un único alfabeto en la operación de transformación del mensaje en criptograma. Por el contrario, si en dicha operación intervienen más de un alfabeto, se dice que el cifrado es polialfabético. Por ejemplo, el cifrado del Cesar (monoalfabético)



Los **cifradores por sustitución polialfabética** utilizan diferentes caracteres para el reemplazo de un mismo carácter de origen.

Ejemplo: Cifrado de Vigenere, se basa en una matriz cuyos filas y columnas son alfabetos en orden.

El texto a cifrar es: TEXTO DE PRUEBA / Se ha cifrado con la clave: ABCD

Clave expandida(es una clave que utilizo en el proceso de cifrado pero no es la clave elegida

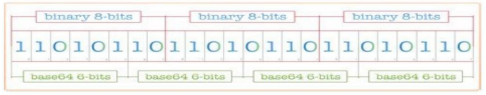
por el usuario, en este caso es expansión de clave por repeticion): ABCDABCDABCDABC

El texto cifrado es: TFZWO EGSRV GEA

***Referencia – Esquiemas de codificación***

Binario: Representa la información a nivel de bits con 0 y 1

Hexadecimal: Representa la información con 16 caracteres con representación grafica(0123456789ABCDEF), asignando cuatro bits a cada carácter

Base64: Representa la información con 64 caracteres representables (ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxy0123456789+/), indicando con un carácter especial (=) los octetos faltantes del ultimo bloque.

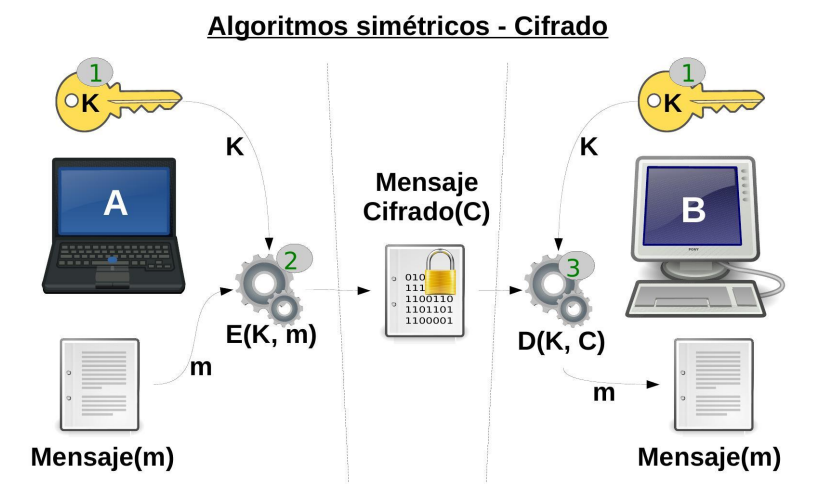
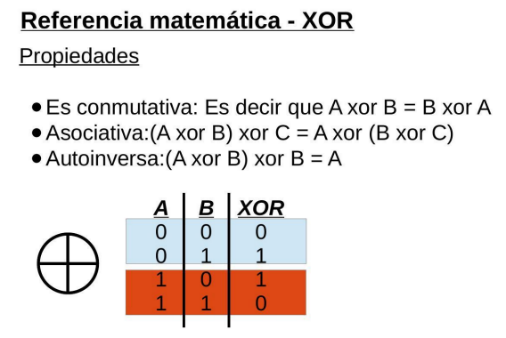


***Algoritmos simétricos***

Un sistema de cifrado simétrico es un tipo de cifrado que usa una misma clave para cifrar y para descifrar. Las dos partes que se comunican mediante el cifrado simétrico deben estar de acuerdo en la clave a usar de antemano. Una vez de acuerdo, el remitente cifra un mensaje usando la clave, lo envía al destinatario, y éste lo descifra usando la misma clave.

Ventajas: Sencillez de implementación, Robustez, Velocidad de cifrado, Longitud del mensaje

Desventajas: La clave debe ser compartida previamente con seguridad, La comunicación entre múltiples actores requiere numerosas claves

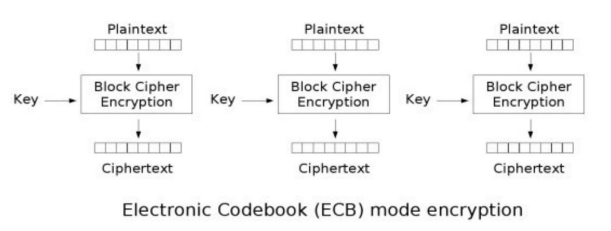


***Algoritmos simétricos de Bloque (Ejemplos)***

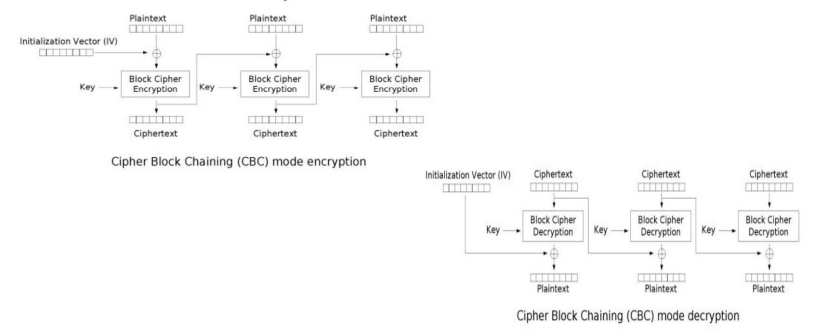
* DES-LUCIFER (1976, Data Encryption Standard)
* 3DES (1998, Triple Data Encryption Standard, NIST)
* **AES-Rijndael** (2001, Advanced Encryption Standard, NIST)
* **Serpent** (1998) (finalista pero gano el AES)
* **Twofish** (finalista pero gano el AES)
* **RC6** (1998, Rivest Cipher 6) (finalista pero gano el AES)
* **MARS** (1998, IBM) (finalista pero gano el AES)
* GOST (1994, Magma URSS)
* CAMELLIA (2000, NTT y Mitsubishi Electric)
* IDEA (1991, International Data Encription Algorithm)
* Blowfish (1993, diseñado por Bruce Schneier)
* RC5 (1994, Rivest Cipher 5)

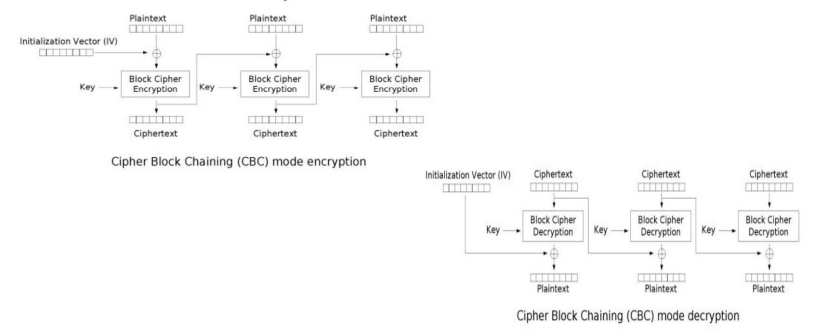
***MODOS DE CIFRADO DE BLOQUE***

***ECB:*** Electronic codebook, en este método el mensaje se fracciona en partes y cada una es cifrada de manera independiente. (aca pasa como en los cifradores monoalfabéticos, que si se repite un carácter de entrada, el cifrado de salida va a ser el mismo, como en el ejemplo del mensaje que se repite la h como carácter de salida).

