Ciberseguridad

Grado en Ingeniería Informática

M2 - Criptografía, Identificación y Control de Accesos

Oscar Delgado
oscar.delgado@uam.es

Álvaro Ortigosa
alvaro.ortigosa@uam.es

UAM Universidad Autónoma de Madrid

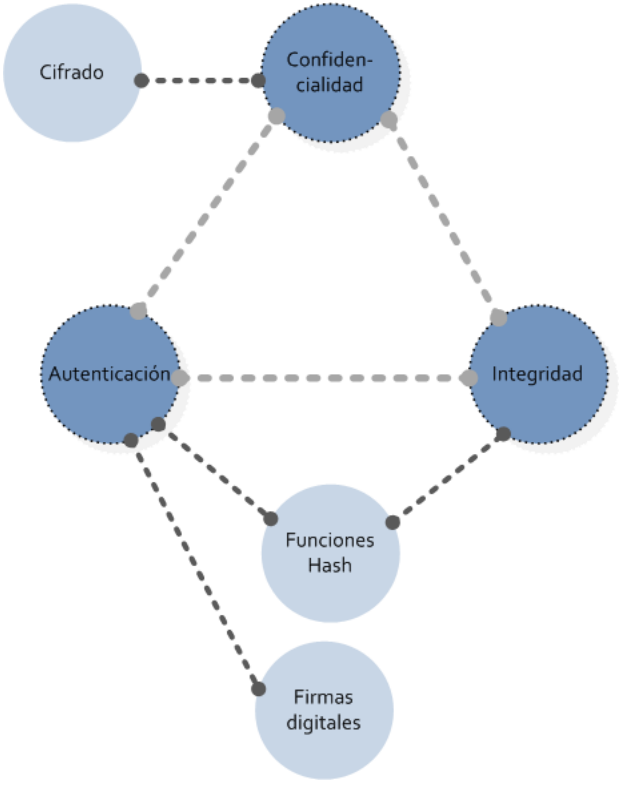
Seguridad de la Información

C.I.A. = **C**onfidencialidad, **I**ntegridad, **A**utenticación

Confidencialidad □ Secreto
Integridad □ Modificación
Autenticación □ Identidad

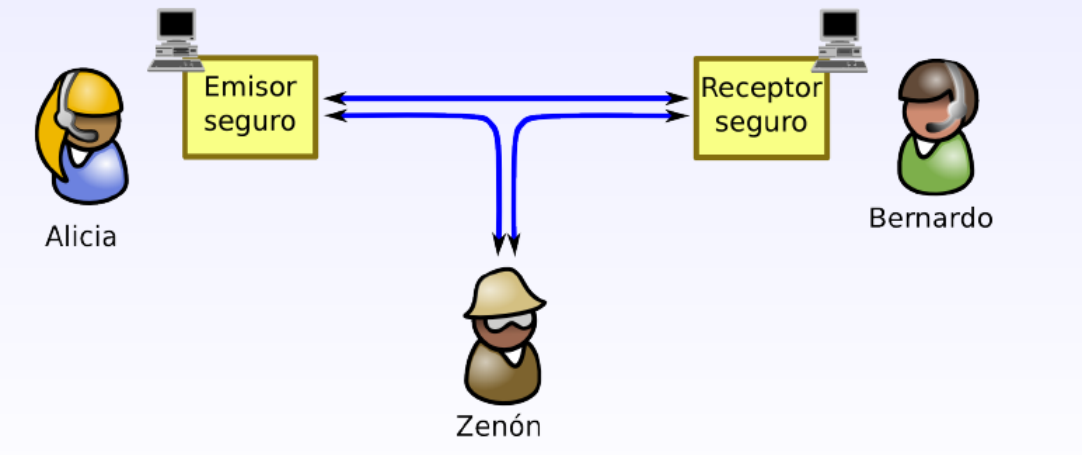
Seguridad de la información

C.I.A.



Amigos y enemigos

- Alicia y Bernardo son “amigos” y quieren comunicarse de forma “segura”.
- Zenón puede intentar interceptar, borrar o añadir mensajes.



Criptografía clásica

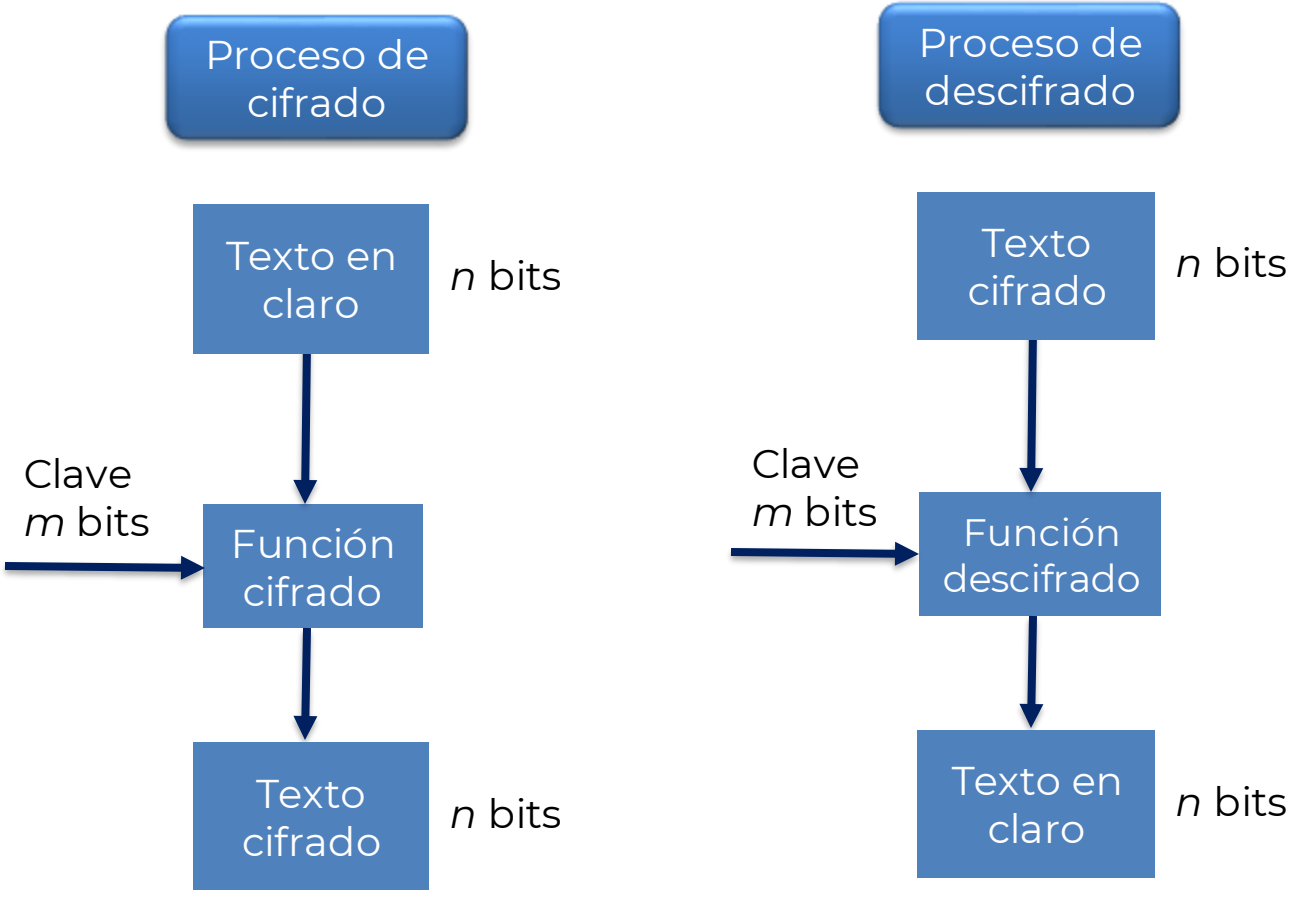
Terminología

criptología = criptografía + criptoanálisis

Cifrar, claves
~~Encriptar, llaves~~

Texto en claro
Texto cifrado, criptograma

Elementos básicos



Historia de la criptografía

Escítala espartana

Cifrado de César

$$C = (M + 3) \text{ (mód } 26)$$

Original	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
Posición	00	01	02	03	04	05	06	07	0 8	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Cifrado	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

M = FIRMA LA PAZ

C = CFOJXIXMXW

Cifrado de Vernam

El cifrado “perfecto”

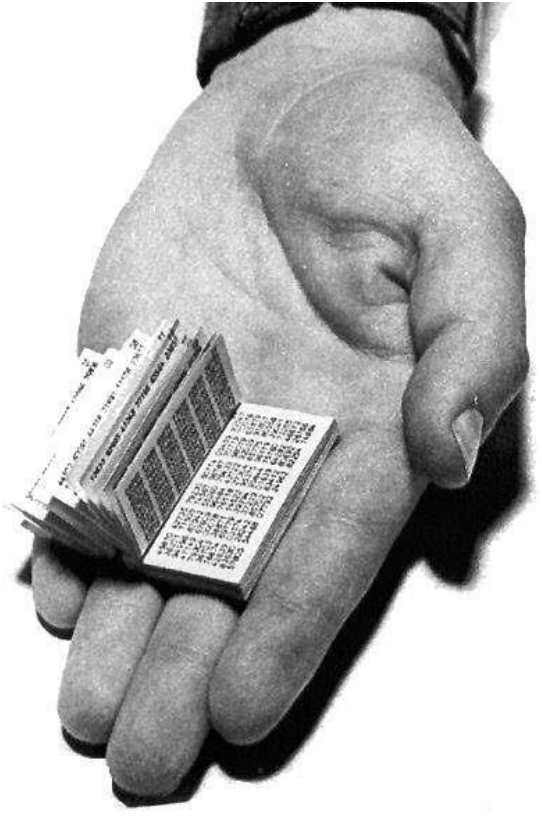
	V	E	R	N	A	M	C	I	P	H	E	R
Texto en claro	21	4	17	13	0	12	2	8	15	7	4	17
Clave aleatoria	76	48	16	82	44	3	58	11	60	5	48	88
Suma	97	52	33	95	44	15	60	19	75	12	52	105
Módulo 26	19	0	7	17	18	15	8	19	23	12	0	1
Texto cifrado	t	a	h	r	s	p	i	t	x	m	a	b

Condiciones:

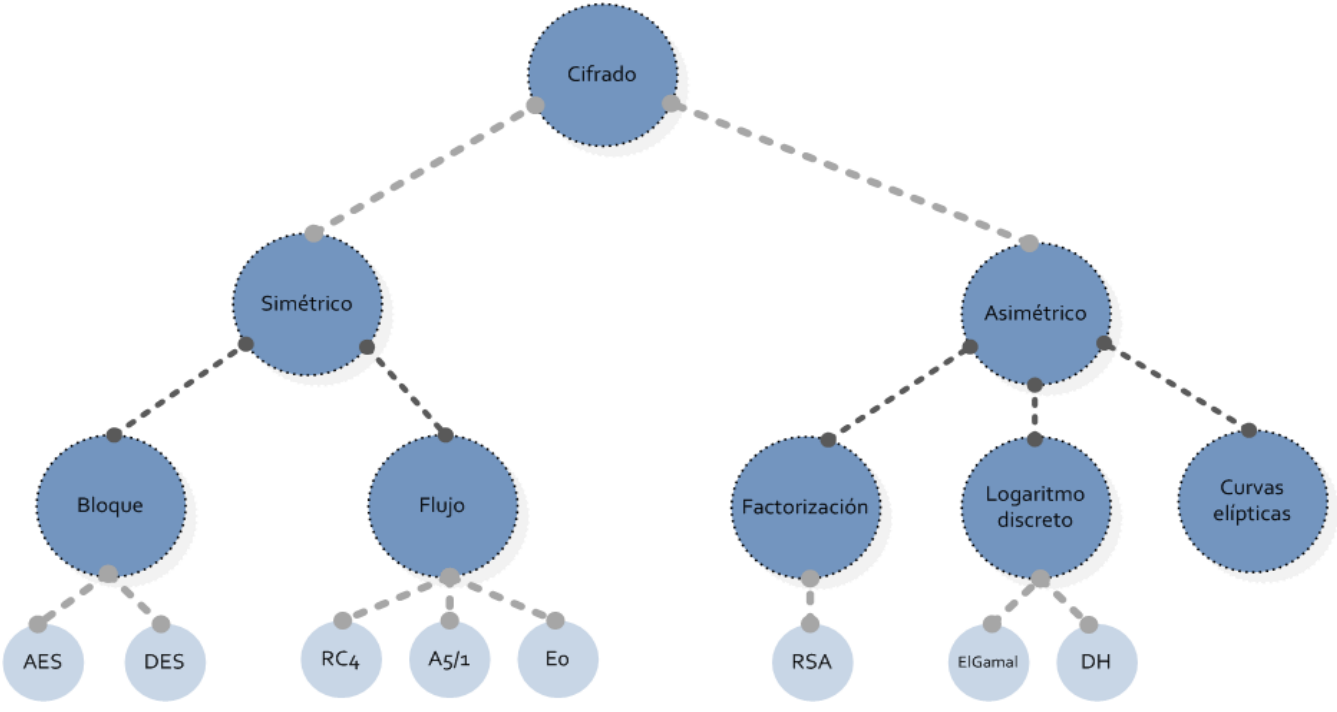
- Clave realmente aleatoria
- ¡No reutilizar nunca!
- XOR + clave aleatoria + no reutilizar = One Time Pad (OTP)

