



UNIVERSIDAD NACIONAL DE LA MATANZA
Departamento de Ingeniería e Investigaciones Tecnológicas

Seguridad y Calidad en Aplicaciones Web



Unidad N° 3: Criptografía

Referente de Cátedra: Walter R. Ureta
Plantel Docente: Pablo Pomar, Walter R. Ureta



Historia de la criptografía



Historia de la criptografía

- 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). Éste 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.



Historia de la criptografía

- 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 las bases matemáticas de la teoría de la información y publica "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 sólo uso.



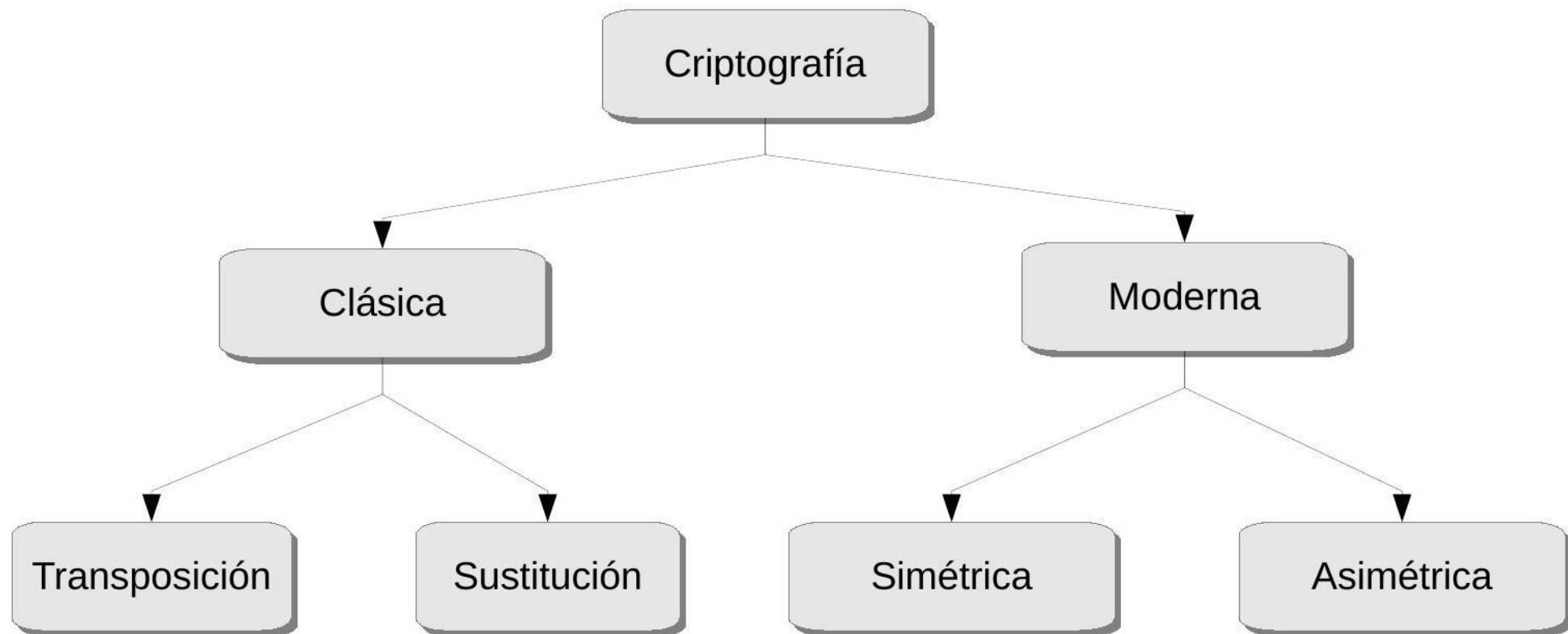
Elementos teóricos de la criptografía

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

- **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.
- **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**.