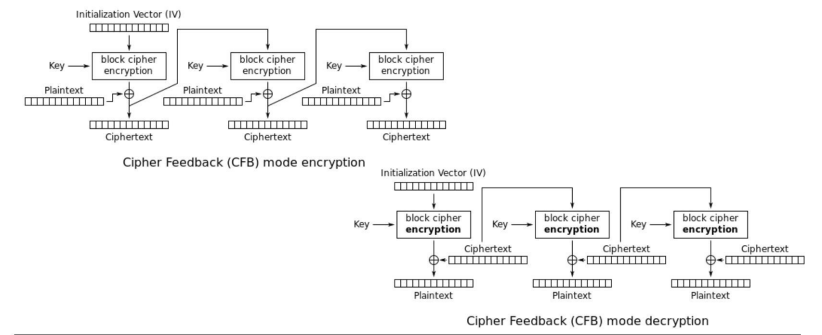


***CBC:*** Cipher block chaining, en este método el mensaje se fracciona en partes y se realiza un XOR con el bloque previo antes de cifrar cada parte (aca se va distorsionando el mensaje de tal forma que por cada bloque de entrada, por mas que sea igual, va haber otro bloque de salida, ya que como principio toma el bloque distorcionado que viene del anterior). El vector de inicialización tiene que tener el mismo tamaño que el bloque de entrada; debería estar compuesta por información aleatoria y esta información no debería ser utilizada en múltiples procesos.

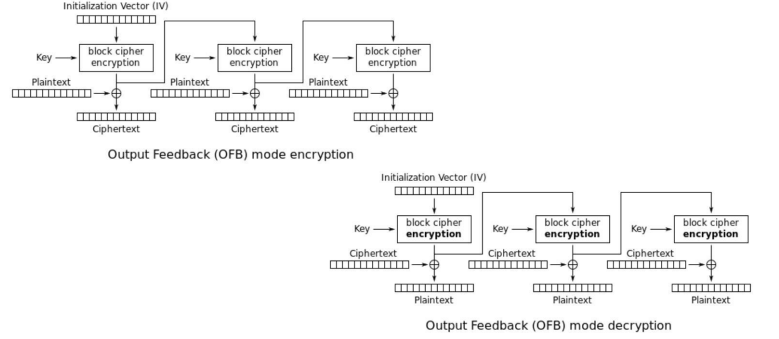




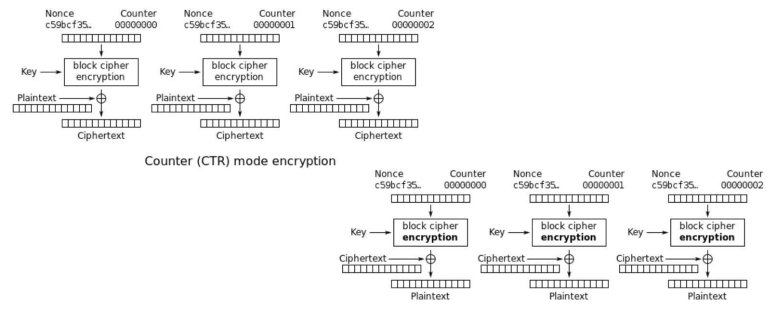
***CFB:*** Cipher Feedback, en este método el mensaje se fracciona en partes, se cifra un vector de inicialización y al resultado se le realiza un XOR con el bloque del mensaje. Los bloques posteriores utilizan como entrada el texto cifrado para reemplazar al vector de inicialización.



***OFB:*** Output Feedback, este método opera de manera similar a CFB con la diferencia que el bloque a ser utilizado como entrada del siguiente proceso es tomado de la salida del algoritmo justo antes de realizar el XOR.



***CTR:*** Modo de Counter o contador, en este modo de operación (al igual que en OFB) se utiliza un "nonce" equivalente al IV anterior, que es alterado por un contador incrementado en cada bloque de datos, para obtener un valor que luego sera operado con el bloque de datos usando XOR .



Otros modos de cifrado de bloques:

Existen diversos modos y algunos de ellos incorporan autenticación a la confidencialidad.

PCBC, CCM, CWC, EAX, GCM (Galois Counter Mode), PCFB, XCBC

***Padding o esquema de relleno***(esto pasa cuando divido el bloque y el utlimo bloque no llega a la cantidad determinada por ese mismo)

Los algoritmos simétricos de bloque requieren que el mensaje sea fragmentado en partes de una longitud fija; esto plantea el problema de que el mensaje en su totalidad o su ultimo bloque podría ser de longitud menor a la requerida, en este caso se recurre a los métodos de padding para resolver el problema.

Algunos de ellos son:

* Bit padding (RFC1321, ISO/EC 9797-1): operando a nivel de bits adiciona 1 y posteriormente N ceros



* ISO/IEC 7816-4: identico a bit padding pero operando a nivel de bytes, agregando el valor 80 y posteriormente N cantidad de 0 hasta completar



* PKCS#7 (RFC2315, Sec 10.3): operando a nivel de bytes N cantidad de bytes idénticos cuyo valor es la cantidad de bytes agregados



* ISO 10126: operando a nivel de bytes agrega N cantidad de bytes aleatorios hasta el ante último, luego ingresa el ultimo byte que contendrá la cantidad de bytes agregados.

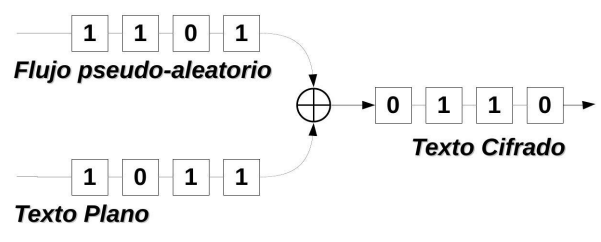


* ANSI X.923: operando a nivel de bytes adiciona N cantidad de 00 hasta el anteúltimo byte, luego ingresa el ultimo que contendrá la cantidad de bytes agregados.



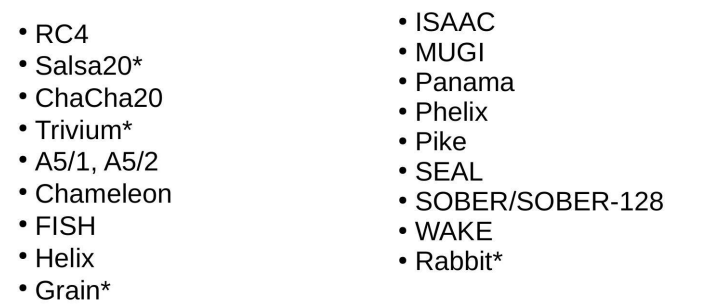
***MODO DE CIFRADO DE FLUJO***

* En 1917, J. Mauborgne y G. Vernam inventaron un criptosistema perfecto según el criterio de Shannon. Dicho sistema consistía en emplear una secuencia aleatoria de igual longitud que el mensaje, que se usaría una única vez (One Time Pad), combinándola mediante alguna función simple y reversible como el or exclusivo(XOR) con el texto en claro carácter a carácter. Este método presenta el grave inconveniente ue que la clave es tan larga como el propio mensaje, y si disponemos de un canal seguro para enviar la clave, ¿por que no emplearlo para transmitir el mensaje directamente?
* Se utiliza una función generadora de bits pseudo-aleatorios a fin de obtener un flujo de bits que pueda ser procesado con los bits del mensaje mediante una operación básica (XOR). (se puede usar parámetros, los cuales son llamados semillas)