Cifrado de Vernam

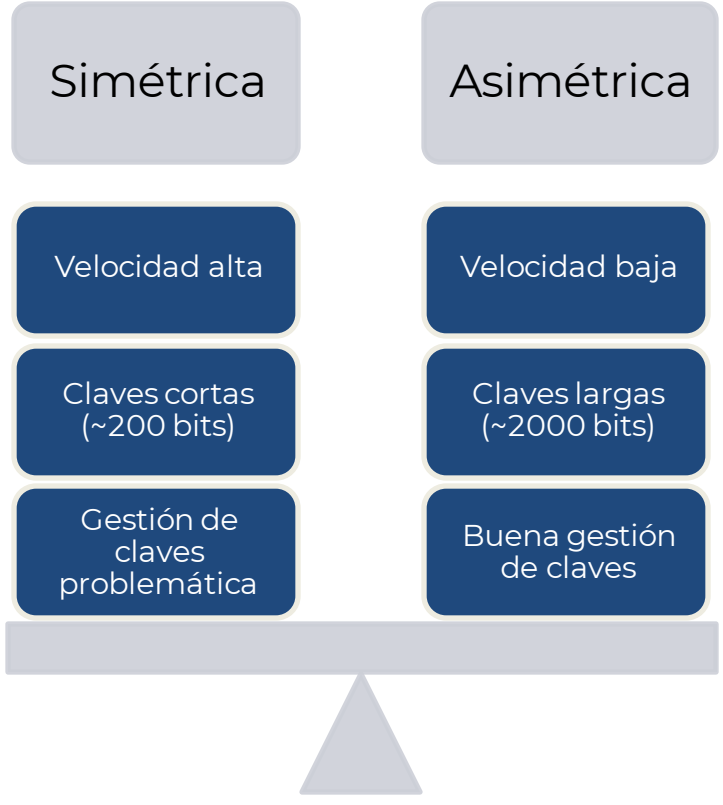
Libro de códigos ruso, capturado por el MI5



Taxonomía del cifrado



Simétrica vs Asimétrica



Principios básicos de diseño

Mecanismos básicos

- **Confusión** – ocultar relaciones entre texto en claro, texto cifrado y clave
- **Difusión** – hacer depender la salida el máximo de la entrada

Confusión

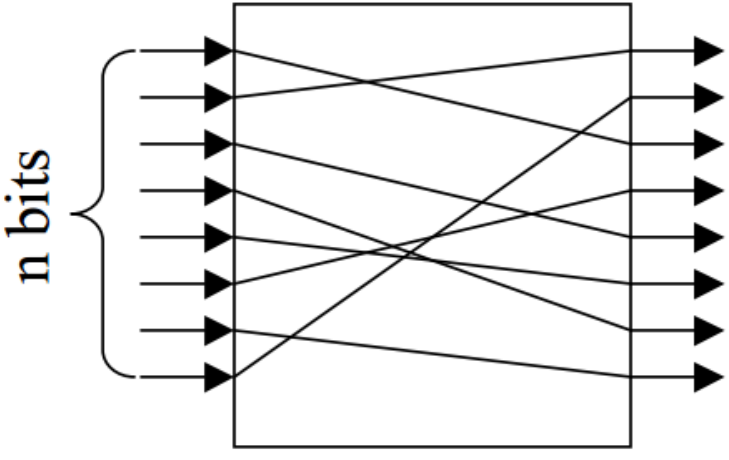
- Una palabra de entrada es sustituida por otra
- El espacio de claves posibles es $2^n!$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7

Caja S de 4x4

Difusión

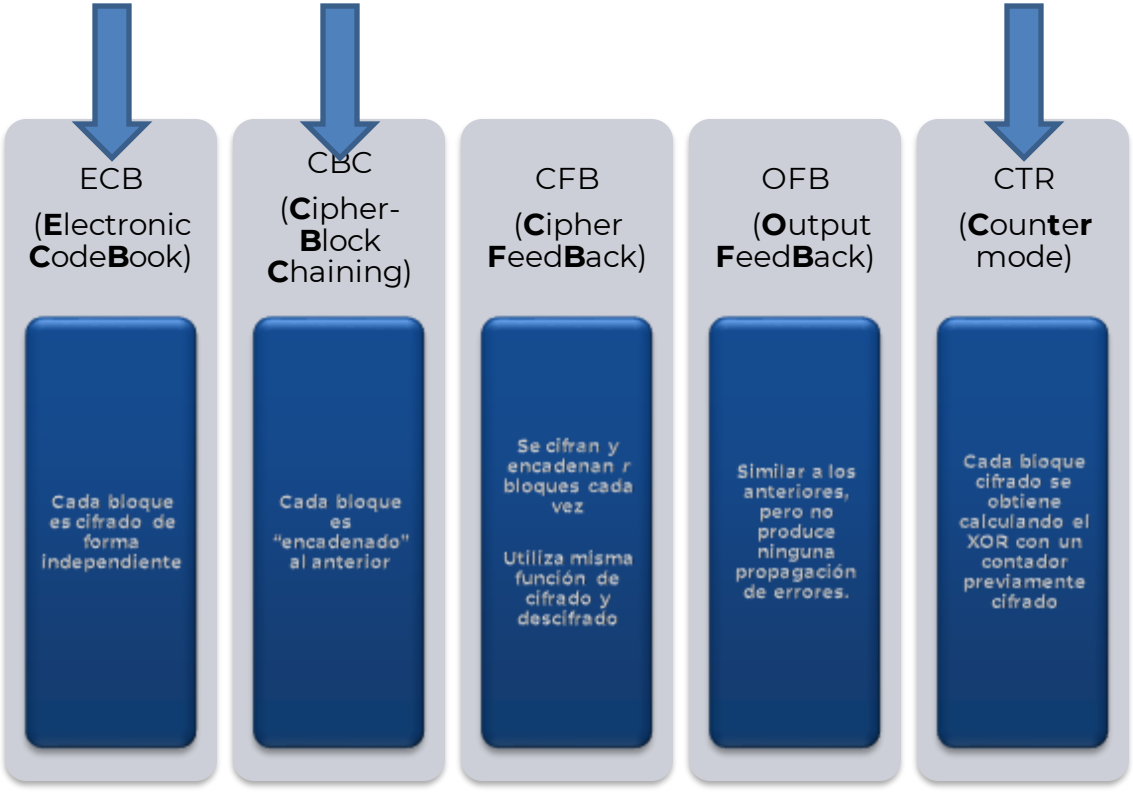
- Se cambia el orden de los bits de una palabra de entrada
- El espacio de claves posibles es n!



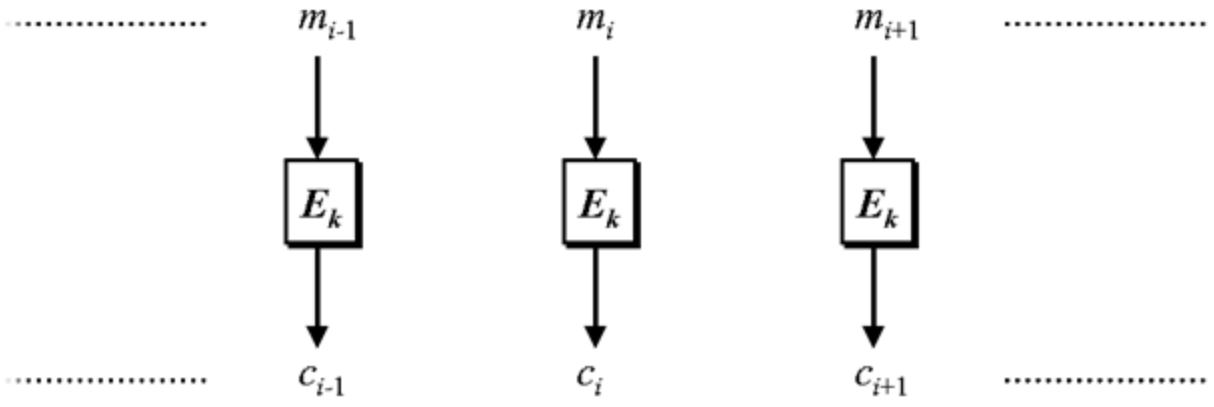
Difusión – Efecto avalancha

- El cambio de un único bit de entrada debe producir, en término medio, el cambio de la mitad de los bits de la salida.

Modos de operación



Modo ECB



- No oculta los patrones en el texto de entrada
- Un atacante puede manipular bloques: cambiar el orden, insertar o eliminar

Modo ECB

Propagación limitada de errores

Pueden reordenarse los bloques cifrados

Igual texto en claro produce idéntico texto cifrado

Vulnerable a *ataque semántico*.

Modo ECB

Imagen en claro

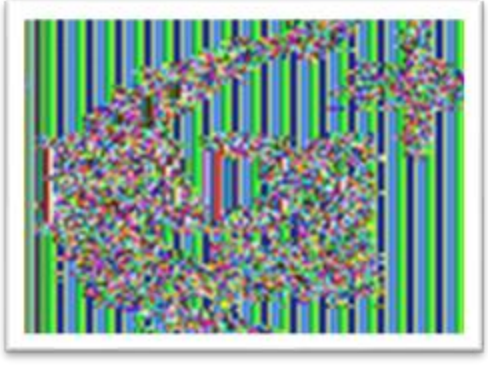


Imagen cifrada con el modo ECB

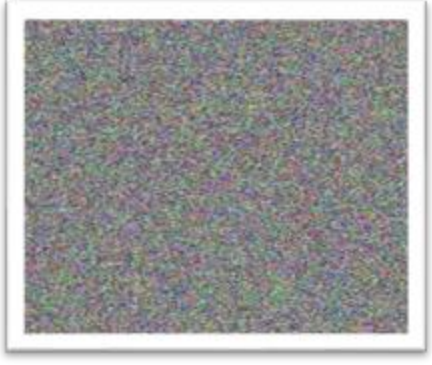
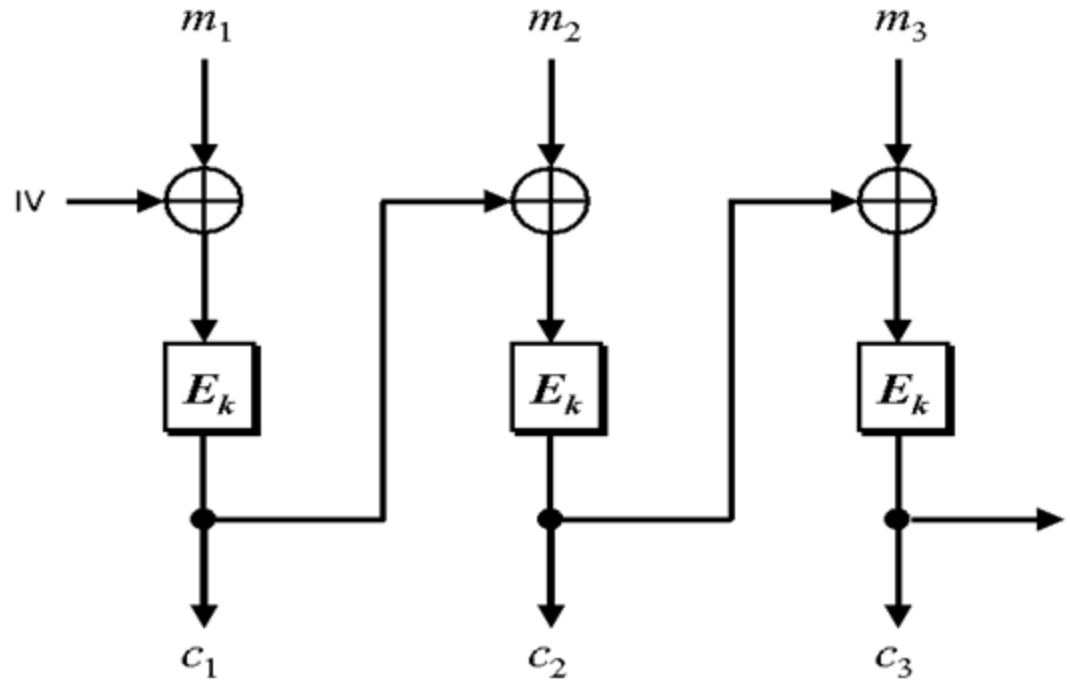


Imagen cifrada con cualquier otro modo

Modo CBC

El vector de inicialización (IV) provee la "semilla" inicial



Cada bloque cifrado se convierte en entrada al siguiente

Vector de inicialización

- Los IV no necesitan ser **secretos**, pero sí **impredecibles**
- ¡NO se deben reutilizar IVs con la misma clave!
- Solución: guardar el IV utilizado al inicio del fichero

Vector de inicialización

Longitud recomendada

- Objetivo: no repetir nunca IV
- Mínimo: 16 bytes = 128 bits = 2^{128} posibles valores
- Idealmente, generar de forma aleatoria para cada proceso de cifrado: fichero, mensaje, etc...