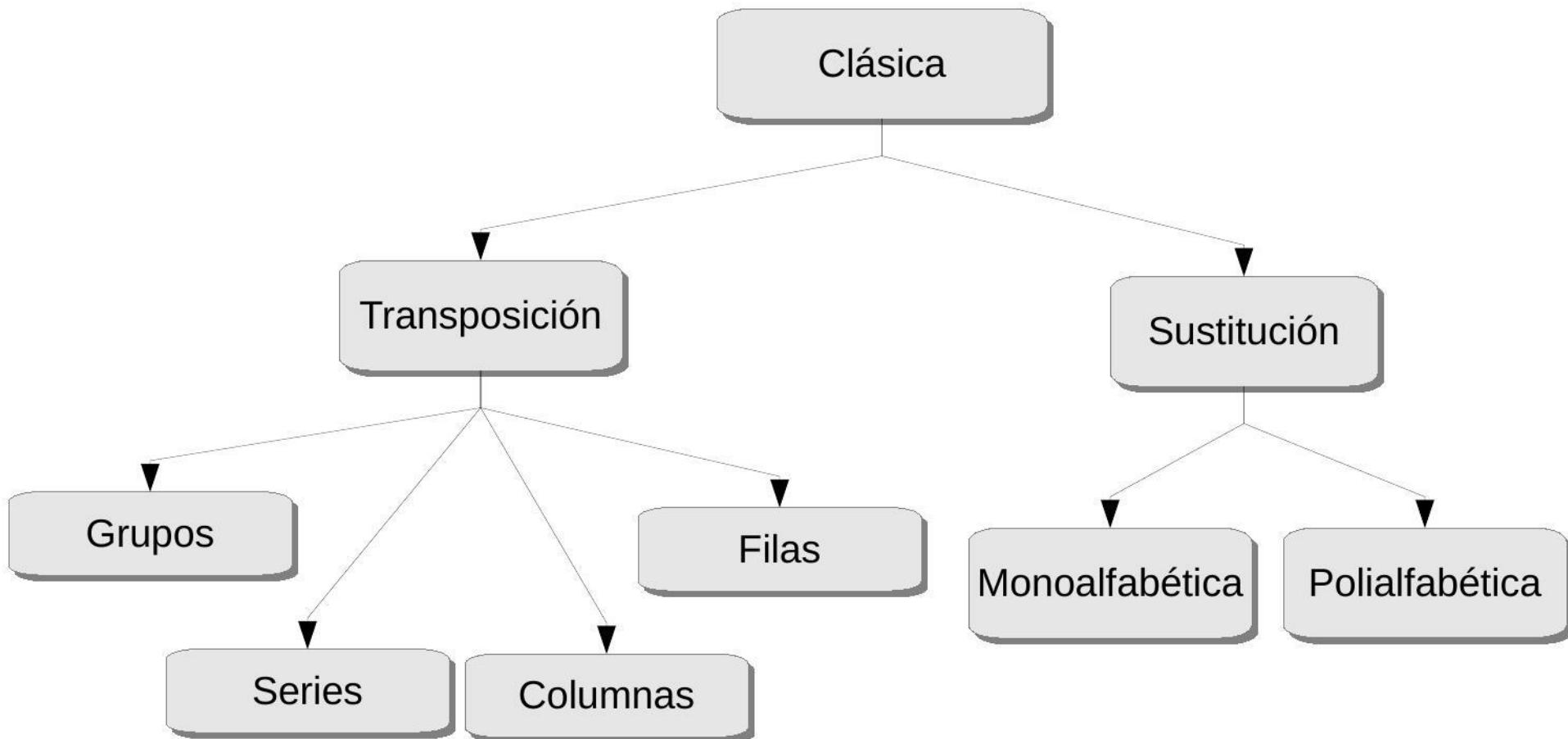


Clasificación





Criptografía clásica Sub-Clasificación

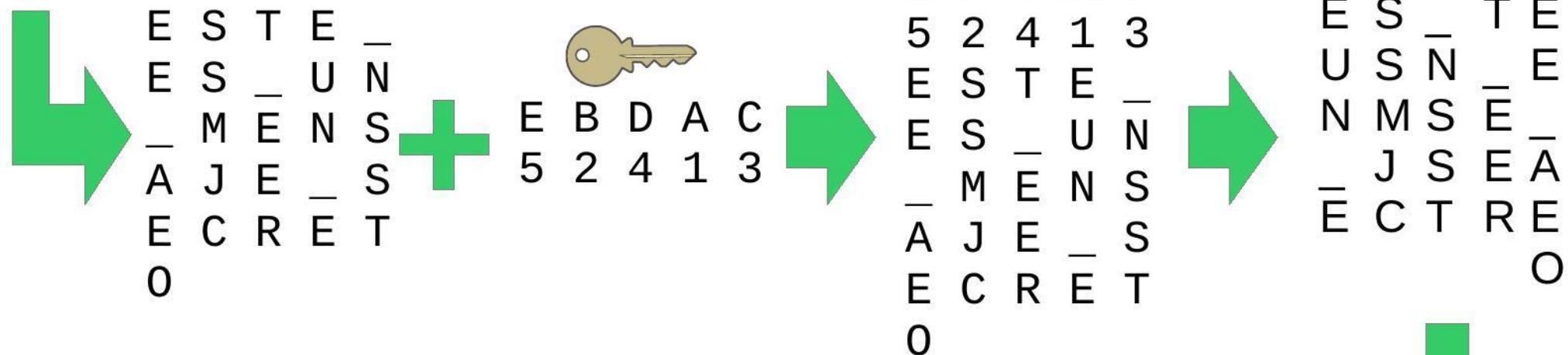




Criptografía clásica

Los **cifradores por transposición** utilizan la técnica de permutación de forma que los caracteres del texto se reordenan mediante un algoritmo específico.

ESTE ES UN MENSAJE SECRETO





Criptografía clásica

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 cifrador es polialfabético. Por ejemplo, el cifrado del Cesar (**monoalfabético**)

ABCDEFGHIJKLMNPQRSTUVWXYZ

MENSAJE



$f(M, 3)$



PHQVDMH



Criptografía clásica

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:

ABCDABCDABCDABC

El texto cifrado es:

TFZWO EGSRV GEA



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z				
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z					
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z						
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z							
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z								
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z									
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z										
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z											
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z												
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z													
O	O	P	Q	R	S	T	U	V	W	X	Y	Z														
P	P	Q	R	S	T	U	V	W	X	Y	Z															
Q	Q	R	S	T	U	V	W	X	Y	Z																
R	R	S	T	U	V	W	X	Y	Z																	
S	S	T	U	V	W	X	Y	Z																		
T	T	U	V	W	X	Y	Z																			
U	U	W	X	Y	Z																					
V	V	W	X	Y	Z																					
W	W	X	Y	Z																						
X	X	Y	Z																							
Y	Y	Z																								
Z	Z																									



Referencia – Esquemas de codificación

Binario

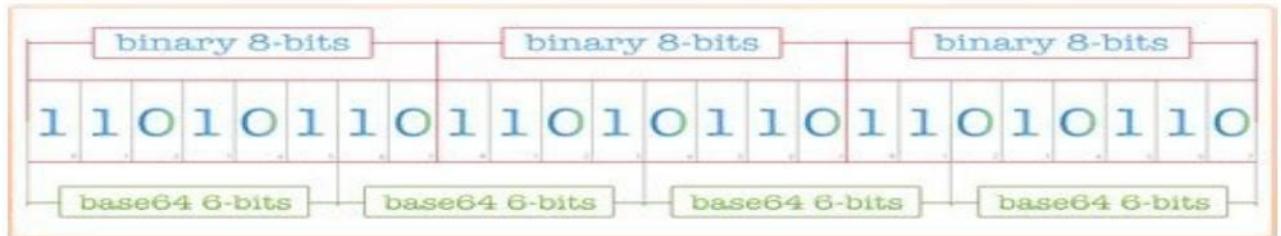
Representa la información a nivel de bits con 0 y 1