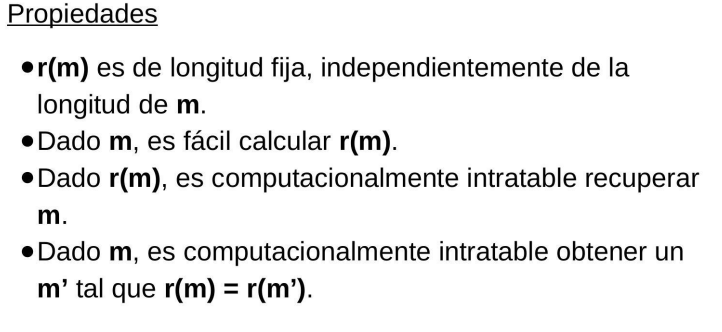
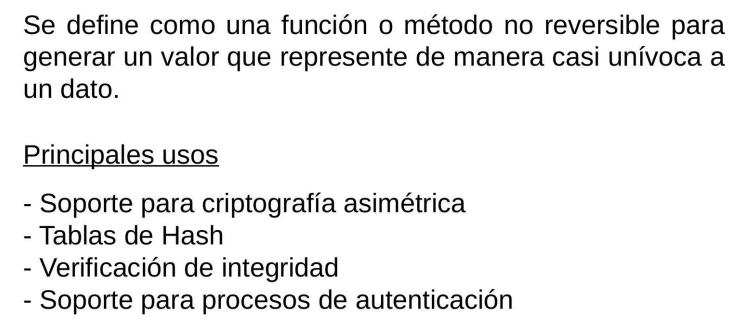


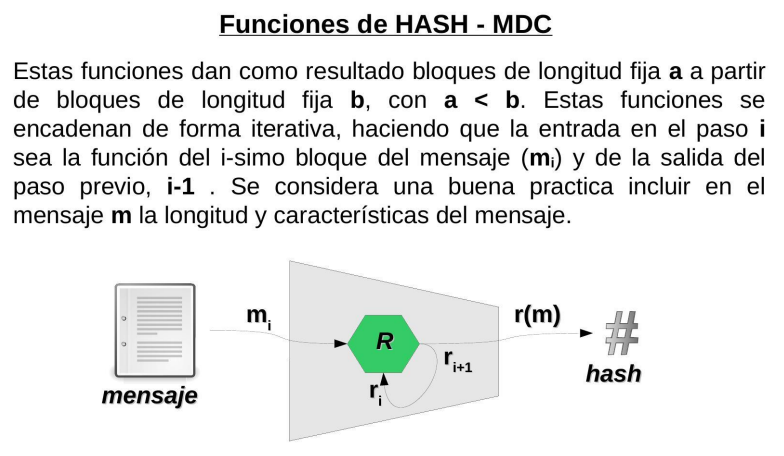
**Secuencias criptográficamente aleatorias:** Para que una secuencia pseudoaleatoria sea criptográficamente aleatoria, ha de cumplir la propiedad de ser impredecible. Esto quiere decir que debe ser computacionalmente intratable el problema de averiguar el siguiente número de la secuencia, teniendo total conocimiento acerca de todos los números anteriores y del algoritmo de generación empleado.

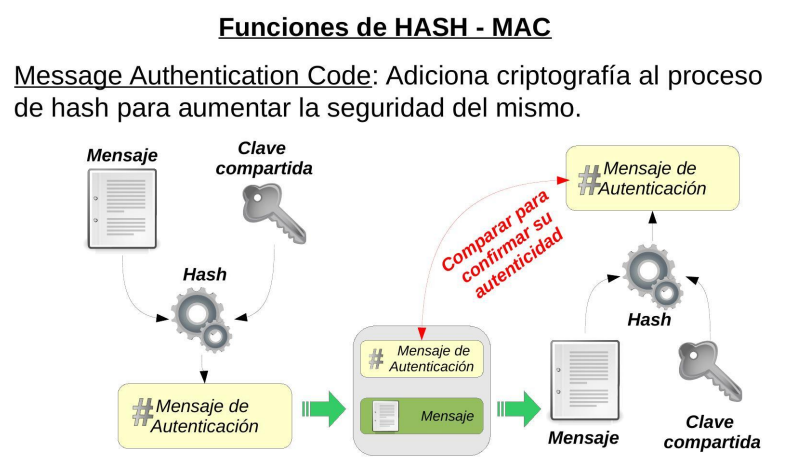
Una **función generadora de bits pseudo aleatoria** es la que permite obtener secuencias criptográficamente aleatorias.

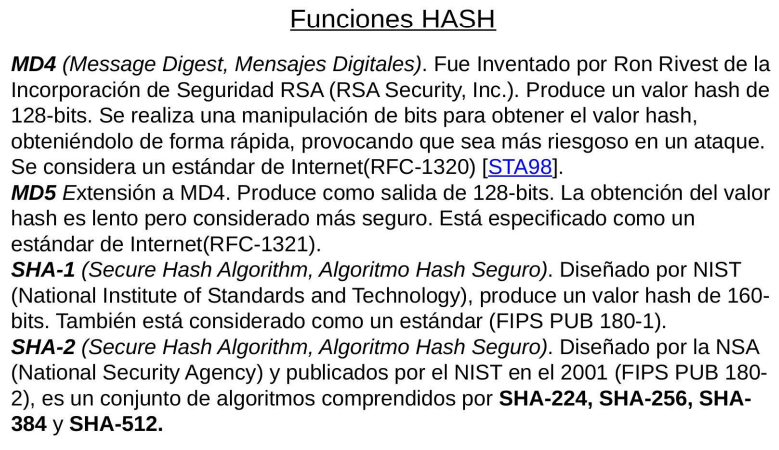
Ejemplo de algunos de los actuales cifradores de flujo:  