Modo CBC

- Uso correcto:
  1. Generar un IV aleatorio
  2. Cifrar en modo CBC
  3. Concatenar IV con salida antes de guardar en disco

AES – Advanced Encryption Standard

- Nuevo estándar de cifrado desde 2000
- No es de tipo Feistel, sino que utiliza álgebra de cuerpos finitos (concretamente un cuerpo de Galois, $GF(2^8)$), aunque sí utiliza cajas S.
- Diseñado para ser eficiente en microprocesadores de cualquier ancho de palabra, desde 8 bits, usados en tarjetas inteligentes o microcontroladores, hasta CPUs de 64 bits.
- La NSA elige AES en 2003 para cifrar su propia información clasificada como secreto y alto secreto

AES – Advanced Encryption Standard

Base matemática	Número de etapas	Cajas S
• Operaciones algebraicas en cuerpos finitos • No es de tipo Feistel	• Flexible según necesidades del usuario.	• Usa un conjunto de Cajas S similares a las del DES.

AES – Advanced Encryption Standard

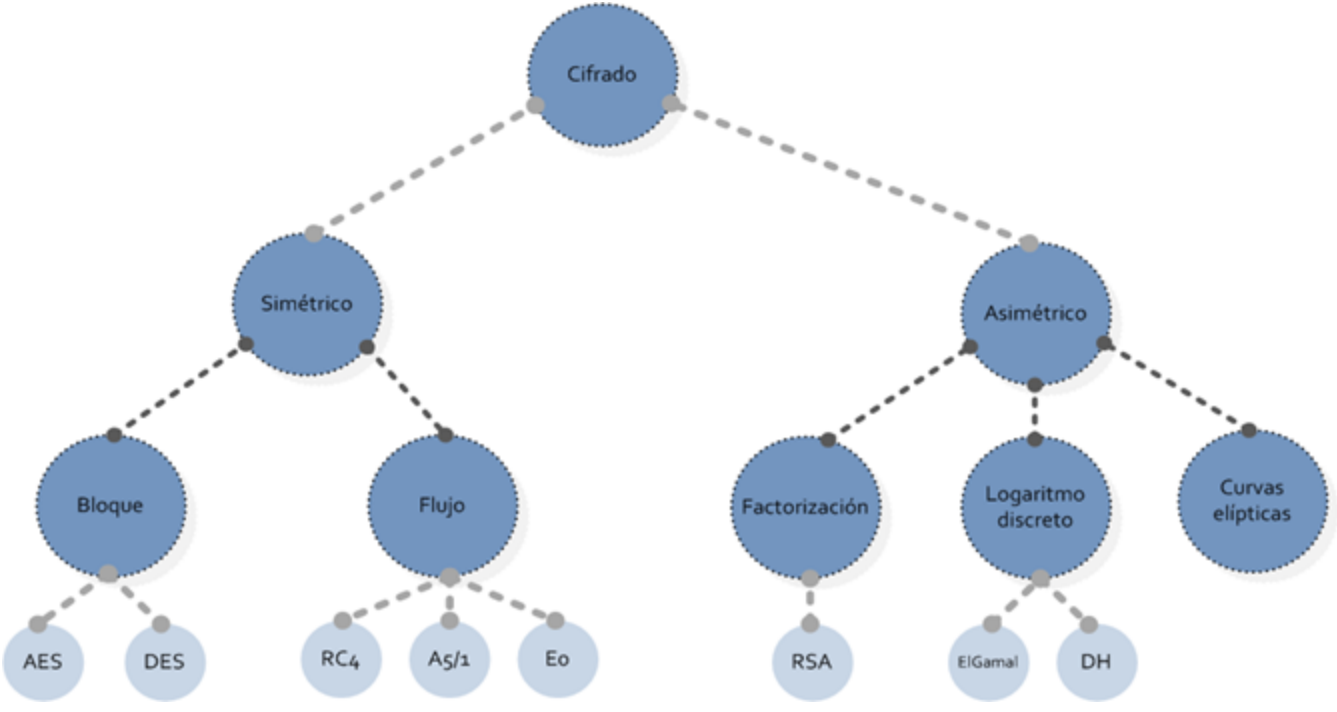
Tamaño de palabra	Tamaño de clave variable	Tamaño del bloque de texto
• 8 bits (tarjetas inteligentes y CPUs)	• 128, 192 y 256 bits (estándar) o bien múltiplo de 4 bytes.	• 128 bits o múltiplo de 4 bytes

Ataques por fuerza bruta

Longitud de la clave	Tiempo necesario para romper la clave
40 bits	2 segundos
48 bits	9 minutos
56 bits	40 horas
64 bits	14 meses
72 bits	305 años
80 bits	78.250 años
96 bits	5.127.160.311 años
112 bits	336.013.578.167.538 años
128 bits	22.020.985.858.787.784.059 años
192 bits	$1.872 * 10^{37}$ años
256 bits	$9.1 \cdot 10^{50}$ años

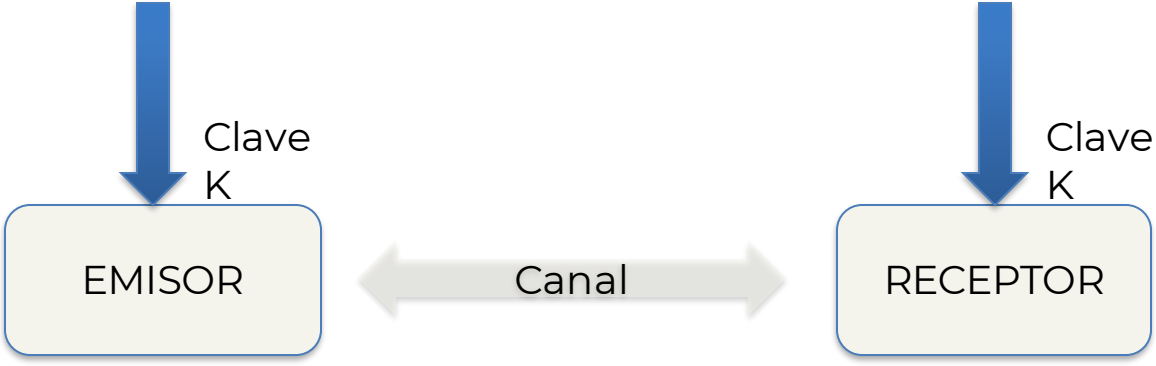
CIFRADO ASIMÉTRICO
Repaso

Taxonomía


Cifrado
Simétrico
Asimétrico
Bloque
Flujo
Factorización
Logaritmo discreto
Curvas elípticas
AES
DES
RC4
A5/1
E0
RSA
ElGamal
DH

Cripto simétrica

Clave simétrica



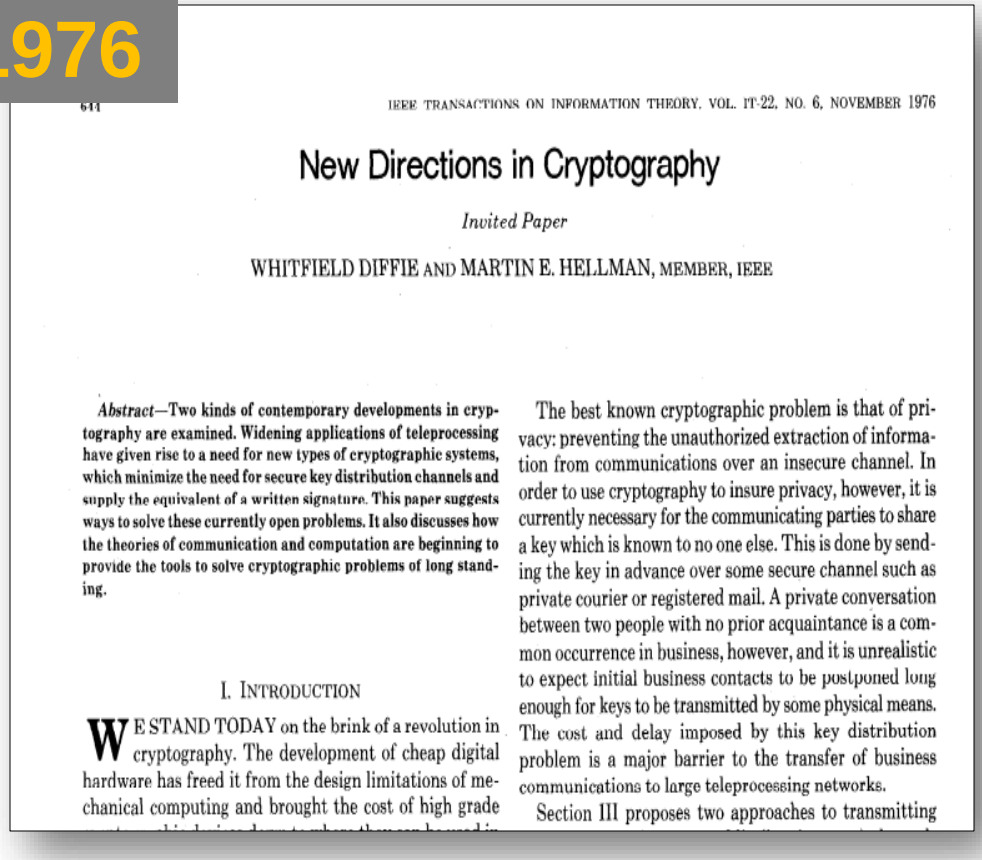
Inconveniente: distribución de claves

Historia de la cripto asimétrica

1976

Historia de la cripto asimétrica

1976



New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

Abstract—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.