Hexadecimal

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

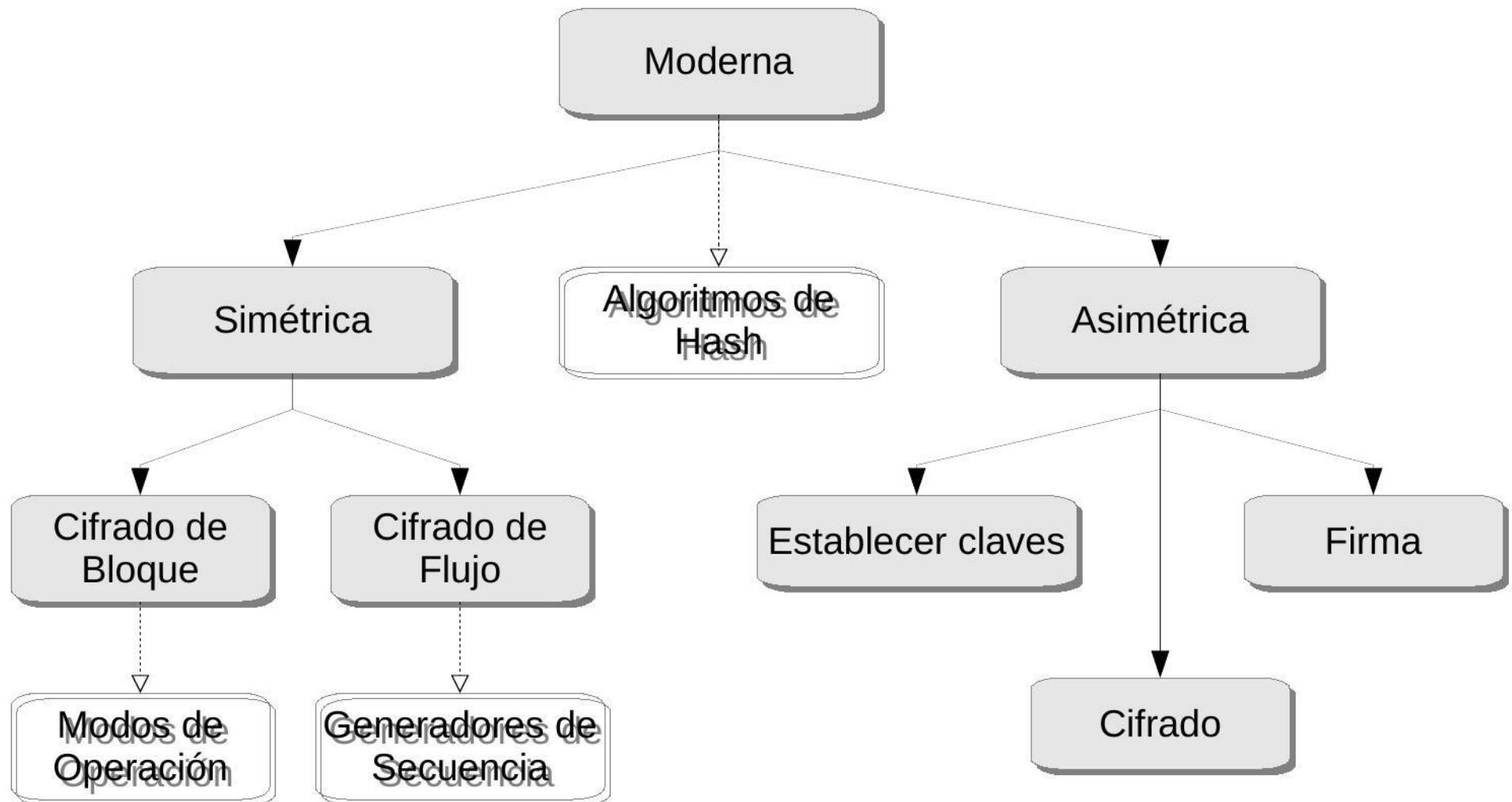
Base64

Representa la información con 64 caracteres representables (ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/-), indicando con un carácter especial (=) los octetos faltantes del ultimo bloque.





Criptografía moderna Sub-Clasificación





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.



Algoritmos simétricos

Ventajas

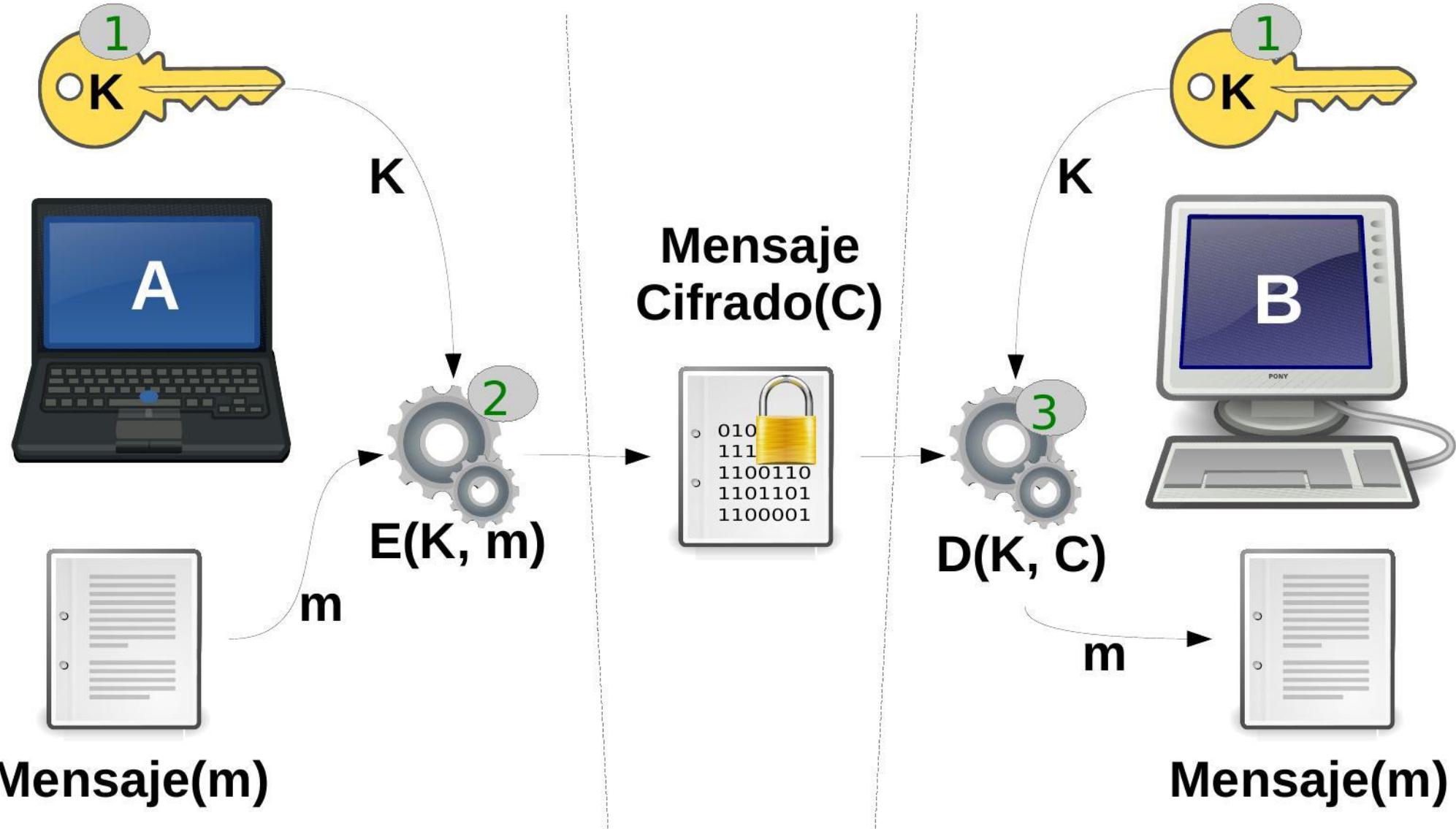
- Sencillez de implementación
- Robustez
- Velocidad de cifrado
- Longitud del mensaje

Desventajas

- La clave debe ser compartida previamente con seguridad
- La comunicación entre múltiples actores requiere numerosas claves