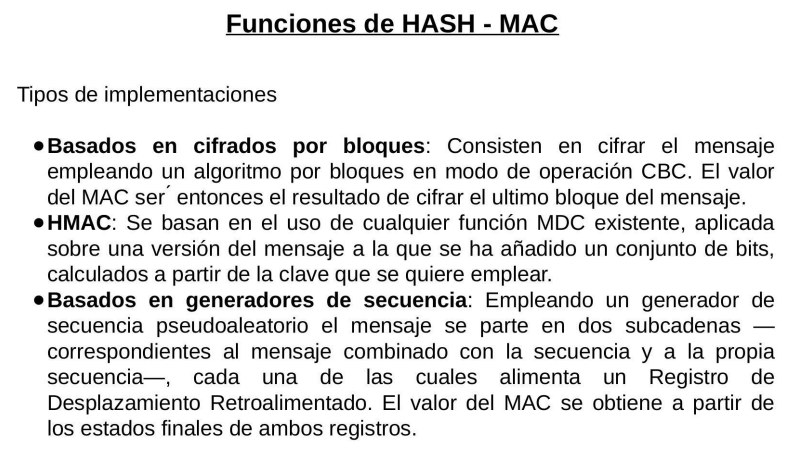
***Algoritmos de hash***

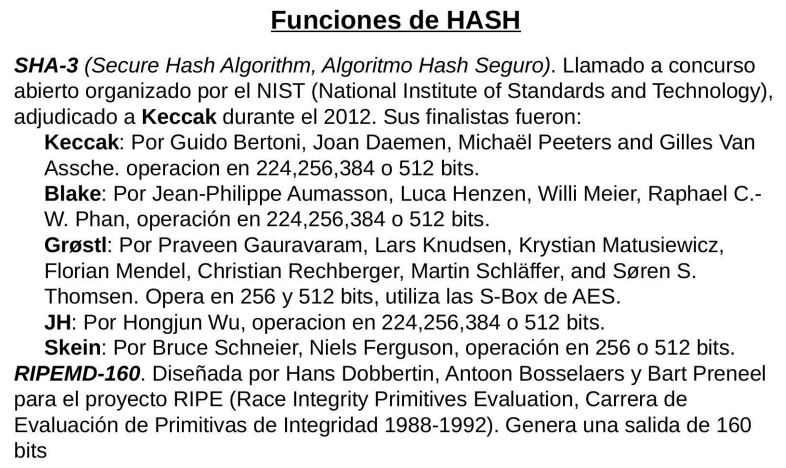
 









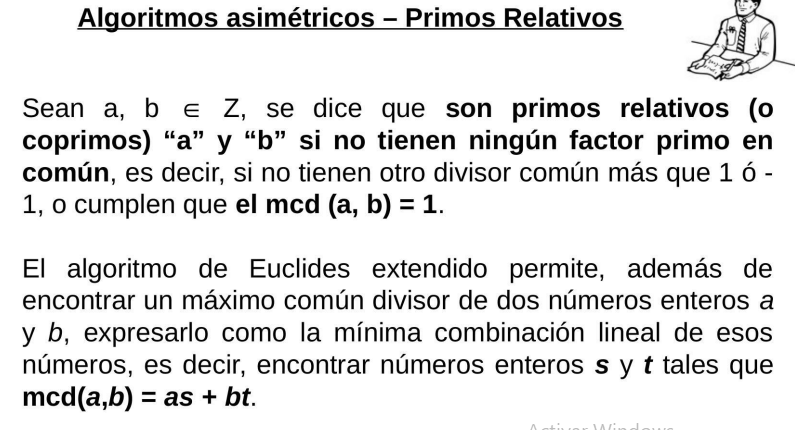


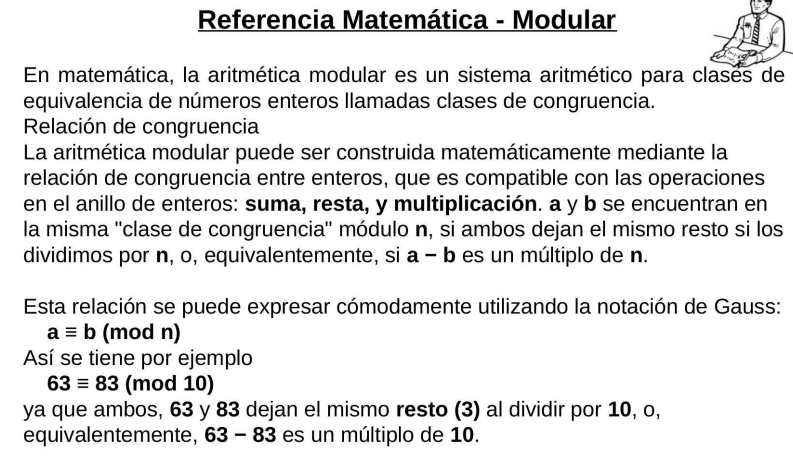
***Funciones de Derivación de Claves (clase 2)***

Conocidas como KDF (Key Derivation Function, viene de la familia de los hash) son funciones no reversibles que tienen el objetivo de generar una o mas claves en base a un valor maestro o clave inicial secretos, mas un conjunto de parámetros que configuran el comportamiento de la función afectando el resultado.

Normalmente se basan en funciones pseudo-aleatorias, funciones de hash con múltiples iteraciones y procesos de inclusión de 'Salt'. Se han originado para evitar a ataques de diccionario y tablas de arcoiris.

* PBKDF2 - (2000) Password-Based Key Derivation Function 2 - RFC2898 y (PKCS#5, NIST SP 800-132)
* bcrypt - (1999) Basado en el algoritmo de Blowfish
* scrypt - (2012) Basado en PBKDF2\_HMAC\_SHA256
* HKDF - (2010) HMAC-based Extract-and-Expand Key Derivation Funcion - RFC5869
* Argon2 - (2015) Por la Universidad de Luxemburgo - RFC9106





***Algoritmos asimétricos***

Introducido por Whitfield Diffie y Martin Hellman a mediados de los años 70. El sistema de cifrado de clave pública usa un par de claves para el envio de mensajes. Las dos claves pertenecen a la misma persona a la que se ha enviado el mensaje. Una clave es pública y se puede entregar a cualquier persona. La otra clave es privada y el propietario debe guardarla para que nadie tenga acceso a ella. El remitente usa la clave pública del destinatario para cifrar el mensaje, y una vez cifrado, sólo la clave privada del destinatario podrá descifrar este mensaje.

***Algoritmos asimétricos – Diffie-Hellman* (no es de cifrado, es solo para establecer una contraseña común)**

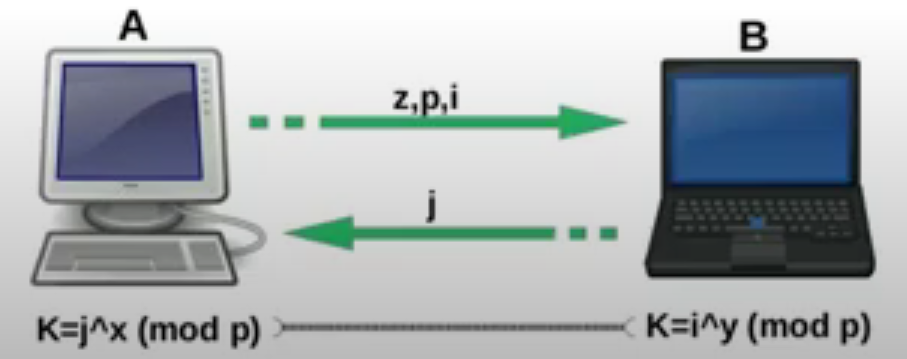
Este algoritmo nos permite compartir un mensaje cifrado entre dos actores que no han tenido contacto previo; por esta razón suele utilizarse para acordar una clave de cifrado a través de un canal inseguro y sin autenticación.

Sean **A** y **B** los interlocutores en cuestión. En primer lugar, se calcula un número primo p y un generador **z** de **Z** •, con **2 ≤ z S p - 2**. Esta información es pública y p conocida por ambos. El algoritmo queda como sigue:

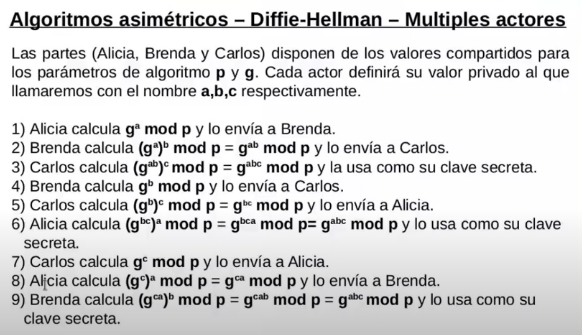
1. **A** escoge un número aleatorio x, comprendido entre 1 y P - 2 y envía a B el valor

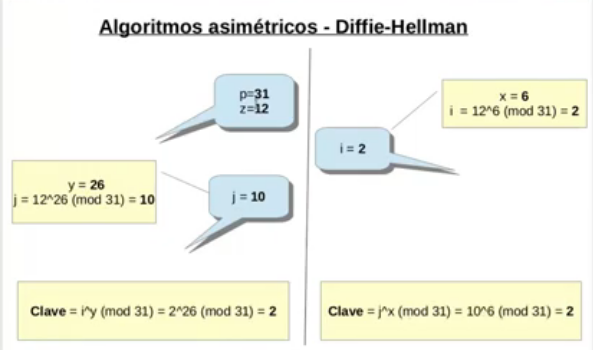
I = zx（mod p）

1. **B** escoge un número aleatorio y, análogamente al paso anterior, y envía a **A** el valor

****j=zy (mod p）

1. **B** recoge **i** y calcula **K** **= iy (mod p) = (zx mod p)y (mod p)**.
2. **A** recoge **j** y calcula **K = jx (mod p) = (zy mod p)x (mod p).**
3. Conclusión **K= zxy (mod p)**