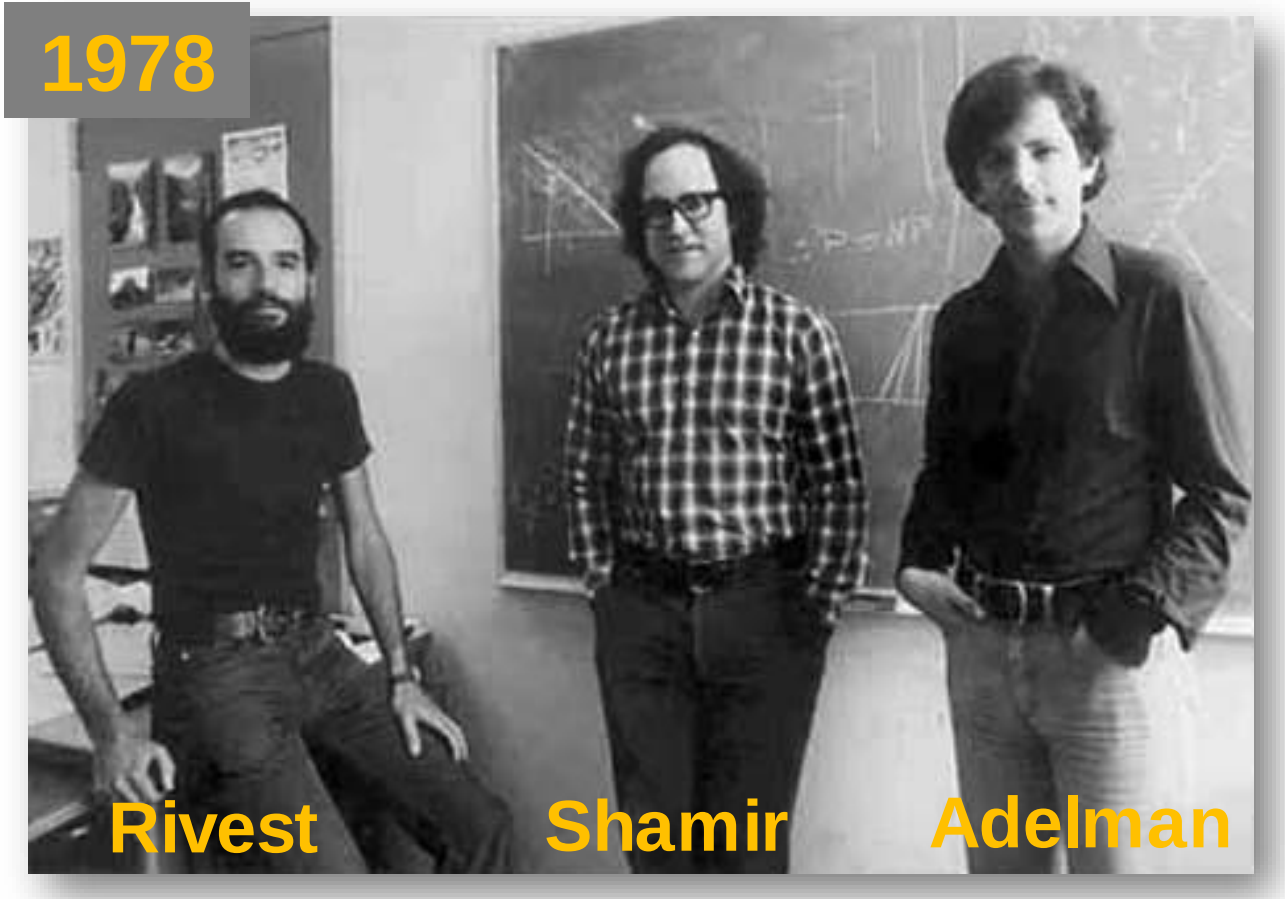
I. Introduction

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting

Historia de la cripto asimétrica

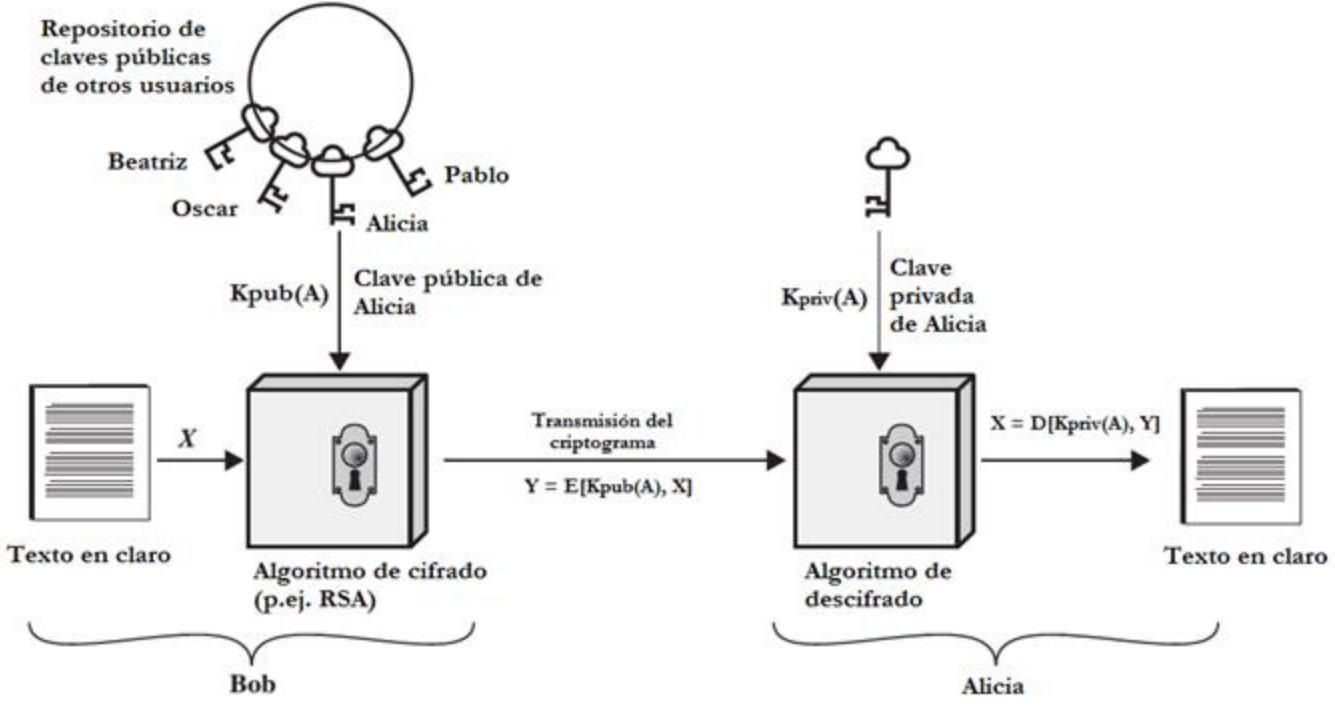


Elementos básicos

- Dos claves, denominadas clave pública y privada, vinculadas matemáticamente.

Principio básico:
"Lo que cifra con una de las claves, sólo se puede descifrar con la otra"

Elementos básicos



Algoritmo RSA

Factorización

$$p \cdot q = n$$

Dado n, ¿cuáles son p y q?

Algoritmo RSA

Aritmética modular

Si trabajamos en $Z_n = \{0,1,\ldots, (n-1)\}$ se cumplen las siguientes propiedades:

$$(a + b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n \quad [1]$$

$$(a - b) \bmod n = [(a \bmod n) - (b \bmod n)] \bmod n \quad [2]$$

$$(a \cdot b) \bmod n = [(a \bmod n) \cdot (b \bmod n)] \bmod n \quad [3]$$

Algoritmo RSA

Aritmética modular

Ejemplos:

$11 \bmod 8 = 3;\ 15 \bmod 8 = 7$

$[(11 \bmod 8) + (15 \bmod 8)] \bmod 8 = [3 + 7] \bmod 8 = 10 \bmod 8 = 2$
$(11 + 15) \bmod 8 = 26 \bmod 8 = 2$

$[(11 \bmod 8) \cdot (15 \bmod 8)] \bmod 8 = [3 \cdot 7] \bmod 8 = 21 \bmod 8 = 5$
$(11 \cdot 15) \bmod 8 = 165 \bmod 8 = 5$

Algoritmo RSA

Aritmética modular

Ejemplos (exponenciación)

Para encontrar 11^7 mod 13, podemos hacer:

11^7 mod 13 = 19487171 mod 13 (Costoso)

O utilizar las propiedades anteriores:

Sabemos que $11^7 = 11^4 \cdot 11^2 \cdot 11$ y por la prop. [3]:
$11^2 = 121 \bmod 13 = 4 \bmod 13$
$11^4 = (11^2)^2 \bmod 13 = 4^2 \bmod 13 = 3 \bmod 13$
$11^7 = (11 \cdot 4 \cdot 3) \bmod 13 = 132 \bmod 13 = 2 \bmod 13$

Algoritmo RSA

Fundamentos matemáticos

Función □ de Euler

□*(n)* = número de enteros positivos **coprimos** con *n*

Algoritmo RSA

Generación de claves

1. p, q, dos números primos grandes
2. $n = pq$, $\varphi(n)=(p-1)\cdot(q-1)$
3. Seleccionar e, con $mcd(\varphi(n), e) = 1; 1 < e < \varphi(n)$
4. Buscar $d \equiv e^{-1} \pmod{\varphi(n)}$ $d\cdot e \equiv 1 \pmod{\varphi(n)}$

Clave pública = {e,n}
Clave privada = {d,n}

Algoritmo RSA

Generación de claves - Ejemplo

1. Elegimos **p** = 3 y **q** = 11
2. Calculamos **n** = **p**·**q**=3·11=33.
3. Calculamos □(n)=(p-1)·(q-1)=2·10=20.
4. Elegimos **e** tal que 1<**e**< □(n) y **e** y **n** sean primos entre sí. Por ejemplo, **e** = 7.
5. Calculamos un valor para **d** tal que (**d**·7)□ 1 (mod 20). Una solución es **d**=3.

1. La clave pública es {**e**,**n**} = {7,33}
2. La clave privada es {**d**,**n**} = {3,33}

Algoritmo RSA

Proceso de cifrado/descifrado

A cifra un mensaje m para B

1. **Cifrado**. A debe:
   a. Obtener la clave pública de B, (e,n)
   b. Representar el mensaje con un entero *m* en el intervalo [0, n-1]
   c. Calcular $c=m^e \pmod n$ y enviar a B.
2. **Descifrado**. Para recuperar el texto en claro de *c*, B debe:
   a. Usar su clave privada *d*, y calculary $m=c^d \pmod n$

Algoritmo RSA

Ejemplo de cifrado

La clave pública es {**e**,**n**} = {7,33}
La clave privada es {**d**,**n**} = {3,33}

1. Cifrado del mensaje: USA TU ARMA!
2. Codificamos cada símbolo de forma numérica (ASCII, por ejemplo)
3. Ciframos cada símbolo (U=85 en ASCII)