Algoritmos simétricos - Cifrado





Algoritmos simétricos de Bloque

Ejemplos

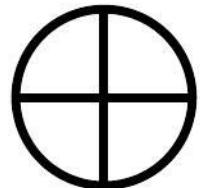
- DES-LUCIFER (1976, Data Encryption Standard)
- 3DES (1998, Triple Data Encryption Standard, NIST)
- **AES-Rijndael (2001, Advanced Encryption Standard, NIST)**
- **Serpent (1998)**
- **Twofish**
- **RC6 (1998, Rivest Cipher 6)**
- **MARS (1998, IBM)**
- GOST (1994, Magma URSS)
- CAMELLIA (2000, NTT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



Referencia matemática - XOR

Propiedades

- Es commutativa: Es decir que $A \text{ xor } B = B \text{ xor } A$
- Asociativa: $(A \text{ xor } B) \text{ xor } C = A \text{ xor } (B \text{ xor } C)$
- Autoinversa: $(A \text{ xor } B) \text{ xor } B = A$



<i>A</i>	<i>B</i>	<i>XOR</i>
0	0	0
0	1	1
1	0	1
1	1	0

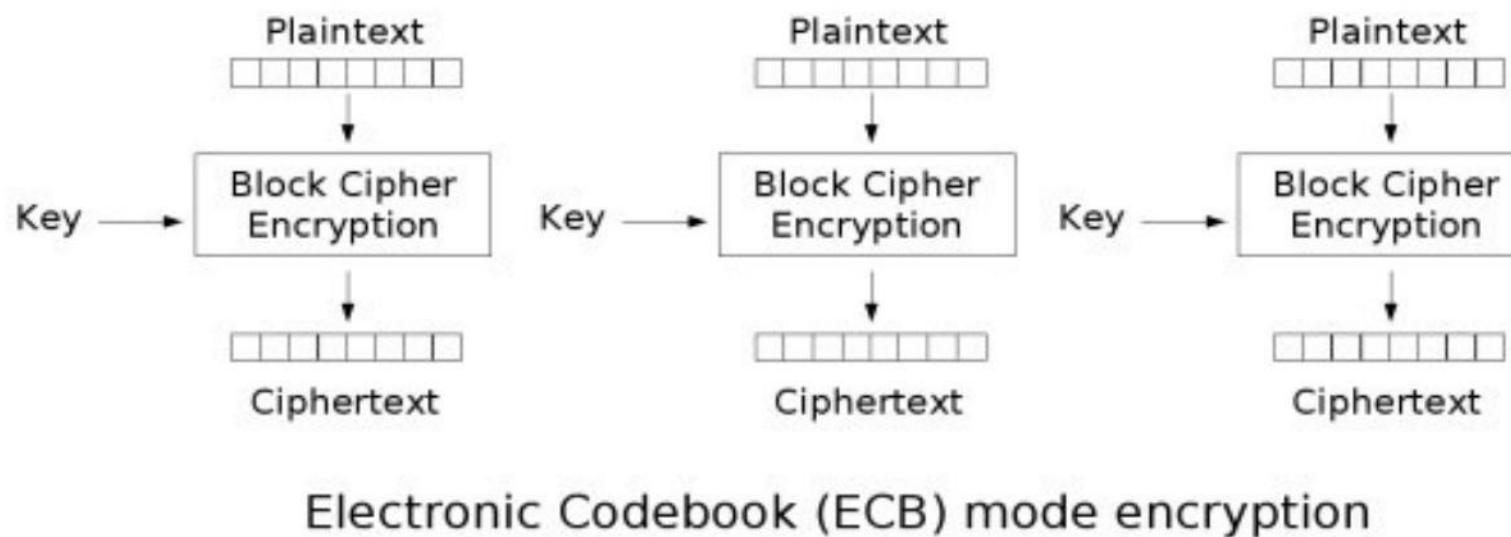


An XOR might keep your kid sister from reading your files, but it won't stop a cryptanalyst for more than a few minutes. -Bruce Schneier



Modos de cifrado de bloques

ECB: Electronic codebook, en este método el mensaje se fracciona en partes y cada una es cifrada de manera independiente.





Padding o Esquema de Relleno

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 último 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/IEC 9797-1)
- ISO/IEC 7816-4
- **PKCS#7 (RFC2315, Sec 10.3)**
- ISO 10126
- ANSI X.923



Padding o Esquema de Relleno

Bit padding: operando a nivel de bits adiciona **1** y posteriormente **N** cantidad de **0** hasta completar el tamaño requerido.

01011100 10110011 01101000 00000000

4 bytes

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

FF FF FF 80 00 00 00 00

8 bytes

PKCS#7: operando a nivel de bytes **N** cantidad de bytes idénticos cuyo valor es la cantidad de bytes agregados.

FF FF 06 06 06 06 06 06

8 bytes



Padding o Esquema de Relleno

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

FF FF FF C2 A7 2E 14 05

8 bytes

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

FF FF FF FF FF 00 00 03

8 bytes

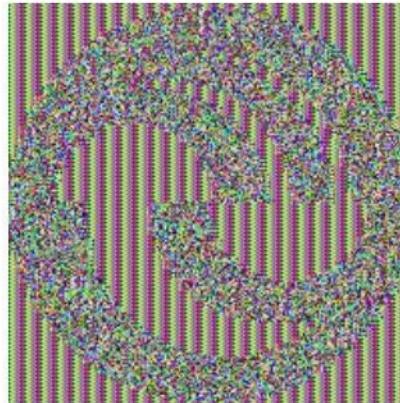


Ataques por marca de agua

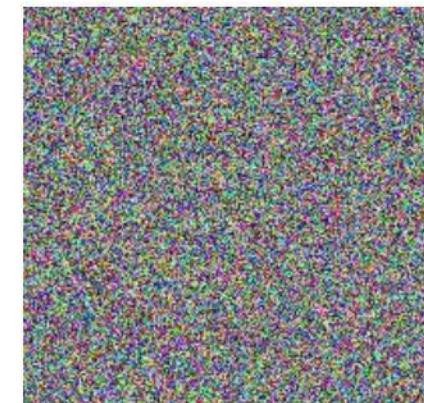
A continuación se observa el resultado de cifrar con AES128 una imagen utilizando diferentes modos de cifrado de bloque.