** Ejemplo de solo dos actores:**

****

**Ejemplo de algoritmos asimetricos:**

Diffie-Hellman, RSA, ElGamal, DSA, ECC(criptografía de curva elíptica(ECDH, ECDSA…))

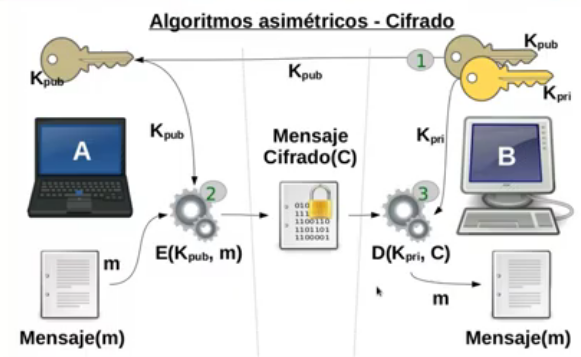
**Algoritmos asimétricos - Cifrado**

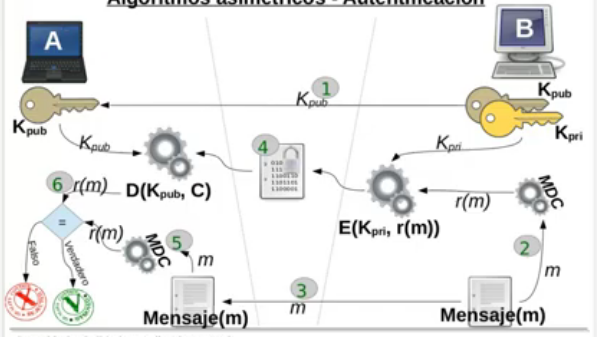
*Ventajas*

No requiere confidencialidad en la distribución de clave / La misma clave puede ser utilizada por múltiples actores en la comunicación / Permite autentificar mensajes

*Desventajas*

Velocidad de cifrado o descifrado / Longitud de mensaje limitado / Tamaño del mensaje cifrado es mayor / Se requieren claves de gran extensión

***Dispone de las siguientes operaciones***

1. Generación de claves (en la imagen)
2. Cifrado (en la imagen)
3. Descifrado (en la imagen)
4. Firma

Algunos algoritmos asimétricos permiten que se

pueda autenticar un mensaje para garantizar su

integridad. en este caso la clave que se emplea

para cifrar es la clave privada, justo al revés que

para la simple codificación de mensajes.

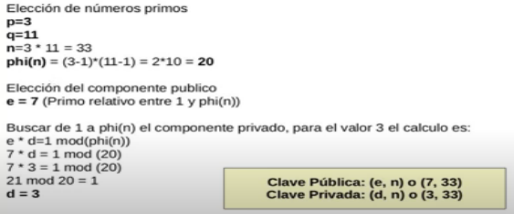
1. Verificación de firma

**RSA**

* Su nombre deriva de Ronald Rivest, Adi Shamir y Leonard Adleman
* Fue publicado en 1977
* Estuvo bajo patente de los Laboratorios RSA hasta el 20 de septiembre de 2000
* Se basa en la dificultad para factorizar grandes números

EXPLICACION:

Se eligen aleatoriamente dos números primos grandes, **p** y **q**. Después se calcula el producto **n = pq**. Escogeremos ahora un número e primo relativo con **(P - 1)(q - 1). (e, n)** sera la clave publica. Nótese que e debe tener inversa **modulo (P - 1)(q - 1)**, por lo que existirá un número d tal que

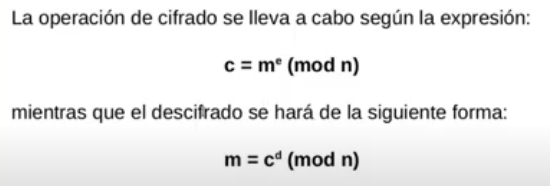
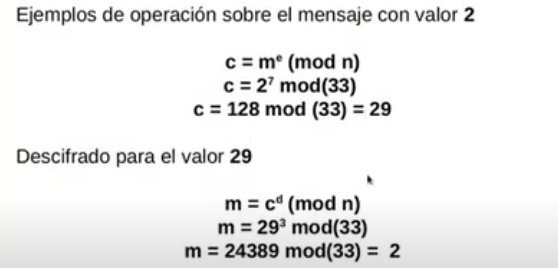


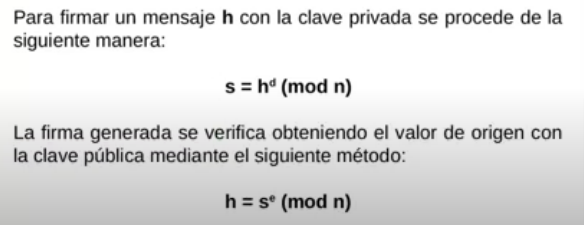
**de = 1 (mod (p - 1)(q - 1))**

**de = 1 (mod mcm(p - 1, q - 1))**

es decir, que d es la inversa de e modulo (p-1)(q -1). (d, n) sera la clave privada. Esta inversa puede calcularse fácilmente empleando el Algoritmo Extendido de Euclides. Nótese que si desconocemos los factores de **n**, este cálculo resulta prácticamente imposible.

ALGORITMO PARA CIFRAR:



FIRMAR DIGITALES DE RSA

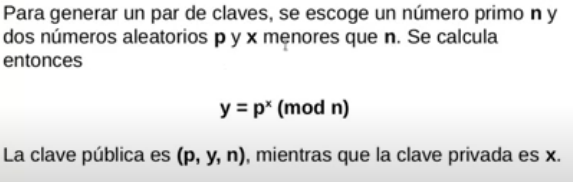
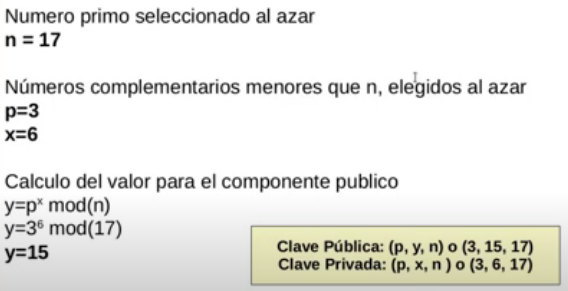
**ELGAMAL**

Se basa en el problema de los logaritmos discretos. Se conoce como logaritmo discreto de x en base a módulo n a resolver la ecuación **x=ay mod** **n** donde **x,n** y **a** son constantes e **y** es la incógnita. A partir de ahora notamos esta situación como:

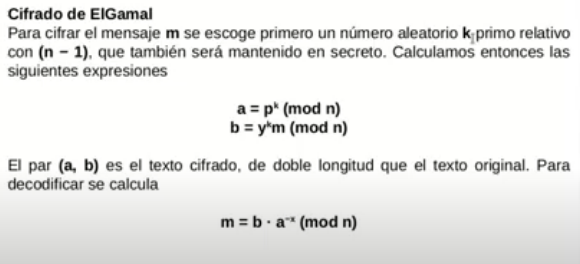
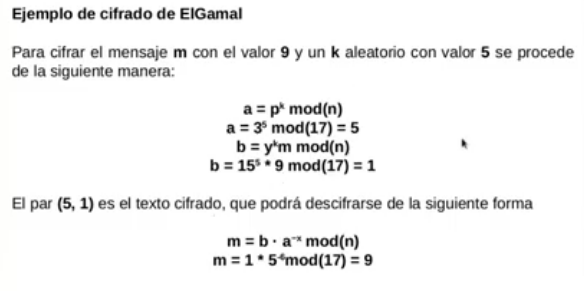
**\* y=log disca(x)**