$$85=19^{7}(\text{mod } 33)= 13$$

1. El descifrado es, simplemente:

$$13^{3}(\text{mod } 33)=19$$

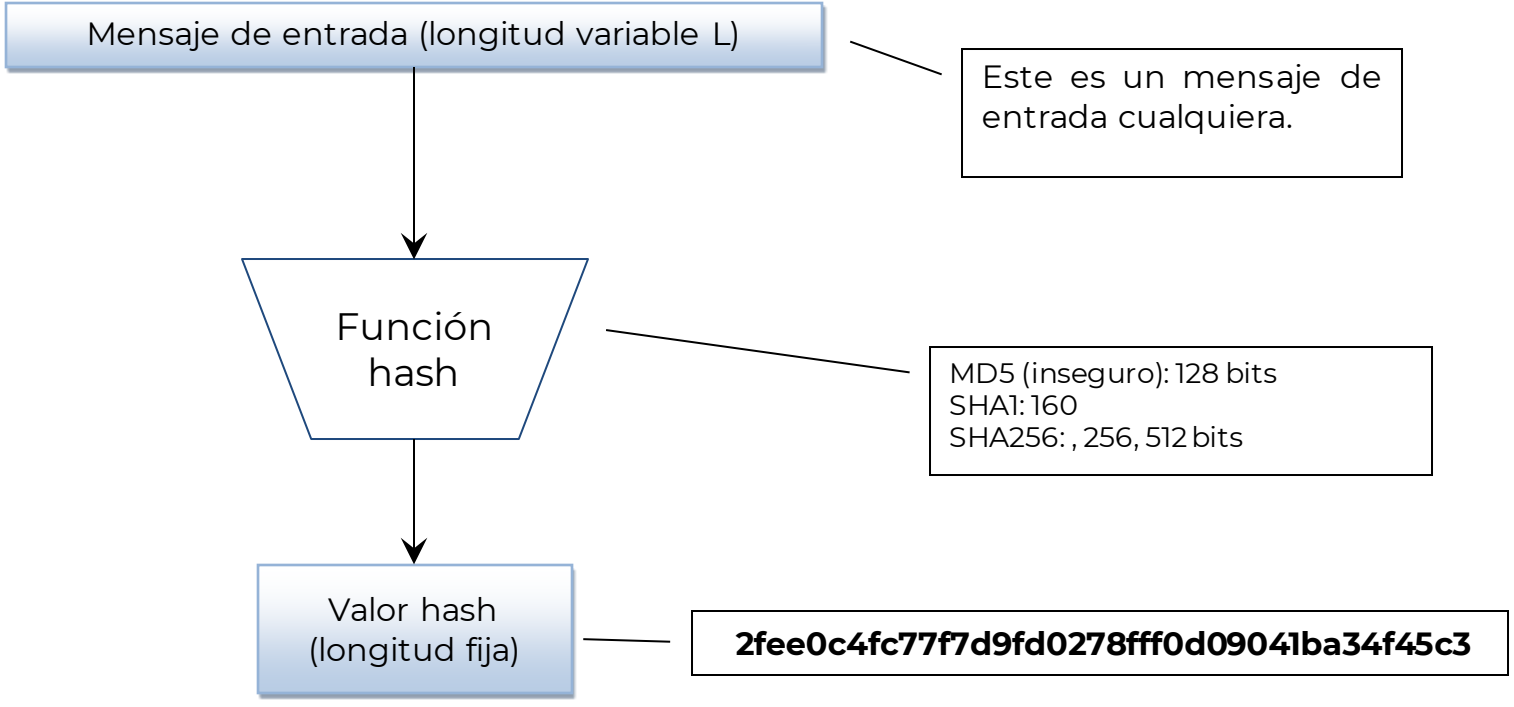
Algoritmo RSA

Disclaimer

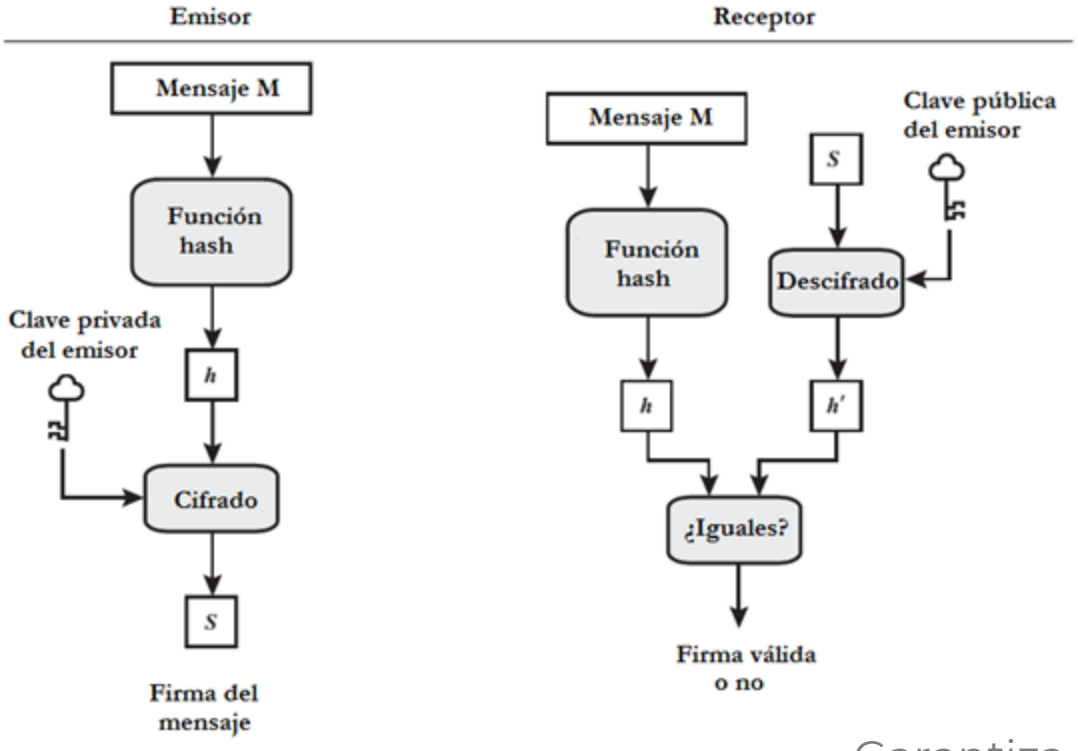
1. ¡Hemos visto ejemplos MUY simplificados! NO ofrecen seguridad real.
2. Cifrando símbolo a símbolo convertimos el proceso es un simple cifrado de sustitución, atacable por un análisis de frecuencias.
3. Una posible solución es combinar varios números en bloques (entonces, **n** debe ser mayor que el mayor número posible del bloque)

En la práctica, RSA se utiliza en combinación con un cifrador simétrico, cifrando sólo su clave secreta.