Original



ECB



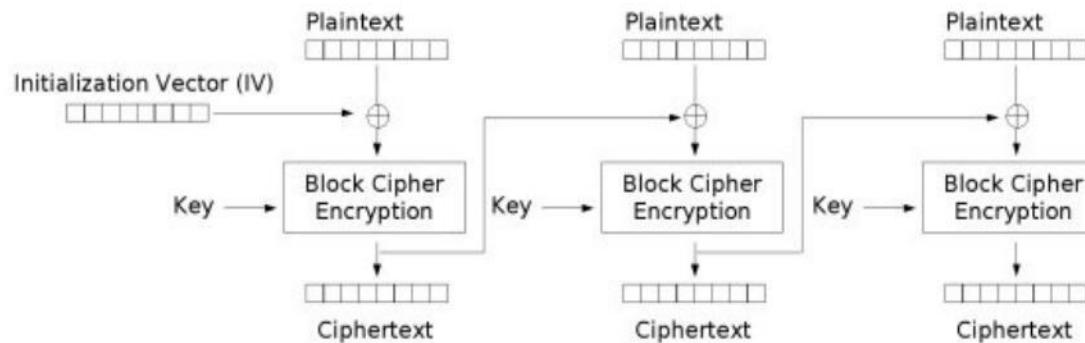
CBC

```
head -n 4 unlam.ppm > header.txt  
tail -n +5 unlam.ppm > body.bin  
openssl enc -aes-128-ecb -nosalt -pass pass:"scaw" -in body.bin -out body.ecb.bin  
cat header.txt body.ecb.bin > unlam.ecb.ppm  
head -n 4 unlam.ppm > header.txt  
tail -n +5 unlam.ppm > body.bin  
openssl enc -aes-128-cbc -nosalt -pass pass:"scaw" -in body.bin -out body.ecb.bin  
cat header.txt body.ecb.bin > unlam.cbc.ppm
```

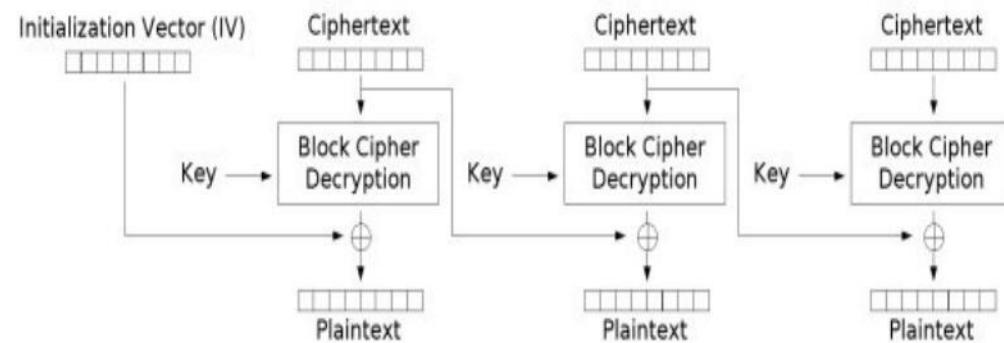


Modos de cifrado de bloques

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.



Cipher Block Chaining (CBC) mode encryption

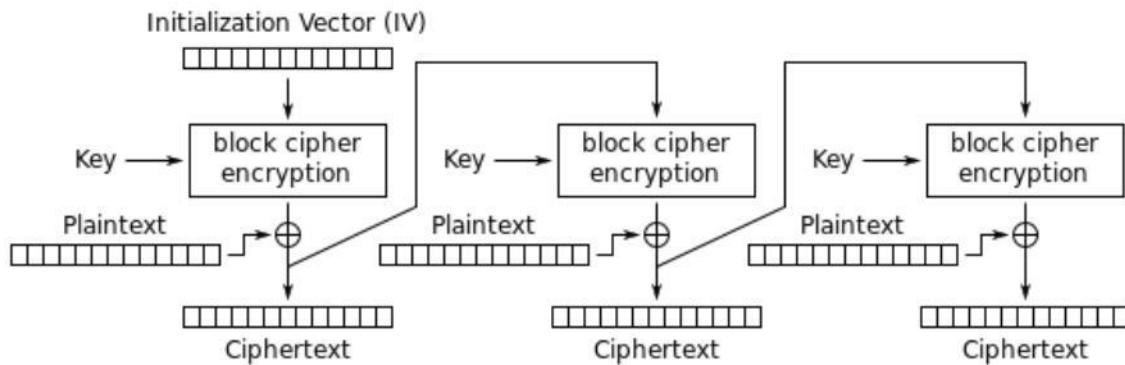


Cipher Block Chaining (CBC) mode decryption

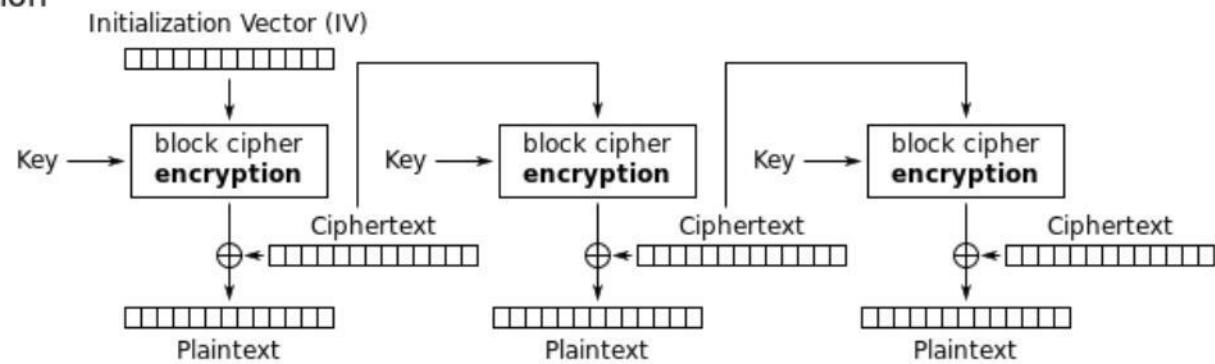


Modos de cifrado de bloques

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.