El hecho de aplicar aritmética modular hace el problema de hallar y irresoluble en un tiempo razonable

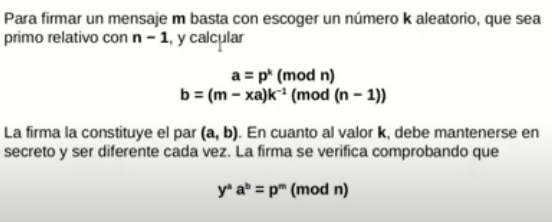
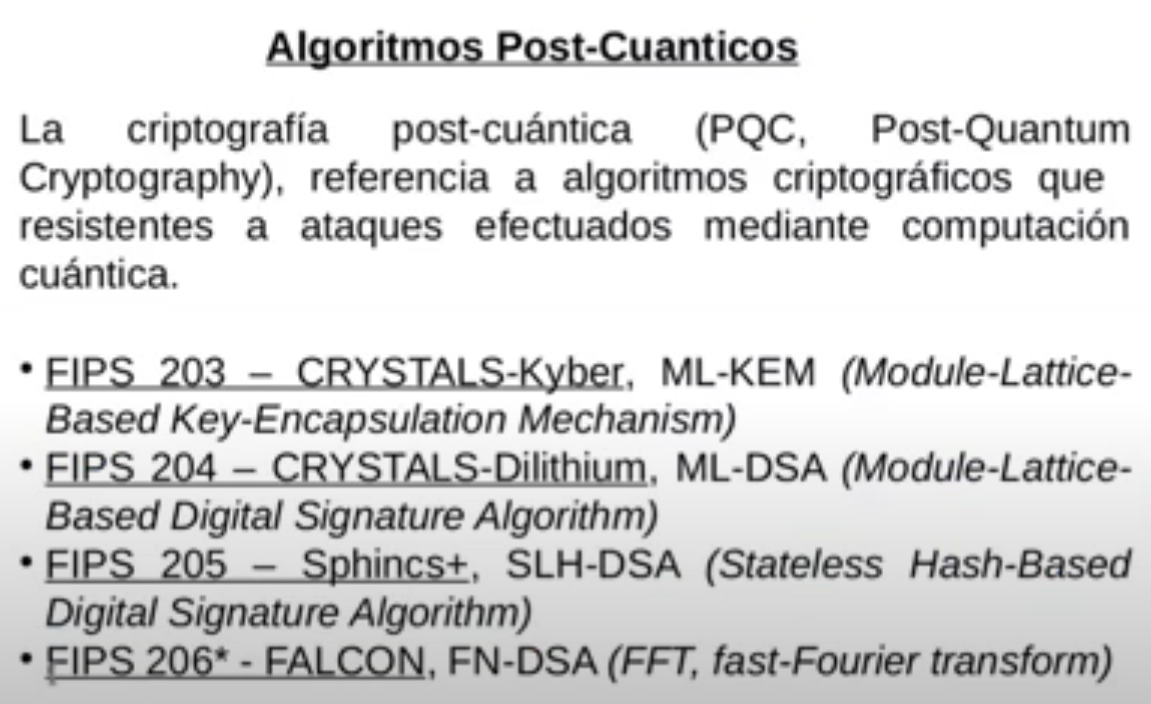
ALGORITMO PARA GENERAR CLAVE: EJEMPLO:

ALGORITMO PARA CIFRAR: EJEMPLO:

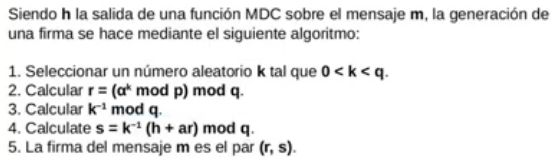
 

FIRMAS DIGITALES: TEMA APARTE:

**DSA**

El algoritmo DSA (Digital Signature Algorithm) es una parte el estándar de firma digital DSS (Digital Signature Standard). definido en el FIPS-186. Este algoritmo, propuesto por el NIST, data de 1991, es una variante del método asimétrico de ElGamal.

GENERACION DE FIRMA: VERIFICACION DE FIRMA

*CURVAS ELIPTICAS*

Una curva elíptica es un tipo especial de curva algebraica definida por una ecuación de la forma: y2 = x3 + ax + b donde a y b son constantes y los números **x** e **y** son variables que pertenecen a un campo definido

Ejemplos de uso criptográfico: P-256(secp256r1), p-384(secp384r1), curve25519, secp256k1

**Algoritmos públicos y privados**

Los algoritmos públicos son aquellos cuya definición y funcionamiento se ponen a disposición pública, permitiendo que cualquier persona o entidad acceda al mismo para su evaluación o investigación. En contraparte los privados son aquellos cuyo funcionamiento interno es desconocido; en el ámbito de la criptografía estos últimos son considerados menos confiables.

Nota: Las patentes pueden condicionar el uso de ambos tipos de algoritmos

***Http vs Https(clase3)***

***http***: Hyper Text Transfer Protocol, protocolo para transmisión de información en plano, sin cifrado. Su puerto por defecto es el número 80.

***https***: Hyper Text Transfer Protocol Secure, protocolo para transmisión información cifrada mediante SSL o TLS. Su puerto por defecto es el numero 443.

***SSL(Secure Sokets layer)***

Es un protocolo que proporciona privacidad e integridad entre dos aplicaciones. El sistema SSL es independiente del protocolo utilizado; esto significa que puede asegurar transacciones realizadas en la Web a través del protocolo HTTP y también conexiones a través de los protocolos FTP, POP e IMAP. SSL actúa como una capa adicional que permite garantizarla seguridad de los datos y que se ubica entre la capa de la aplicación y la capa de transporte (por ejemplo, el protocolo TCP) Originado en Netscape su Version 3.0 data de 1996, definida en la RFC-6101 por la Internet Engineering Task Force (IETF).

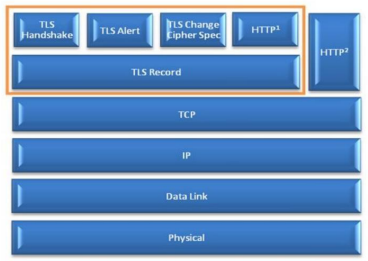
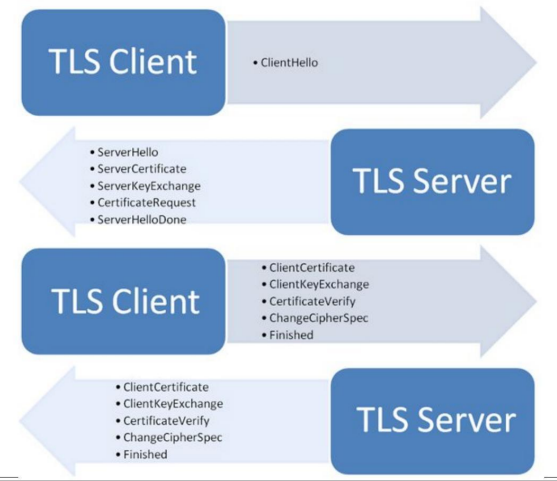
Los datos que circulan en un sentido y otro entre el cliente y el servidor se cifran mediante un algoritmo simétrico como DES o RC4. Un algoritmo de clave pública -generalmente RSA- se utiliza para el intercambio de las claves de cifrado y para las firmas digitales. El algoritmo utiliza la clave pública en el certificado digital del servidor. Con el certificado digital del servidor, el cliente también puede verificar la identidad del servidor. Las versiones 1 y 2 del protocolo SSL sólo proporcionan autenticación de servidor. La versión 3 agrega la autenticación del cliente, utilizando los certificados digitales de cliente y de servidor.