Funciones hash

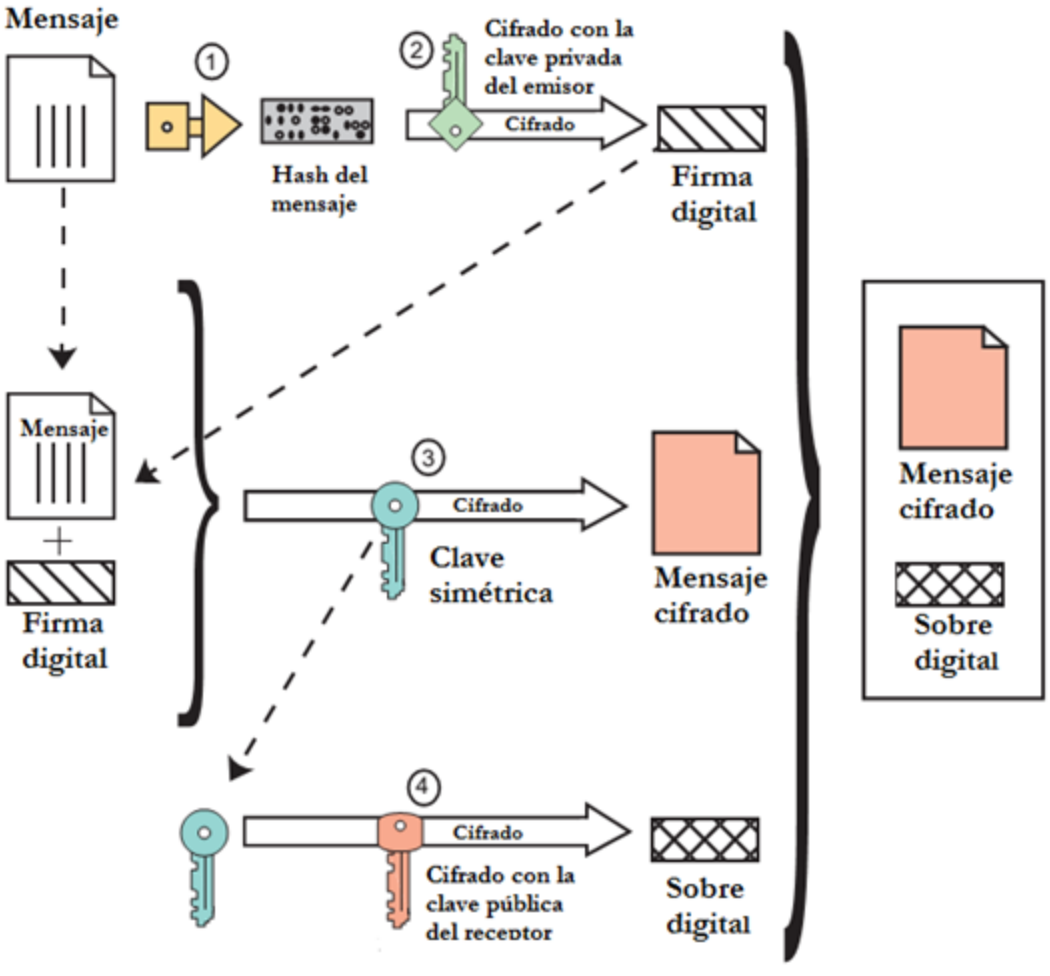


Firma digital



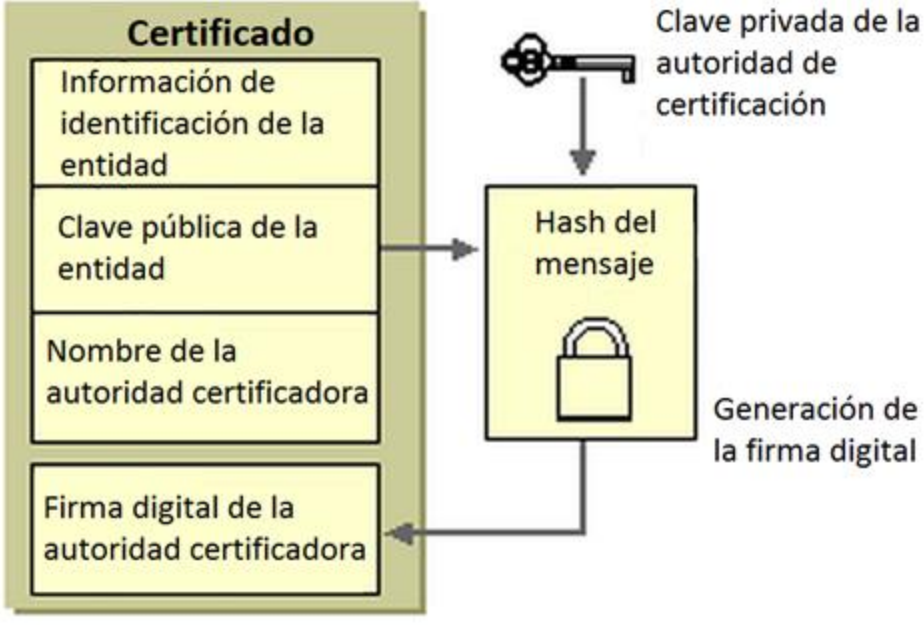
Garantiza el no repudio

Esquemas híbridos

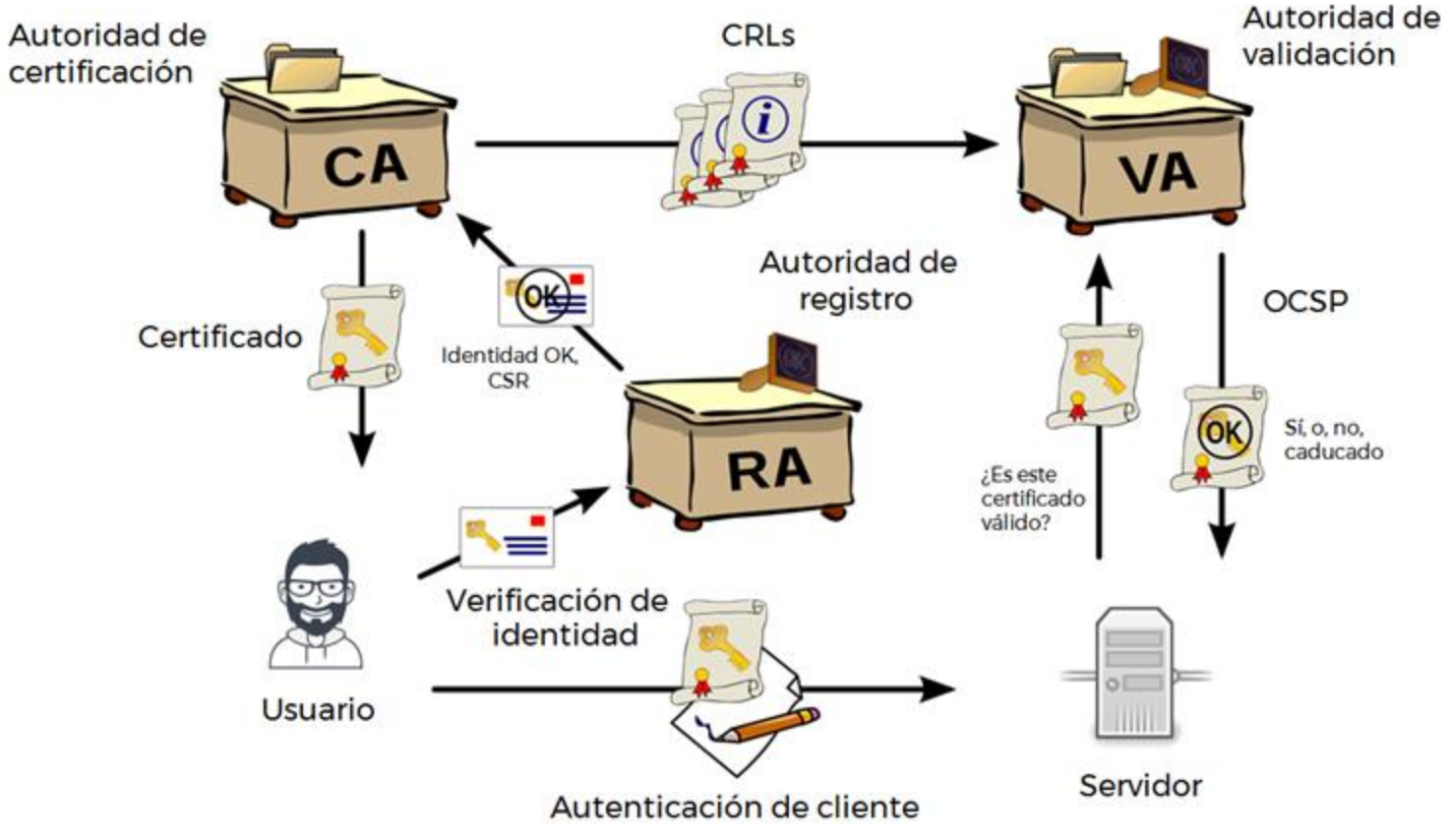


Certificados digitales y PKIs

Certificados digitales



PKI – Public Key Infraestructure

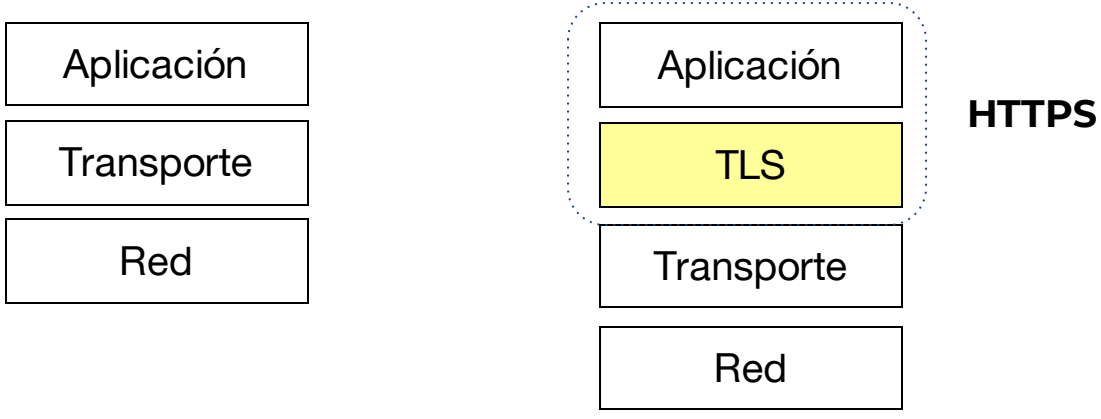


Seguridad Web // TLS

Transport Layer Security

- Protocolo que aporta **secreto**, **autenticidad** e **integridad** a las comunicaciones, típicamente utilizado sobre HTTP (HTTPS)
- TLS evolucionó sobre *Secure Sockets Layer* (SSL), desarrollado inicialmente por Netscape.
- Son protocolos diferentes, pero aún se suelen utilizar SSL como referencia genérica.

TLS // Protocolo de transporte



- TLS es un protocolo de transporte que puede proteger (casi) cualquier aplicación
- Es necesario que la app se adapte a TLS (no son *sockets* normales)

Negociación TLS

El proceso comienza con una negociación (*handshake*) C-S en la que:

1. Se especifica qué versión TLS usarán (TLS 1.0, 1.2, 1.3, etc.)
2. Deciden qué suite criptográfica soportan
3. El cliente autentica al servidor utilizando su certificado
4. Generan **claves de sesión** para cifrar las comunicación cuando acabe el *handshake*

Suite criptográfica

- Algoritmo de clave pública (RSA, EC)
- Algoritmo de clave secreta (AES)
- Algoritmo de hashing

- Es importante tener la oportunidad de "negociar", por si se descubre alguna vulnerabilidad en algún protocolo

- Negociación: el cliente ofrece opciones y servidor elige una (puede rechazar la conexión si todas son inseguras)

Algoritmos simétricos habituales

- AES – Advanced Encryption Standard
- ChaCha - cifrado en flujo

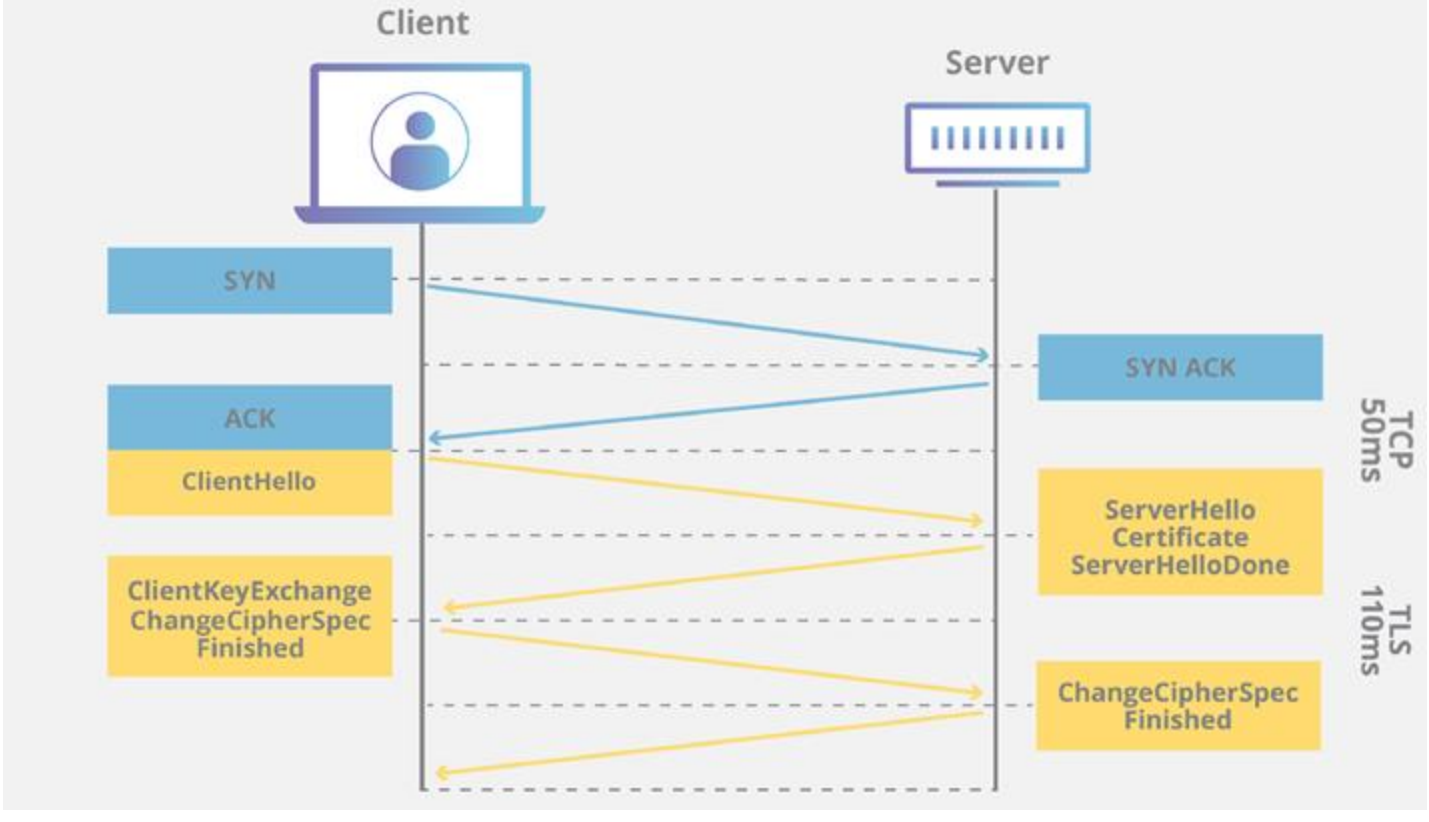
Clave pública

- Curvas elípticas (RSA se va abandonando porque son más lentas y posible riesgo del computador cuántico)

Hash

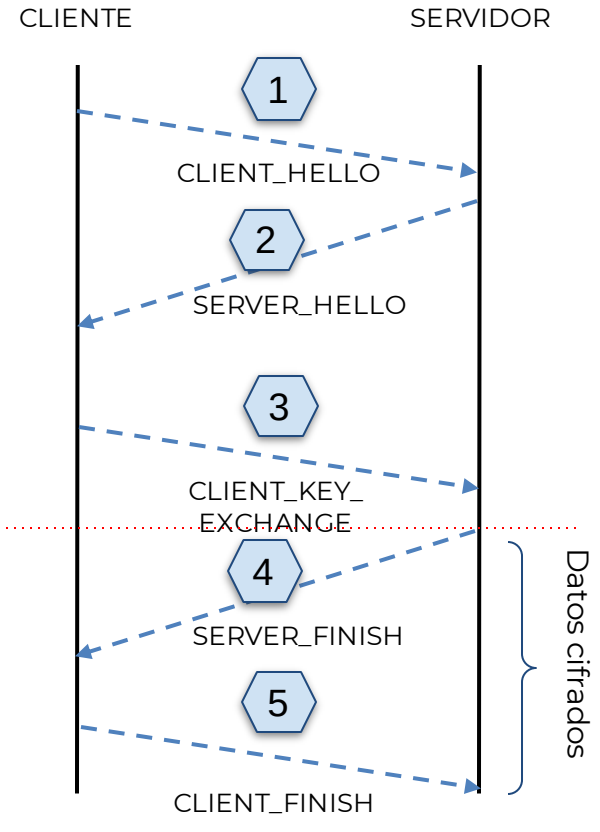
- Solo familia SHA2 y SHA3

Handshake // Alto nivel



Fuente: Cloudfare

Handshake // Detalle



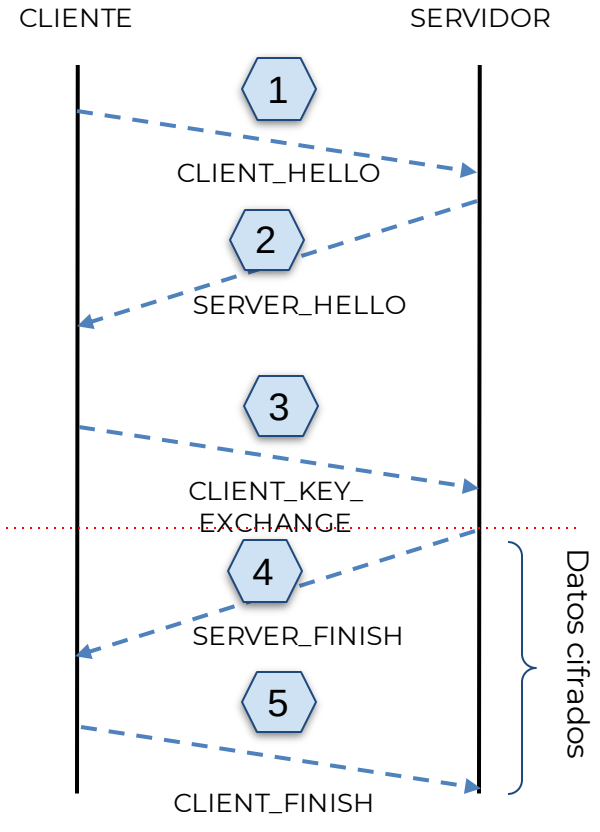
1. **CLIENT_HELLO**: El cliente envía la versión de TLS que quiere utilizar, su suite criptográfica y un número aleatorio (nonce), n_c
2. **SERVER_HELLO**: El servidor envía su certificado, los algoritmos elegidos, su propio nonce (n_s) y el nonce del cliente cifrado con su clave privada
3. **CLIENT_KEY_EXCHANGE**: el cliente:
   a. Comprueba el certificado del servidor
   b. Genera el *pre_master_secret* (un número aleatorio de 48 bytes) y lo cifra con la clave pública del servidor
4. **SERVER_FINISH**: el servidor deriva la **clave de sesión** (*master_secret*):

$$master\_secret = hash(pre\_master\_secret \,||\, "master\ secret" \,||\, n_c + n_s)\ [0..47]$$

y utiliza solo los primeros 48 bytes para cifrar la comunicación a partir de este punto con el algoritmo simétrico elegido. Añade un MAC de todos los mensajes anteriores

1. **CLIENT_FINISH**: deriva la misma **clave de sesión** y añade su MAC de los mensajes anteriores

Handshake // Detalle



- Los **nonces** son números aleatorios, que sirven para evitar el **ataque de repetición** (*replay attack*): un atacante captura los mensajes intercambiados y simplemente los repite en un momento posterior

- Los mensajes 4 y 5 “firman” todo el proceso y los vinculan con los nonces utilizados

