Criptografía simétrica

Introducción

Isaac A. Canales Martínez

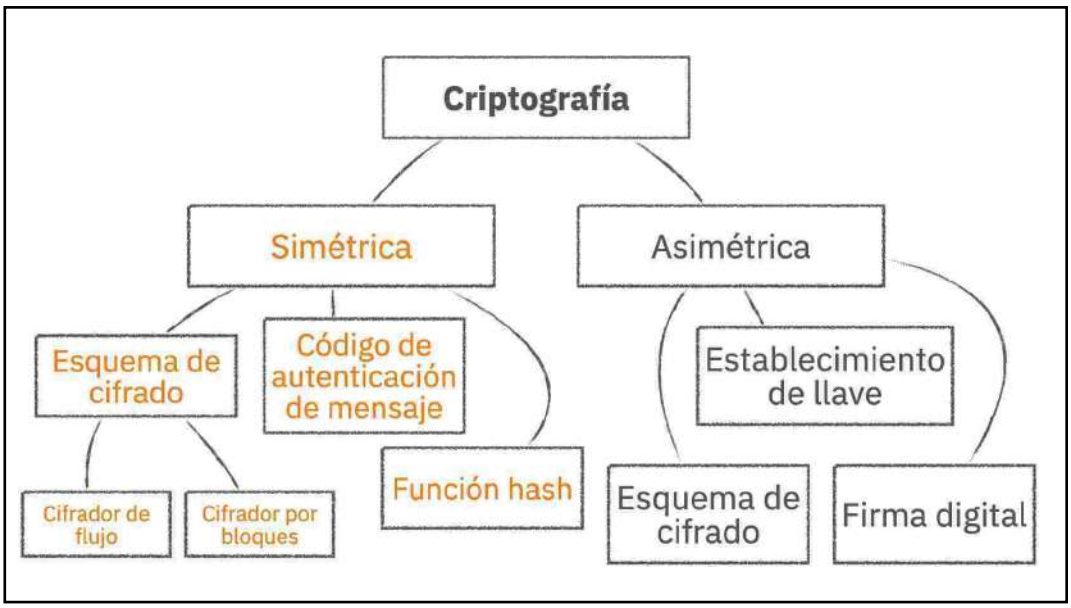
Septiembre 30, 2023

1

Criptografía

Estudio de técnicas para establecer comunicación segura en presencia de adversarios

2



3

Criptografía y servicios de seguridad

	Criptografía simétrica	Criptografía asimétrica
Privacidad	Esquema de cifrado	Esquema de cifrado
Integridad	Código de autenticación de mensaje	Firma digital
No repudio		Firma digital

4

Seguridad de un sistema criptográfico

- Seguridad se define con base en
  - Objetivo del adversario
  - Capacidades del adversario
- Sistema criptográfico no seguro: adversario logra su objetivo dadas sus capacidades
- Si no es el caso, el sistema se puede considerar seguro

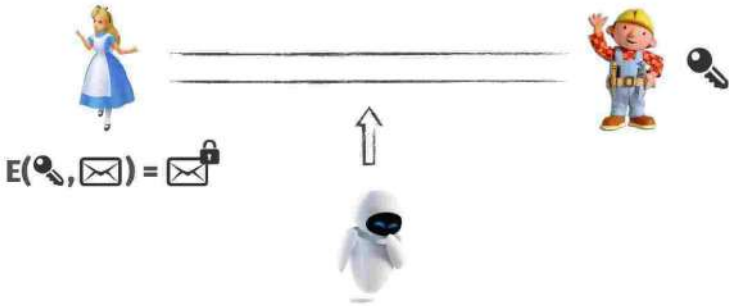
5

Criptografía simétrica

- Alicia y Bob acuerdan una llave común 🔑
  - La llave debe mantenerse secreta
  - Las operaciones criptográficas requieren esta llave
- Si Alicia desea comunicarse con alguien más, debe acordar otra llave común
- Establecer una llave puede ser un proceso complicado