*Fases*

1. Establecimiento de la conexión y negociación de los algoritmos criptográficos que van a usarse en la comunicación, a partir del conjunto de algoritmos soportados por cada uno de los interlocutores.
2. Intercambio de claves, empleando algún mecanismo de clave pública y autentificación de los interlocutores a partir de sus certificados digitales.
3. Cifrado simétrico de trafico.

***TLS (Transport Layer Security)***

TLS (Transport Layer Security) es una evolución del protocolo SSL (Secure Sockets Layer), es un protocolo mediante el cual se establece una conexión segura por medio de un canal cifrado entre el cliente y servidor.

******

*Características adicionales:*

* Incompatible con SSL v3.0
* Uso de funciones MAC en lugar de funciones MDC únicamente
* Numeración secuencial de todos los campos que componen la comunicación, e incorporación de esta información al cálculo de los MAC.
* Protección frente a ataques que intentan forzar el empleo de versiones antiguas —menos seguras— del protocolo o cifrados más débiles.
* El mensaje que finaliza la fase de establecimiento de la conexión incorpora una signatura (hash) de todos los datos intercambiados por ambos interlocutores.

Algoritmos utilizados

* Cifrado Asimetrico: RSA, Diffie-Hellman(DHE), Curva Eliptica (ECDHE), DSA (Digital Signature Algorithm).
* Cifrado Simetrico: RC2, RC4, IDEA (International Data Encryption Algorithm), DES (Data Encryption Standard), Triple DES o AES (Advanced Encryption Standard)
* Funciones de Hash: MD5 o de la familia SHA.

**RFC 2246 (1999) - The TLS Protocol Version 1.0 -->** Primer versión del protocolo

**RFC 4346 (2006) - The TLS Protocol Version 1.1 -->** Se destacan mejoras en el manejo del padding y protecciones a ataques por CBC usando IV explícitos

**RFC 5246 (2008) / RFC 6176 (2011) - The TLS Protocol Version 1.2 -->** Incorporación de SHA-256 para función de PRF (pseudo-aleatoria) y cierre de mensajes, SHA-1 para firma digital, entre otros.

**RFC 8446 (2018) - The TLS Protocol Version 1.3 -->** Remocion de cifrados obsoletos, optimización del protocolo (handshaking con Zero Round-Trip Time, etc)

***Firma electrónica***

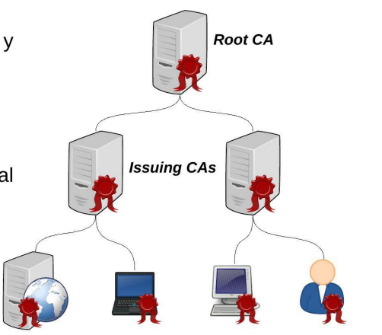
Se entiende por firma electrónica al conjunto de datos electrónicos integrados, ligados o asociados de manera lógica a otros datos electrónicos, utilizado por el signatario como su medio de identificación, que carezca de alguno de los requisitos legales para ser considerada firma digital. En caso de ser desconocida la firma electrónica corresponde a quien la invoca acreditar su validez.

***Firma digital***

Se entiende por firma digital al resultado de aplicar a un documento digital un procedimiento matemático que requiere información de exclusivo conocimiento del firmante, encontrándose ésta bajo su absoluto control. La firma digital debe ser susceptible de verificación por terceras partes, tal que dicha verificación simultáneamente permita identificar al firmante y detectar cualquier alteración del documento digital posterior a su firma.

*Propiedades*:

* Va ligada indisolublemente al mensaje. Una firma digital válida para un documento no puede ser válida para otro distinto.
* Solo puede ser generada por su legítimo titular. Al igual que cada persona tiene una forma diferente de escribir, y que la escritura de dos personas diferentes puede ser distinguida mediante análisis caligráficos, una firma digital sólo puede ser construida por la persona o personas a quienes legalmente corresponde.
* Es públicamente verificable. Cualquiera puede comprobar su autenticidad en cualquier momento, de forma sencilla.

***PKI – Infraestructura de Clave Publica***

Es una combinación de hardware, software, políticas y procedimientos de seguridad

que define un entorno de confianza **centralizado** y provee garantías para operaciones

criptográficas como el cifrado, la firma digital o el no repudio de transacciones electrónicas.

***Certificados Digitales***

Se entiende por certificado digital al documento digital firmado digitalmente por un certificador, que vincula los datos de verificación de firma a su titular Es esencialmente una clave pública, un identificador e información accesoria; firmados digitalmente por una autoridad de certificación, y su utilidad es demostrar que una clave pública pertenece a un usuario concreto.

El estándar X.509 solo define la sintaxis de los certificados, por lo que no está atado a ningún algoritmo en particular, y contempla los siguientes campos:

- Versión.

- Número de serie.

- Identificador del algoritmo empleado para la firma digital.

- Nombre del certificador.

- Periodo de validez.

- Nombre del sujeto.

- Clave pública del sujeto.

- Identificador único del certificador.

- Identificador único del sujeto.

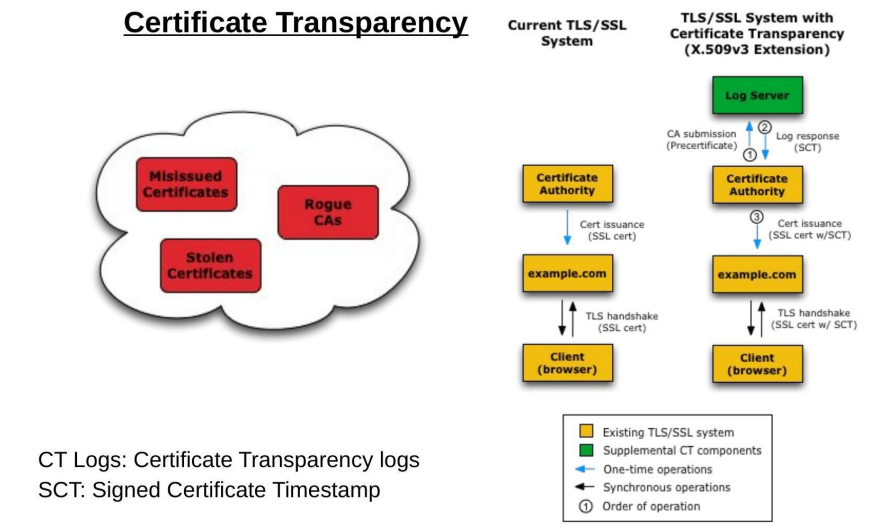
- Extensiones.

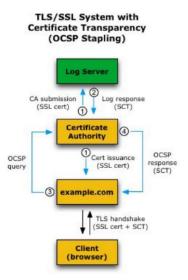
- Firma digital de todo lo anterior generada por el certificador.

*Certificados digitales de revocación*

Cuando una clave pública pierde su validez —por destrucción o robo de la clave privada correspondiente, por ejemplo—, es necesario anularla. Para ello se emplean los denominados certificados de revocación que no son más que un mensaje que identifica a la clave pública que se desea anular, firmada por la clave privada correspondiente.