- El mensaje CLIENT_KEY_EXCHANGE es un tipo sencillo de primitiva llamada **Transporte de claves**

ACTIVIDAD

TLS handshake

Vamos a ver la negociación TLS en acción. Para ello, podemos utilizar el comando ‘openssl’

```
# openssl s_client -connect amazon.es:443
```

Podemos ver más detalle, cómo evoluciona el estado de cliente y servidor con el flag -state:

```
# openssl s_client -state -connect amazon.es:443
```

Preguntas

1. ¿Cuál es la versión TLS finalmente negociada? ¿Quién la ha elegido?
2. Observa los campos Cipher y Master-key

Sellado de tiempo

Sellado de tiempo

- Permite probar que un dato *existía en un momento dado del tiempo*, y que no ha sido modificado desde entonces

- Esencial para la **firma digital**, pues ésta, *per se*, no contiene información sobre el momento de su creación

Sellado de tiempo

- Idealmente, la marca de tiempo debe ser generada por una parte diferente a quien realiza la firma: *Time Stamp Authority* (TSA)

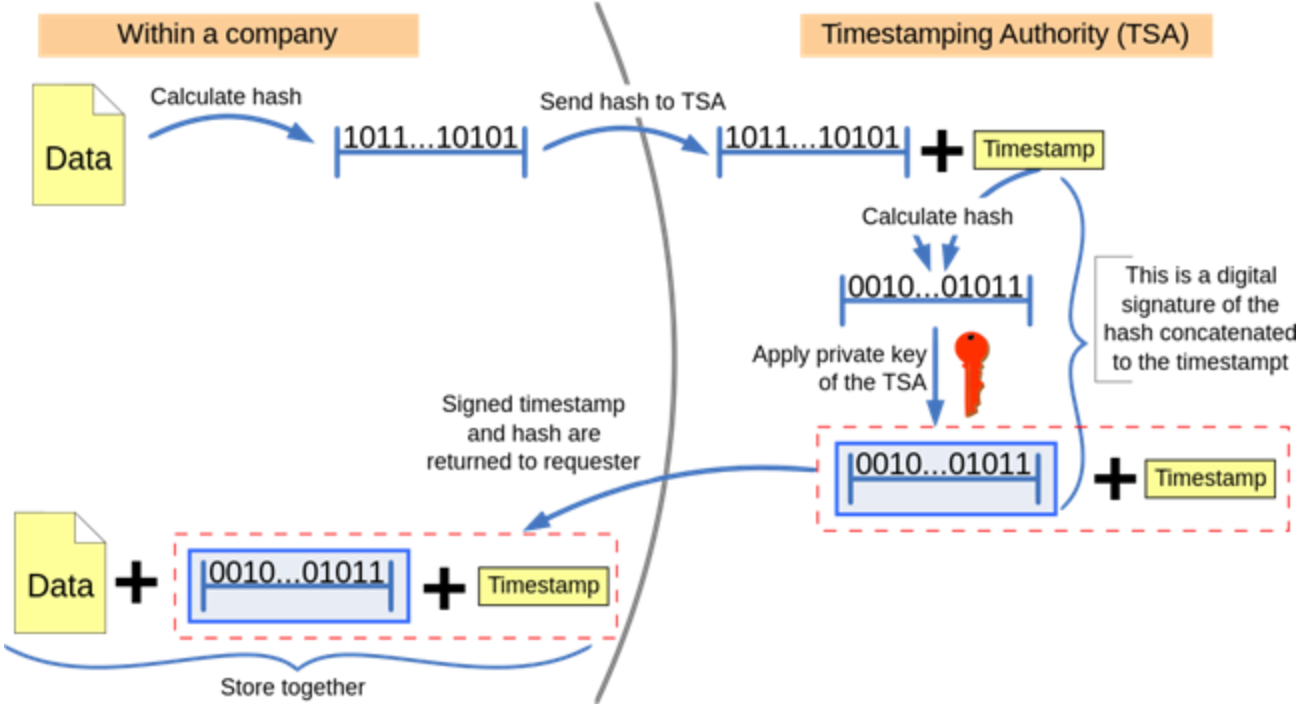
- La FNMT es también TSA, utilizando una fuente de tiempo segura: la del Real Observatorio de la Armada, fuente oficial de la hora legal en España (protocolo NTP)

Sellado de tiempo // Aplicaciones

Firmas digitales	Notaría digital	Análisis forense
• Aumenta su seguridad, al incluir referencia temporal	• Prueba de existencia de unos datos en un momento del tiempo • Protección de copyright	• Cadena de custodia de logs, ficheros, etc..