Cipher Feedback (CFB) mode encryption

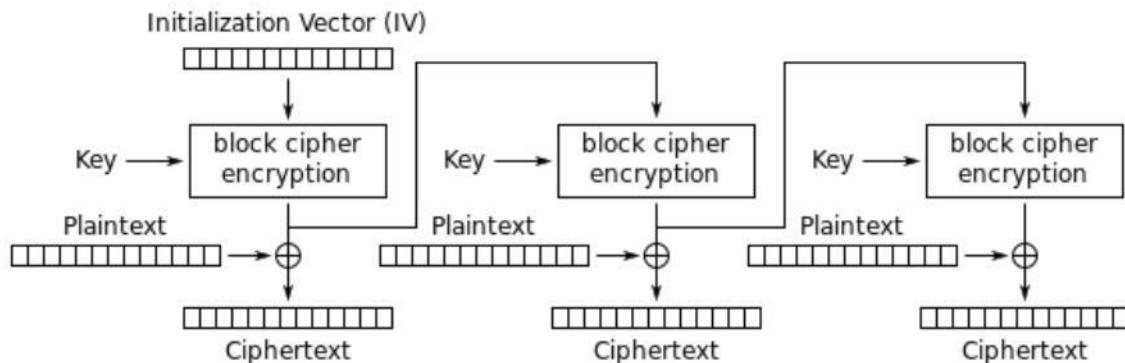


Cipher Feedback (CFB) mode decryption

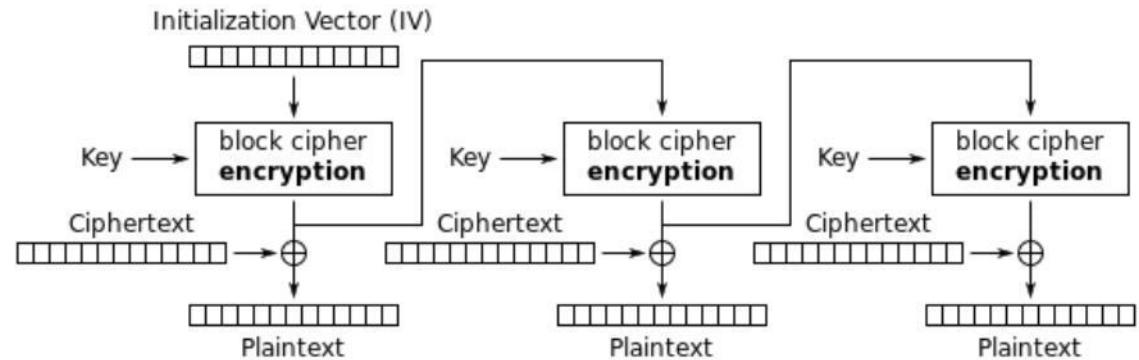


Modos de cifrado de bloques

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.



Output Feedback (OFB) mode encryption

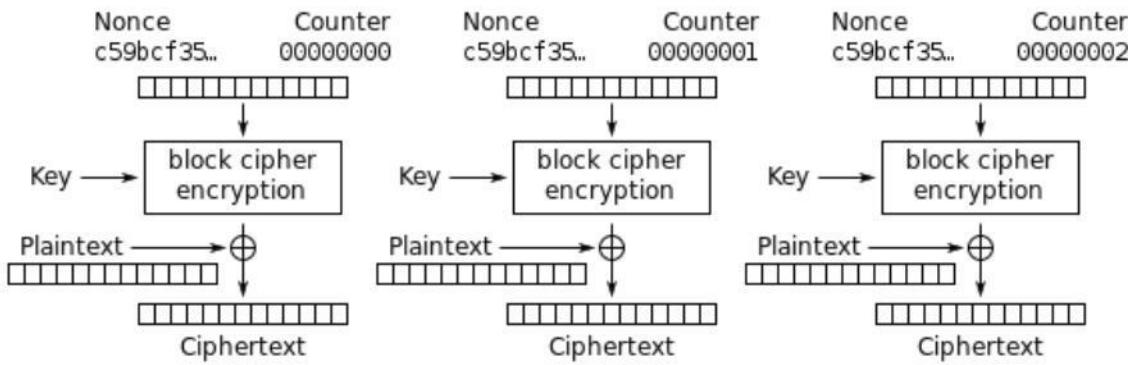


Output Feedback (OFB) mode decryption

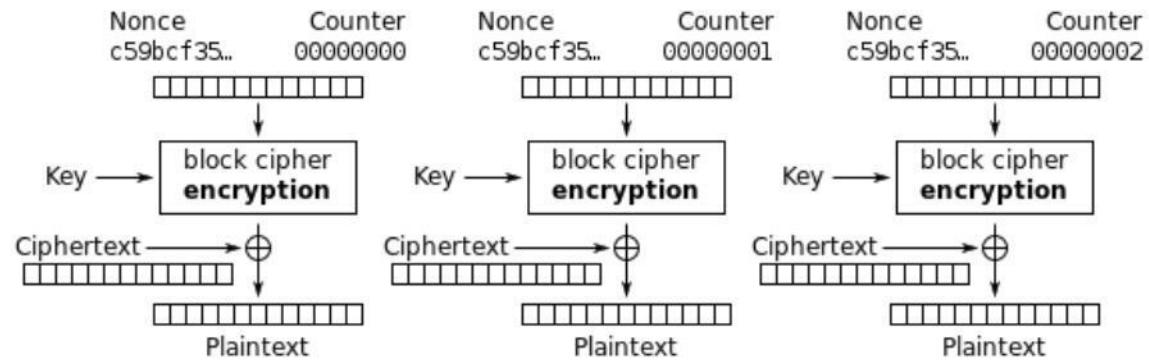


Modos de cifrado de bloques

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 .



Counter (CTR) mode encryption



Counter (CTR) mode decryption



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



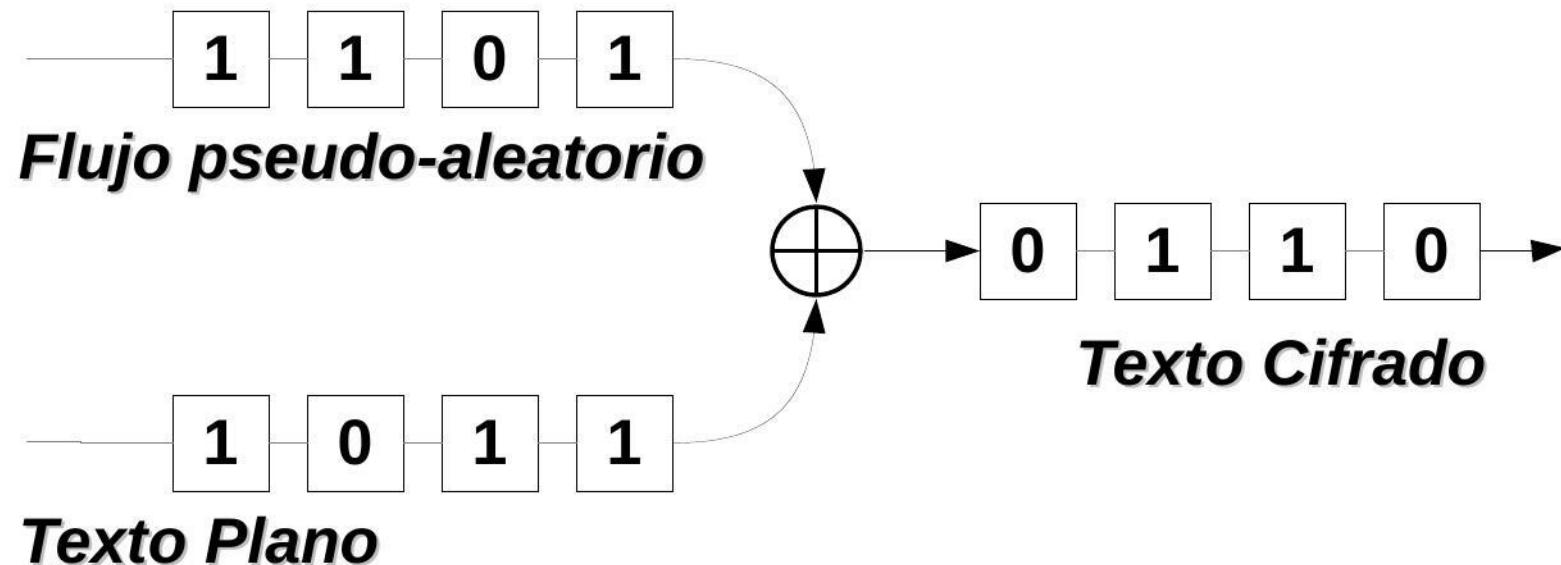
Cifrado de flujo

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 de 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?