*Componentes*

* **Autoridad de Certificación** (CA, Certificate Authority): Emite y revoca certificados, vinculando las claves publicas con la identidad del propietario.
* **Autoridad de Registro** (RA, Registration Authority): Verifica la relación de los certificados y la identidad de sus titulares.
* **Autoridad de Validación** (VA, Validation Authority): Es la encargada de comprobar la validez de los certificados digitales.
* **Autoridad de Sellado de Tiempo** (TSA, TimeStamp Authority): Es la encargada de firmar documentos con la finalidad de probar que existían antes de un determinado instante de tiempo.
* **Los repositorios:** Almacenan información relativa a la PKI, como certificados y listas de revocacion (CRL, Certificate Revocation List).
* **Los usuarios y entidades finales** que poseen un par de claves (pública y privada) y un certificado asociado a su clave pública.

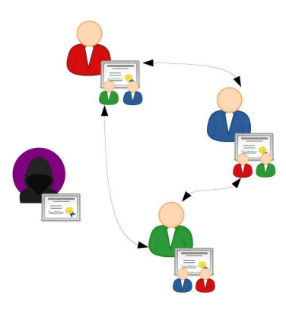


***OCSP***

OCSP: es un método para determinar el estado de

vigencia de un certificado digital X.509 usando otros

medios que no sean el uso de CRL (Listas de Revocación de Certificados).



***Anillo o Circulo de confianza***

Es un modelo de confianza distribuido que provee

garantías para operaciones criptográficas como el

cifrado, la firma o el no repudio de transacciones.

electrónicas basado en la cantidad de firmas de

actores de confianza que posea una clave pública.

Ejemplo: El concepto original es un aporte de Phil Zimmermann, documentado en el manual de PGP Version 2.0 (1992)

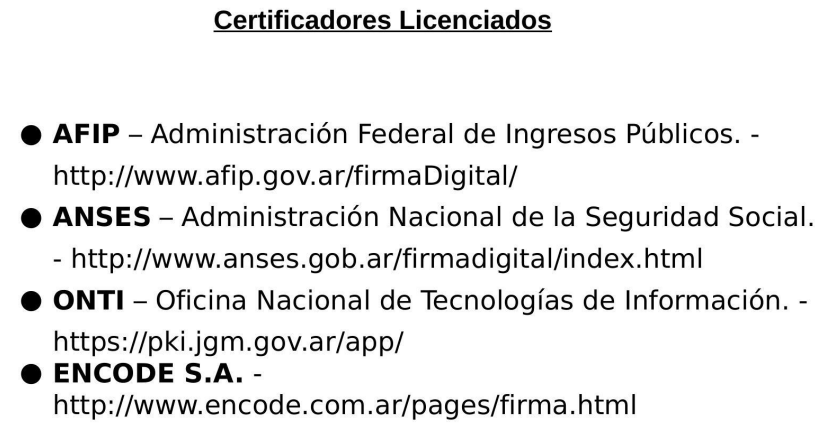
Con el paso del tiempo, usted acumulará claves de otras personas que podría querer designar como introductores de confianza. Todos los demás elegirán sus propios introductores de confianza. Y todos gradualmente acumularán y distribuirán junto con su clave una colección de firmas certificadas por otras personas, en la expectativa de que quien quiera que la reciba confiará por lo menos en una o dos de las firmas. Esto llevará a la aparición (espontánea) de un anillo de confianza descentralizado y resistente a los fallos para todas las claves públicas.

***Ley de Firma Digital – Republica Argentina***

Ley 25.506

Consideraciones generales. Certificados digitales. Certificador licenciado. Titular de un certificado digital. Organización institucional. Autoridad de aplicación. Sistema de auditoría. Comisión Asesora para la Infraestrutura de Firma Digital. Responsabilidad. Sanciones. Disposiciones Complementarias. Sancionada: Novembre 14 de 2001. Promulgada de Hecho: Diciembre 11 de 2001.





***Logging (clase 4)***

Su finalidad se vincula a los siguientes puntos:

* Identificar incidentes de seguridad
* Monitorear violaciones a las políticas
* Asistir en controles de no-repudio
* Proveer información sobre problemas o situaciones atípicas
* Contribuir con información especifica para la investigación de incidentes que no pueda obtenerse de otras fuentes
* Contribuir con la defensa ante vulnerabilidades y exploits mediante de la detección de ataques

*¿ Donde registrar ?* Sistemas de archivos, almacenamiento en la nube, bases de datos SQL y NoSQL

*¿ Que registrar ?* Fallos de validación, autenticación, autorización, anomalías, fallas de aplicación, eventos legales...

*¿ Que NO registrar ?* Código fuente, identificadores de sesión, credenciales y tokens de acceso, claves de cifrado, SPl...

***Enmascaramiento de datos***

Es el proceso mediante el cual se reemplazan los datos sensibles de un sistema con el objetivo de proteger esta información confidencial ante situaciones que escapen al control sobre la misma.

Segun GDPR, "seudonimización", es el tratamiento de datos personales de manera tal que ya no puedan atribuirse a un interesado sin utilizar información adicional, siempre que dicha información adicional figure por separado y esté sujeta a medidas técnicas y organizativas destinadas a garantizar que los datos personales no se atribuyan a una persona física identificada o identificable.

***Debilidades asociadas***

* CWE-117 Improper Output Neutralization for Logs
* CWE-223 Omission of Security-relevant Information
* CWE-532 Insertion of Sensitive Information into Log File
* CWE-778 Insufficient Logging

***Validación de entrada***

* La debilidad de seguridad más común en aplicaciones web es la falta de validación apropiada de las entradas del cliente o del entorno. Esta debilidad lleva a casi todas las principales vulnerabilidades en las aplicaciones web, tales como inyección a intérprete, ataques locale/Unicode, ataques al sistema de archivos y desbordamientos de memoria. Nunca se debe confiar en los datos introducidos por el cliente, ya que tiene todas las posibilidades de manipular los datos.
* La validación debe cubrir los aspectos semánticos como sintácticos de la información ingresada. Se sugiere implementar una metodología de validación de tipo "White List" para solo validar el ingreso de datos permitidos. Para este fin se debe describir el tipo válido de datos, se sugiere implementarlo mediante listas de valores o expresiones regulares.
* Expresiones regulares de validación de entrada por tipos:

**URL**:^(((https?|fts?|gopher|telnet|nntp)://)|(mailto:/news:))(%[0-9A-Fa-((2}I-0\_1-\*:/?:@&=+$,A-Za-z0-9])+D.l';/?: J[[:blank:]])?$

**IP**:^(25(0-5]|20-4][0-9]|[01]?(0-9][0-9]?)1(25(0-5]|2[0-4][0-9]|[01]?[0-9)[0-9|?)L(25[0-5]|2[0-4][0 91|[01)7[0-91[0-9]?)L(25[0-51|210-4][0-9]|[01]710-9][0-9]?)S

**EMAIL:**^[a-zA-ZO-9+&-]+(?:.[a-zA-ZO-9\_+&-]+)\*@(?:[a-zA-ZO-9-]+.)+[a-zA-2)(2,7)S

**NUMEROS:**^(cero|uno|dos|tres|cuatro|cinco|seis|siete/ocho|nueve)$

***Autenticación en la Web***

Su objetivo es proveer servicios de autenticación segura a las aplicaciones Web, mediante:

* Vinculando una unidad del sistema a un usuario individual mediante el uso de una credencial
* Proveyendo controles de autenticación razonables de acuerdo al riesgo de la aplicación.
* Denegando el acceso a atacantes que usan varios métodos para atacar el sistema de autenticación.

Consideraciones generales

* La autenticación es solo tan fuerte como sus procesos de administración de usuarios
* Use la forma más apropiada de autenticación adecuada para su clasificación de bienes
* Re-autenticar al usuario para transacciones de alto valor y acceso a áreas protegidas
* Autenticar la transacción, no el usuario
* Las contraseñas son trivialmente rotas y no son adecuadas para sistemas de alto valor