Sellado de tiempo

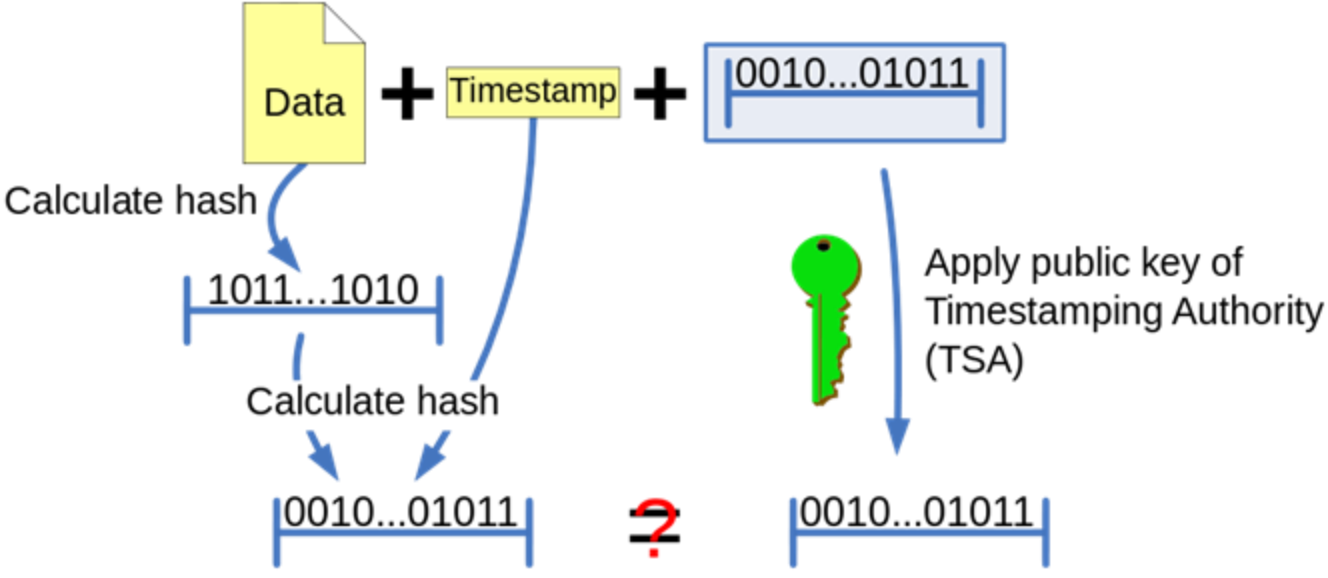
Generación



Fuente: Cryptography and Network Security, William Stallings

Sellado de tiempo

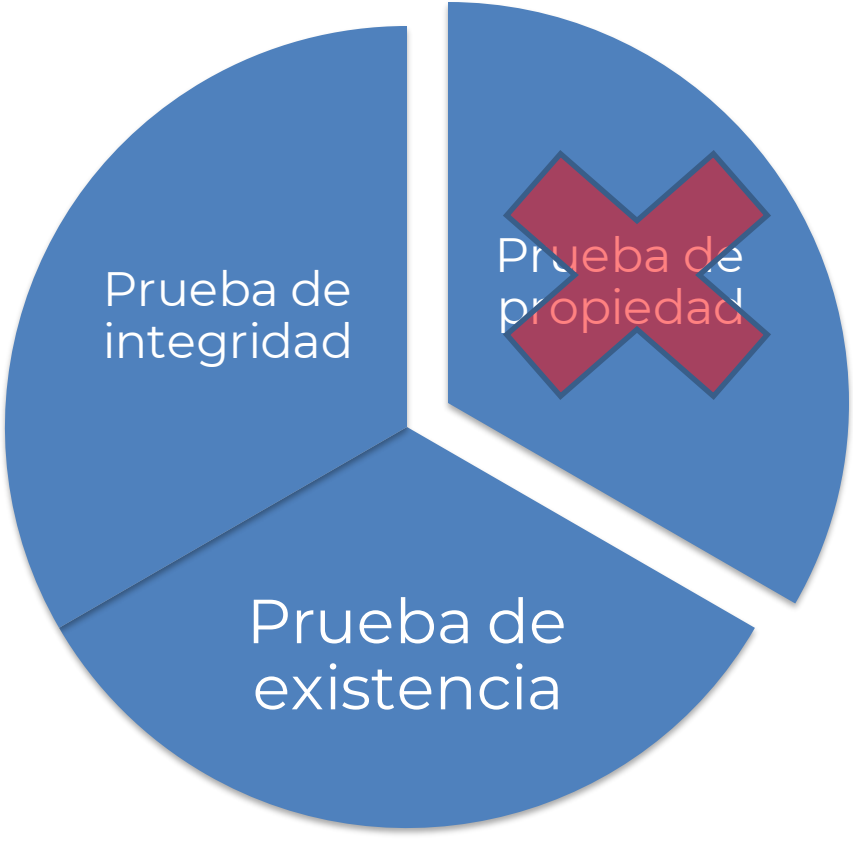
Verificación



Fuente: Cryptography and Network Security, William Stallings

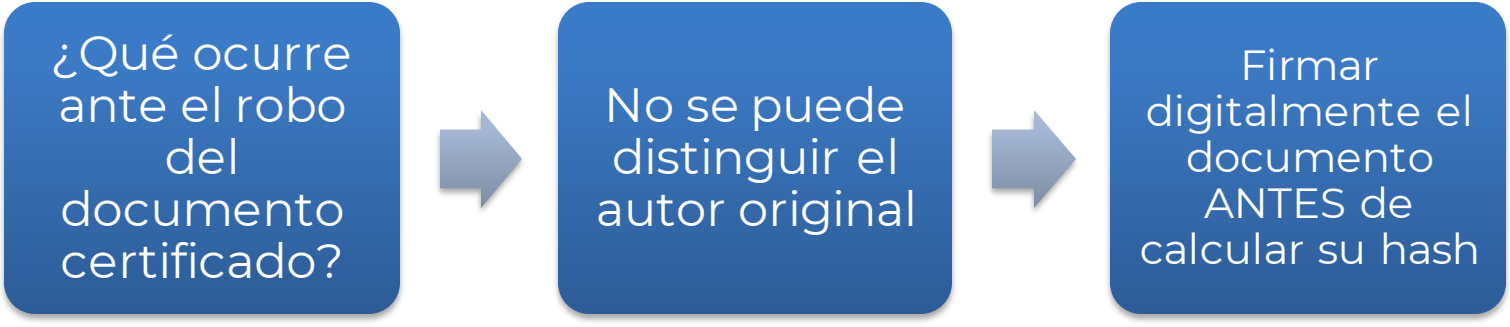
Sellado de tiempo

Limitaciones

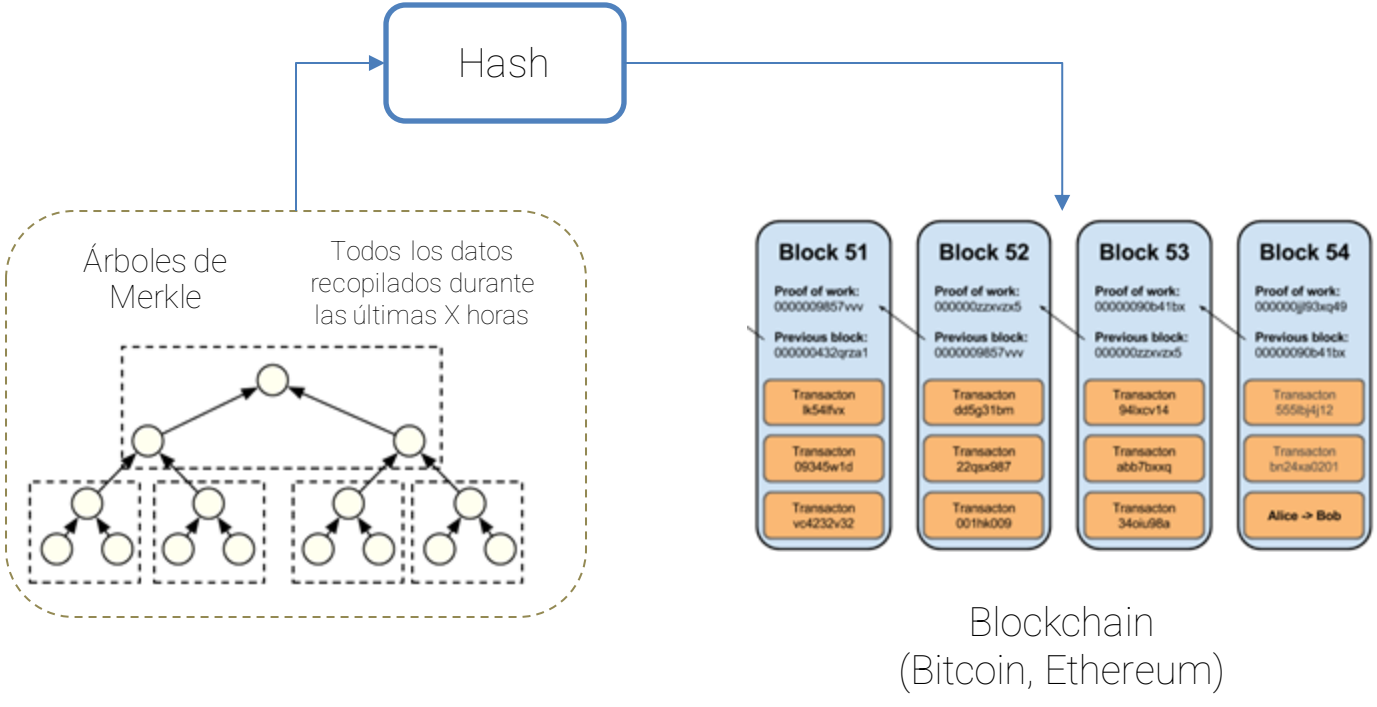


Sellado de tiempo

Prueba de propiedad



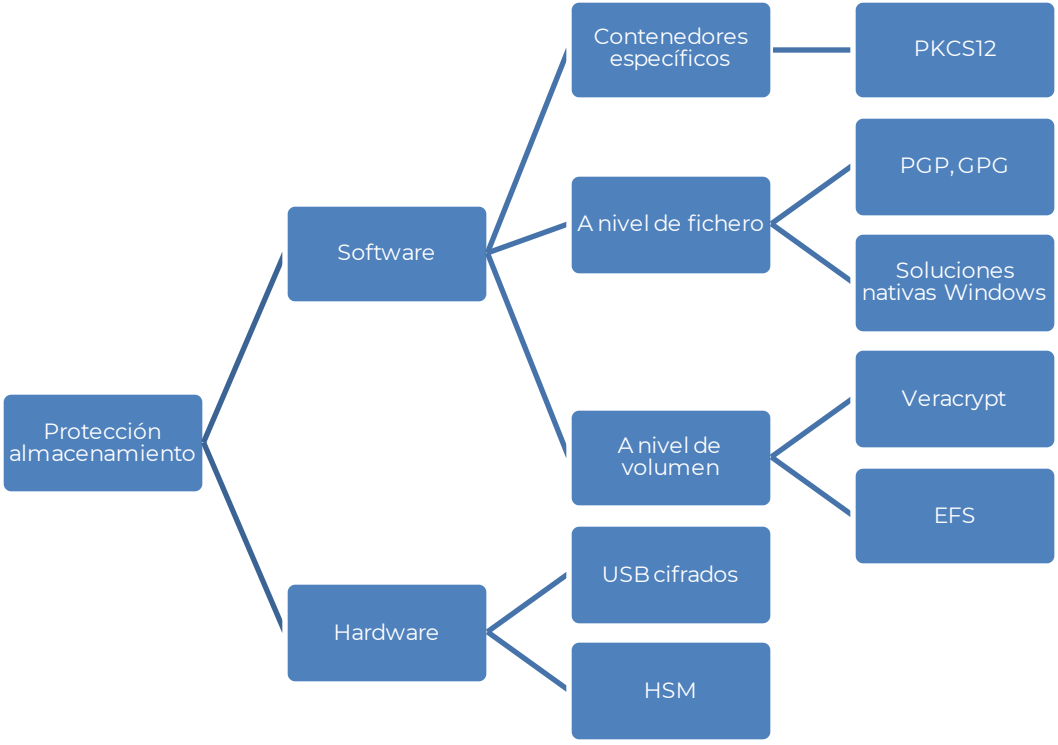
Certificación basada en *blockchain*



Blockchain
(Bitcoin, Ethereum)

Almacenamiento seguro de certificados y claves

Protección de claves y certificados



Protección por software

Ventajas

Utilizar contenedores criptográficos específicos, como PKCS#12

Se pueden utilizar tanto para unidades de almacenamiento interno como externo, como USBs o discos duros

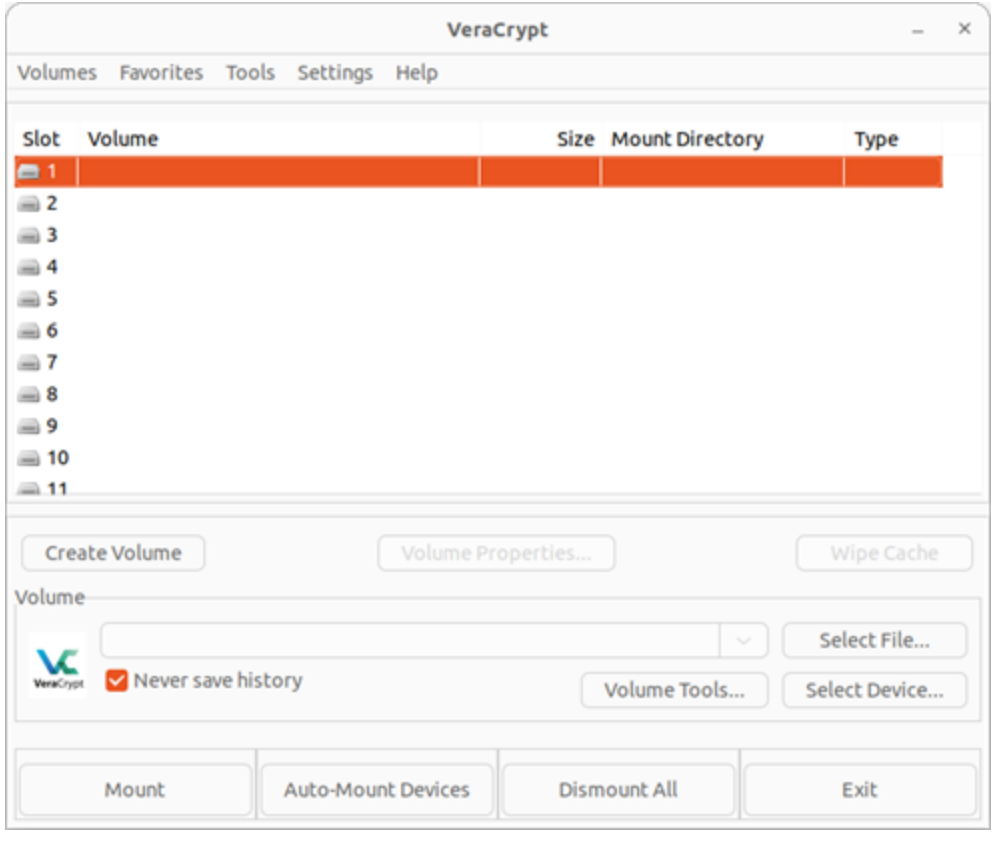
Protección por software

Inconvenientes

Necesario instalar software en los ordenadores desde los que se desee almacenar información “segura” y recuperarla

Dependen de un único secreto (contraseña) para su protección

Veracrypt


VeraCrypt
Volumes
Favorites
Tools
Settings
Help
Slot
Volume
Size
Mount Directory
Type
1
2
3
4
5
6
7
8
9
10
11
Create Volume
Volume Properties...
Wipe Cache
Volume
Select File...
Never save history
Volume Tools...
Select Device...
Mount
Auto-Mount Devices
Dismount All
Exit

VeraCrypt

- **Unidades virtuales:**
  - Generadas por el programa bajo demanda del usuario
  - Se comportan exactamente igual que los discos duros
  - Pueden ser montadas y desmontadas por el usuario

- **Contenedor:**
  - Tipo de fichero creado por el programa que pueden ser accedidos a través de una unidad virtual
  - Contienen información cifrada

VeraCrypt

- El *driver* de TrueCrypt procesa todas las peticiones que se realizan sobre una unidad virtual
- Todas aquellas peticiones que se realizan sobre información “no sensible” son realizadas directamente por el sistema operativo

VeraCrypt // Pre-boot Authentication

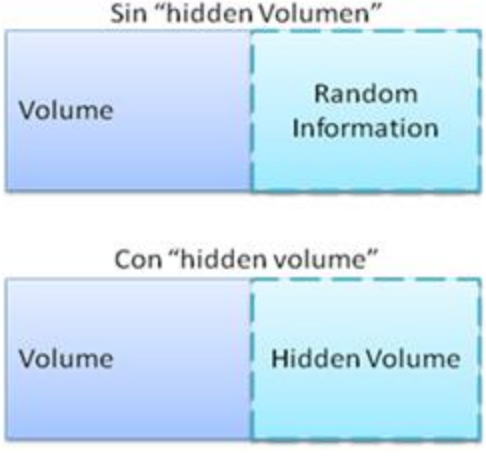
- Cifrado de todo el volumen en el que se encuentra un sistema instalado

- Alto nivel de seguridad ya que todos los archivos del sistema, como temporales, ficheros de hibernación, información sobre las aplicaciones ejecutadas, nombres y localizaciones de ficheros, etc., se mantienen cifrados

- VeraCrypt utiliza su propio Boot Loader para autenticar a los usuarios antes de la carga del sistema.

VeraCrypt // Volumen oculto

- Residen dentro de volúmenes convencionales, de forma que a simple vista no pueden ser detectados.
- Cuando se crea un volumen, TrueCrypt rellena el espacio escogido para ese volumen, con información aleatoria.
- La clave utilizada para este segundo volumen escondido debe ser completamente diferente a la utilizada para el volumen convencional.



Protecciones hardware

Wallets específicas

Soluciones hardware

✓ Ventajas

- Generalmente no necesitan de ninguna instalación
- Se conectan como unidades de almacenamiento convencionales

X Desventajas

- Menos flexible, ya que el software puede ser instalado en distintos tipos de dispositivos

Hardware Security Modules (HSM)

- HSM gama de dispositivos que ofrecen gran seguridad a la información
- Seguridad a nivel físico y lógico

HSM – Medidas antimanipulación

- La seguridad de estos dispositivos se basa en sus extremas medidas físicas antimanipulación.
- Pueden clasificarse, básicamente, en medidas:
  - **Pasivas**: tratan de dificultar la inspección y manipulación de los componentes activos y las claves almacenadas.
  - **Activas**: detectan los intentos de intrusión e, inmediatamente, destruyen todas las claves criptográficamente sensibles.

HSM – Medidas antimanipulación pasivas

- La más sencilla es una simple carcasa de acero.
- Útil sólo si se combina con seguridad física de acceso (a un CPD, por ejemplo).
- El peso puede suponer también una medida disuasoria y algunos HSM incluyen peso extra a propósito.

HSM – Medidas antimanipulación pasivas

- Encapsulado de todo el dispositivo en material antimanipulación (resina epoxy):

HSM – Medidas antimanipulación activas

- Algunas de las medidas activas incluyen:
  - **Sensores térmicos**: protegen contra ataques que impliquen enfriamiento. El enfriamiento extremo provoca efectos de remanencia de datos en memorias RAM.
  - **Sensores de rayos X** : detectan este tipo de radiación y borran las claves antes de que éstas puedan ser “quemadas” en la RAM por la radiación.

HSM – Medidas antimanipulación activas

- **Sensores de luz**: borran las claves cuando detectan luz visible. Pueden ser engañados trabajando con luz ultravioleta.
- **Sensores de movimiento**: constituyen más una medida antirrobo que antimanipulación. Sólo son útiles en un CPD.
- **Membranas conductoras**: forman una malla alrededor del núcleo y son químicamente indistinguibles del encapsulado. Borran las claves en cuanto son expuestas.