6

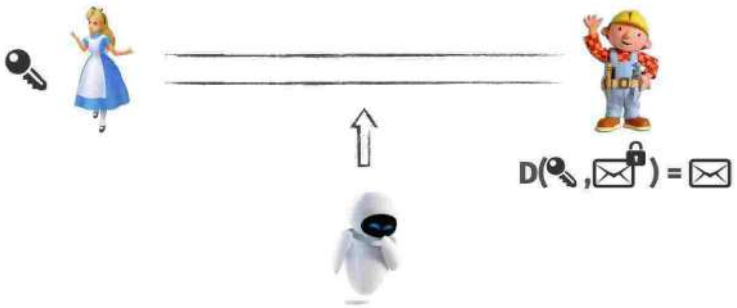
Cifrado



E( ) : algoritmo de cifrado

7

Cifrado



D( ) : algoritmo de descifrado

8

Cifrado

- Ofrece privacidad en la comunicación
- Sin embargo, no ofrece integridad (evitar que el mensaje sea modificado)

9

Esquema de cifrado

- G( ): Generación de llave (probabilista)
  - Salida: llave k
- E( ): Algoritmo de cifrado (probabilista)
  - Entrada: llave k y mensaje p
  - Salida: texto cifrado c
- D( ): Algoritmo de descifrado (determinista)
  - Entrada: llave k y texto cifrado c
  - Salida: mensaje p

Para todas las llaves k y mensajes p:

$$p = \mathrm{D}(k, \mathrm{E}(k, p))$$

10

Una idea relacionada: One-time pad

Cifrado

$$k = \boxed{k_1}\ \boxed{k_2}\ \boxed{k_3}\ \ldots\ \boxed{k_n}$$

$$\oplus$$

$$p = \boxed{p_1}\ \boxed{p_2}\ \boxed{p_3}\ \ldots\ \boxed{p_n}$$

$$c = \boxed{c_1}\ \boxed{c_2}\ \boxed{c_3}\ \ldots\ \boxed{c_n}$$

11

Una idea relacionada: One-time pad

Descifrado

$$k = \boxed{k_1}\ \boxed{k_2}\ \boxed{k_3}\ \ldots\ \boxed{k_n}$$

$$\oplus$$

$$c = \boxed{c_1}\ \boxed{c_2}\ \boxed{c_3}\ \ldots\ \boxed{c_n}$$

$$p = \boxed{p_1}\ \boxed{p_2}\ \boxed{p_3}\ \ldots\ \boxed{p_n}$$

12

Una idea relacionada: One-time pad

- k se selecciona de manera aleatoria
- Longitud de k = longitud de p
- k (o una parte de k) nunca se reusa y siempre se mantiene secreta
- Esquema de cifrado perfectamente seguro
  - c no revela información de p
- Uso no práctico: llave nueva por mensaje, longitud de la llave

13

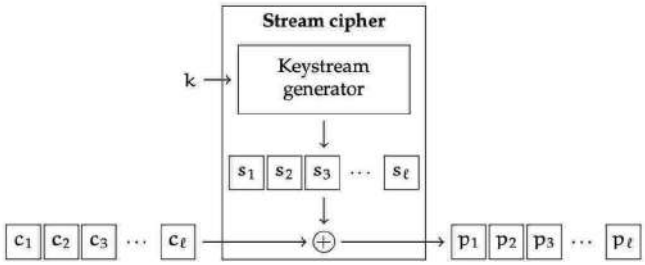
Cifrador de flujo

- Misma idea que one-time pad
- k genera una sucesión pseudo-aleatoria de bits s
  - Es difícil distinguir s de una sucesión realmente aleatoria
- s actúa como la llave de uso único

14

Cifrador de flujo

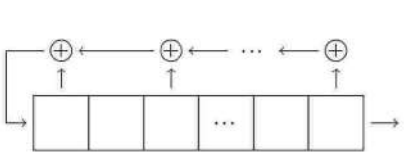
Descifrado



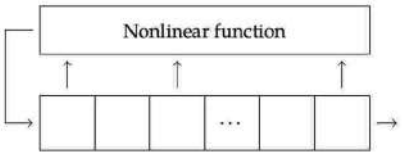
15

Componentes en el diseño de cifradores de flujo

Shift registers (registro de desplazamiento)