Cifrado de flujo

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).





Cifrado de flujo

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 numero 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.



Cifrado de flujo

Estos son algunos de los actuales algoritmos de cifrado de flujo

- RC4
- Salsa20*
- ChaCha20
- Trivium*
- A5/1, A5/2
- Chameleon
- FISH
- Helix
- Grain*
- ISAAC
- MUGI
- Panama
- Phelix
- Pike
- SEAL
- SOBER/SOBER-128
- WAKE
- Rabbit*

*(eSTREAM portfolio, de EU eCRYPT)



Funciones de HASH

Se define como una función o método no reversible para generar un valor que represente de manera casi unívoca a un dato.

Principales usos

- Soporte para criptografía asimétrica
- Tablas de Hash
- Verificación de integridad
- Soporte para procesos de autenticación



Funciones de HASH

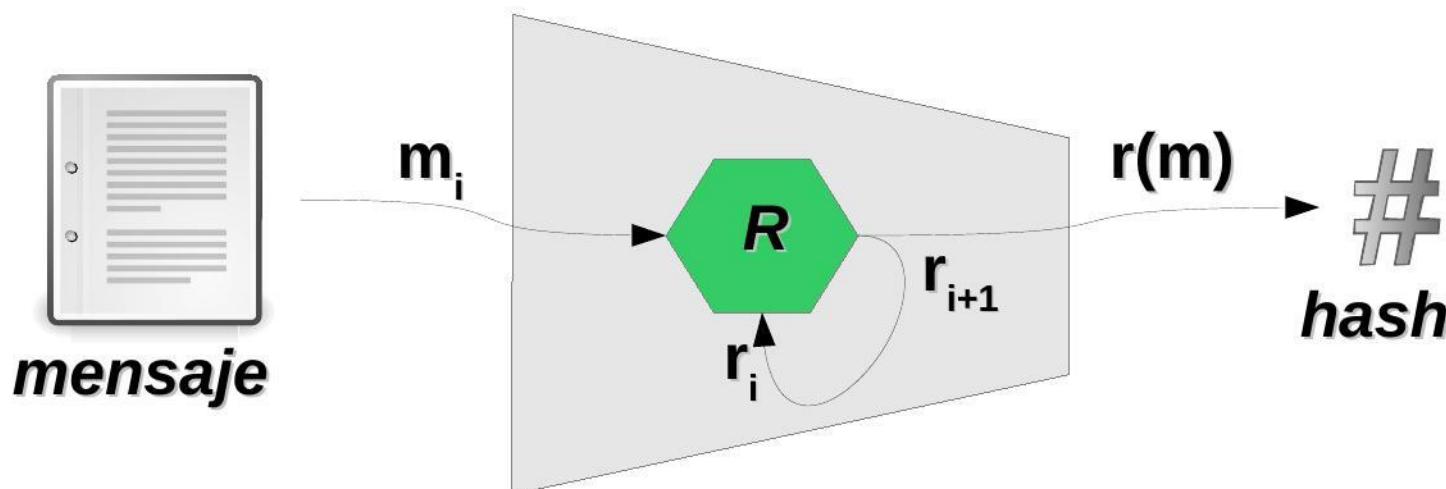
Propiedades

- $r(m)$ es de longitud fija, independientemente de la longitud de m .
- Dado m , es fácil calcular $r(m)$.
- Dado $r(m)$, es computacionalmente intratable recuperar m .
- Dado m , es computacionalmente intratable obtener un m' tal que $r(m) = r(m')$.



Funciones de HASH - MDC

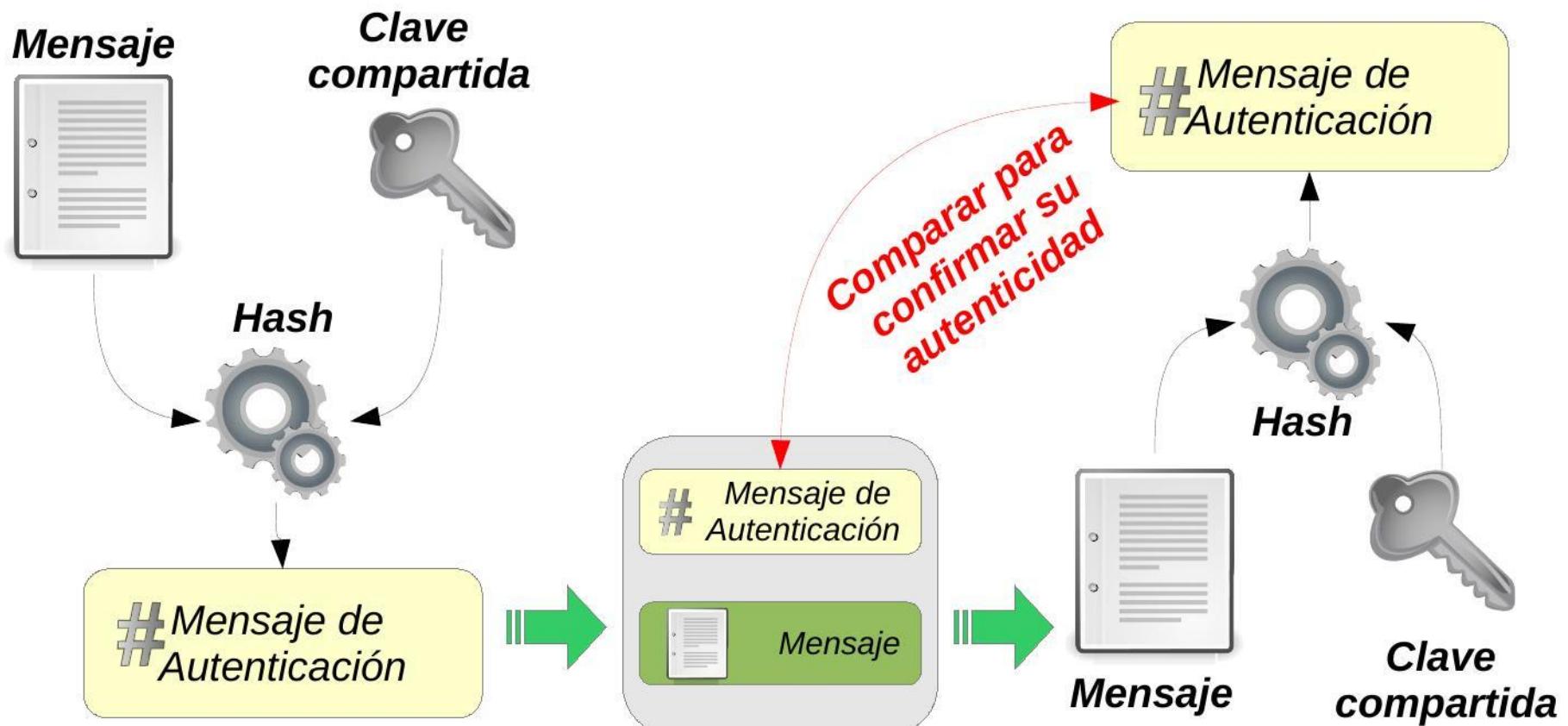
Estas funciones dan como resultado bloques de longitud fija **a** a partir de bloques de longitud fija **b**, con $a < b$. Estas funciones se encadenan de forma iterativa, haciendo que la entrada en el paso **i** sea la función del i-simo bloque del mensaje (m_i) y de la salida del paso previo, **i-1**. Se considera una buena practica incluir en el mensaje **m** la longitud y características del mensaje.