LFSR: Linear feedback shift register



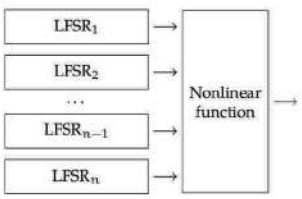
NFSR: Nonlinear feedback shift register

* Linealidad no es buena en diseños criptográficos

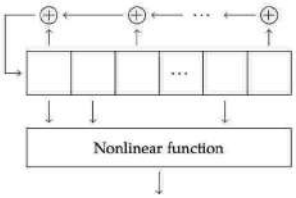
16

Componentes en el diseño de cifradores de flujo

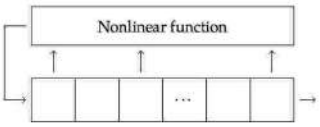
Keystream generator



Combination generator



Filter generator



NFSR

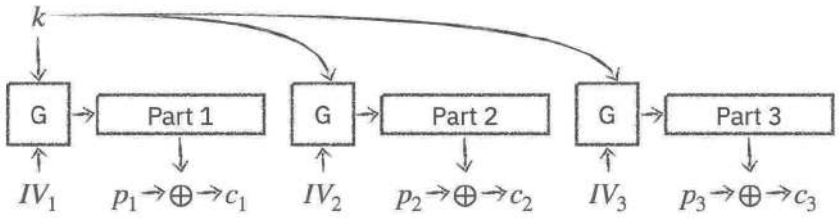
17

Cifrador de flujo

- Cifrado y descifrado se realiza siguiendo un *modo de operación*
- Un *vector de inicialización* (IV) es una sucesión de bits para hacer que el cifrado sea aleatorio

18

Modos de operación - No sincronizado



G: Keystream generator

19

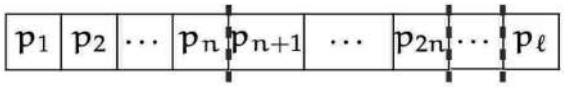
Uso de cifradores de flujo

- Operaciones de cifrado/descifrado son rápidas
- En general, simple implementación
- Adecuados en aplicaciones con restricciones en consumo de energía y espacio en hardware
- Ejemplos
  - eSTREAM portfolio: HC-128, Rabbit, Salsa20, SOSEMANUK, Grain, MICKEY, Trivium
  - Others: Grain-128, SNOW, A5/1, A5/2, RC4

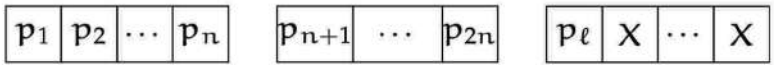
20

Cifrador por bloques

- Cifrado y decifrado se hace en bloques de símbolos
- p se divide en bloques de longitud n



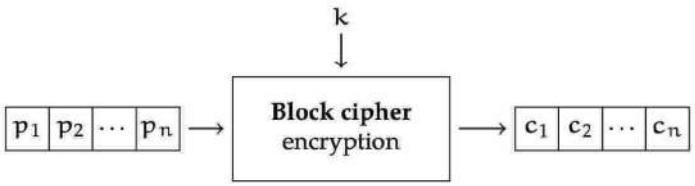
- Si es necesario, el último bloque se extiende hasta que su longitud sea n (padding)