Funciones de HASH - MAC

Message Authentication Code: Adiciona criptografía al proceso de hash para aumentar la seguridad del mismo.





Funciones de HASH - MAC

Tipos de implementaciones

- **Basados en cifrados por bloques:** Consisten en cifrar el mensaje empleando un algoritmo por bloques en modo de operación CBC. El valor del MAC será entonces el resultado de cifrar el último bloque del mensaje.
- **HMAC:** Se basan en el uso de cualquier función MDC existente, aplicada sobre una versión del mensaje a la que se ha añadido un conjunto de bits, calculados a partir de la clave que se quiere emplear.
- **Basados en generadores de secuencia:** Empleando un generador de secuencia pseudoaleatorio el mensaje se parte en dos subcadenas —correspondientes al mensaje combinado con la secuencia y a la propia secuencia—, cada una de las cuales alimenta un Registro de Desplazamiento Retroalimentado. El valor del MAC se obtiene a partir de los estados finales de ambos registros.



Funciones HASH

MD4 (*Message Digest, Mensajes Digitales*). Fue Inventado por Ron Rivest de la Incorporación de Seguridad RSA (RSA Security, Inc.). Produce un valor hash de 128-bits. Se realiza una manipulación de bits para obtener el valor hash, obteniéndolo de forma rápida, provocando que sea más riesgoso en un ataque. Se considera un estándar de Internet(RFC-1320) [[STA98](#)].

MD5 Extensión a MD4. Produce como salida de 128-bits. La obtención del valor hash es lento pero considerado más seguro. Está especificado como un estándar de Internet(RFC-1321).

SHA-1 (*Secure Hash Algorithm, Algoritmo Hash Seguro*). Diseñado por NIST (National Institute of Standards and Technology), produce un valor hash de 160-bits. También está considerado como un estándar (FIPS PUB 180-1).

SHA-2 (*Secure Hash Algorithm, Algoritmo Hash Seguro*). Diseñado por la NSA (National Security Agency) y publicados por el NIST en el 2001 (FIPS PUB 180-2), es un conjunto de algoritmos comprendidos por **SHA-224, SHA-256, SHA-384** y **SHA-512**.



Funciones de HASH

SHA-3 (*Secure Hash Algorithm, Algoritmo Hash Seguro*). Llamado a concurso abierto organizado por el NIST (National Institute of Standards and Technology), adjudicado a **Keccak** durante el 2012. Sus finalistas fueron:

Keccak: Por Guido Bertoni, Joan Daemen, Michaël Peeters and Gilles Van Assche. operacion en 224,256,384 o 512 bits.

Blake: Por Jean-Philippe Aumasson, Luca Henzen, Willi Meier, Raphael C.-W. Phan, operación en 224,256,384 o 512 bits.

Grøstl: Por Praveen Gauravaram, Lars Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. Opera en 256 y 512 bits, utiliza las S-Box de AES.

JH: Por Hongjun Wu, operacion en 224,256,384 o 512 bits.

Skein: Por Bruce Schneier, Niels Ferguson, operación en 256 o 512 bits.

RIPEMD-160. Diseñada por Hans Dobbertin, Antoon Bosselaers y Bart Preneel para el proyecto RIPE (Race Integrity Primitives Evaluation, Carrera de Evaluación de Primitivas de Integridad 1988-1992). Genera una salida de 160 bits



Funciones de Derivación de Claves

Conocidas como **KDF (Key Derivation Function)** 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 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 Function - RFC5869
- **Argon2** – (2015) Por la Universidad de Luxemburgo - RFC9106



Referencia Matemática - Modular



En matemática, la aritmética modular es un sistema aritmético para clases de equivalencia de números enteros llamadas clases de congruencia.

Relación de congruencia

La aritmética modular puede ser construida matemáticamente mediante la relación de congruencia entre enteros, que es compatible con las operaciones en el anillo de enteros: **suma, resta, y multiplicación**. **a** y **b** se encuentran en la misma "clase de congruencia" módulo **n**, si ambos dejan el mismo resto si los dividimos por **n**, o, equivalentemente, si **a – b** es un múltiplo de **n**.

Esta relación se puede expresar cómodamente utilizando la notación de Gauss:

$$a \equiv b \pmod{n}$$

Así se tiene por ejemplo

$$63 \equiv 83 \pmod{10}$$

ya que ambos, **63** y **83** dejan el mismo **resto (3)** al dividir por **10**, o, equivalentemente, **63 – 83** es un múltiplo de **10**.



Algoritmos asimétricos – Primos Relativos



Sean $a, b \in \mathbb{Z}$, se dice que **son primos relativos (o coprimos)** “ a ” y “ b ” si no tienen ningún factor primo en común, es decir, si no tienen otro divisor común más que 1 ó -1, o cumplen que **el mcd (a, b) = 1**.

El algoritmo de Euclides extendido permite, además de encontrar un máximo común divisor de dos números enteros a y b , expresarlo como la mínima combinación lineal de esos números, es decir, encontrar números enteros s y t tales que **$mcd(a,b) = as + bt$** .