21

Cifrador por bloques

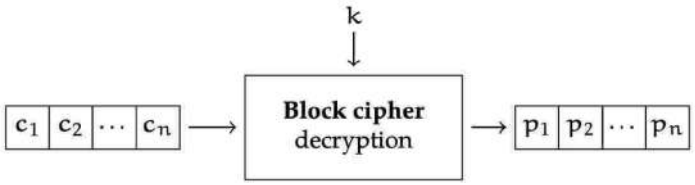
Cifrado



22

Cifrador por bloques

Descifrado



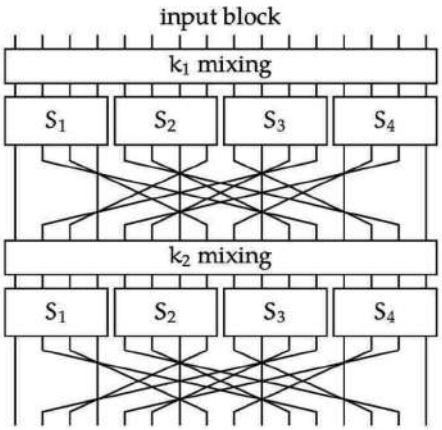
23

Diseño de cifradores por bloques

- Importante que cada bit del bloque de entrada afecte a distintos bits del bloque de salida
  - Idealmente, todos los bits
- *Confusión y difusión*:
  - Diseño donde una capa mezcla los bits y otra los revuelve
  - Una *ronda* es la aplicación consecutiva de estas capas
  - Llaves por ronda k_i se derivan de la llave k
  - Ejemplos: Red de sustitución y permutación (SPN), red Feistel

24

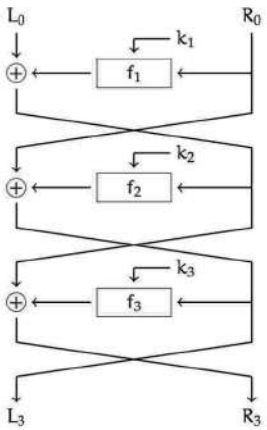
Red de sustitución y permutación (SPN)



- S_i funciones no lineales

25

Red Feistel



- f_i funciones no lineales

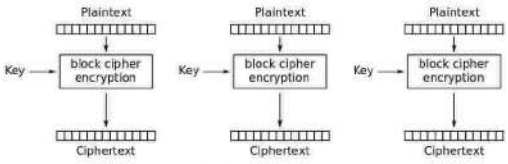
26

Cifrador por bloques

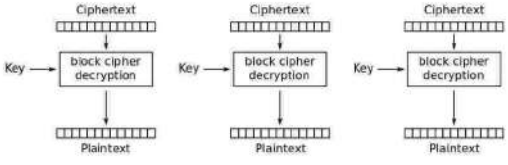
- Los bloques se cifran/descifran de acuerdo a un *modo de operación*
- Un *vector de inicialización* (IV) es un bloque para hacer que el cifrado sea aleatorio

27

Modos de operación - ECB



Electronic Codebook (ECB) mode encryption

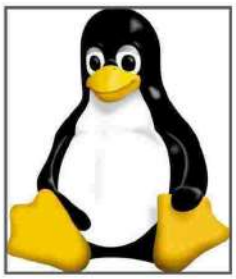


Electronic Codebook (ECB) mode decryption

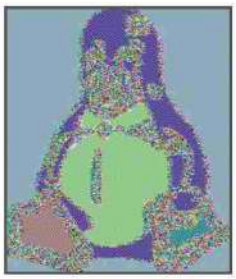
Imágenes: Dominio público, vía Wikimedia Commons.

28

Modos de operación - ECB



cifrado con ECB

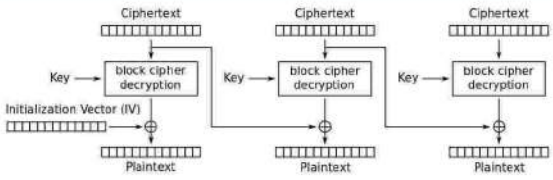


- ECB es inseguro: cifrado no es probabilistic y no hay difusión de información

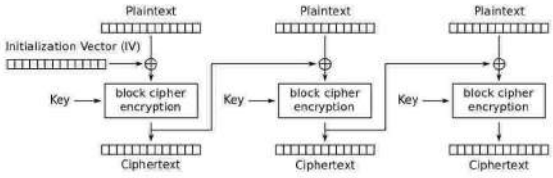
Imágenes:
Larry Ewing, Simon Budig (http://www.home.unix-ag.org/simon/penguin/README), Garrett LeSage, via Wikimedia Commons.
Larry Ewing, Garrett LeSage, Rumpelstilzchen 666 (CC BY-SA 4.0 https://creativecommons.org/licenses/by-sa/4.0/deed.en), via Wikimedia Commons.

29

Modos de operación - CBC



Cipher Block Chaining (CBC) mode decryption



Cipher Block Chaining (CBC) mode encryption

Imágenes: Dominio público, via Wikimedia Commons.

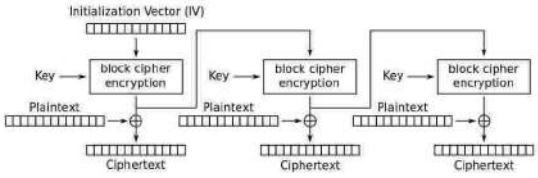
30

Modos de operación - CBC

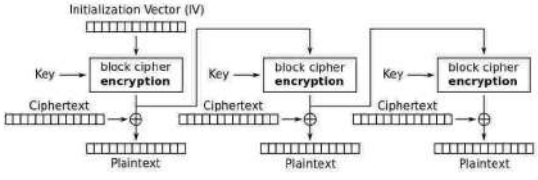
- *Modo de operación seguro*
- Cifrado se debe hacer de manera secuencial

31

Modos de operación - OFB



Output Feedback (OFB) mode encryption



Output Feedback (OFB) mode decryption

Imágenes: Dominio público, vía Wikimedia Commons.

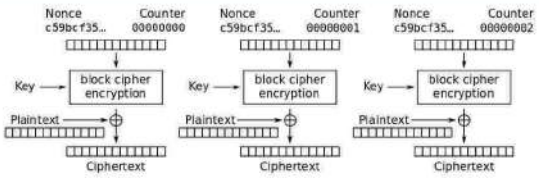
32

Modos de operación - OFB

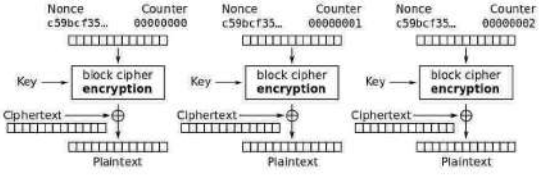
- *Modo de operación seguro*
- No es necesario utilizar el algoritmo de descifrado
- Cifrado se debe hacer de manera secuencial
- Comportamiento como un cifrado de flujo
- La sucesión pseudo-aleatoria se puede obtener en pre-cómputo

33

Modos de operación - CTR



Counter (CTR) mode encryption



Counter (CTR) mode decryption

Imágenes: Dominio público, vía Wikimedia Commons.

34

Modos de operación - CTR

- *Modo de operación seguro*
- También se puede ver como un cifrado de flujo
- Cifrado y descifrado permiten paralelismo

35

Cifradores por bloques

- Más populares comparados con cifradores de flujo
- Usados como cifradores de flujo, construcción de funciones hash, códigos de autenticación de mensajes, etc.
- Ejemplos
  - AES (Rijndael), Twofish, Blowfish, Serpent, DES

36

AES - Advanced Encryption Standard

- Estructura: SPN
- Tamaño de llave: 128, 192 o 256 bits
- 10, 12 o 14 rondas (depende, respectivamente, del tamaño de la llave)

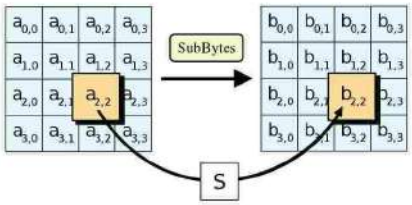
37

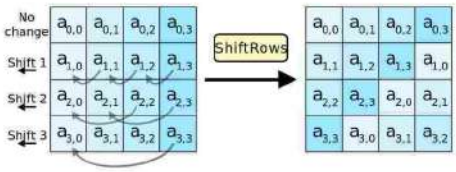
AES - Advanced Encryption Standard

1. KeyExpansion
2. AddRoundKey
3. Rondas (9, 11 o 13)
    1. SubBytes
    2. ShiftRows
    3. MixColumns
    4. AddRoundKey
4. Ronda final
    1. SubBytes
    2. ShiftRows
    3. AddRoundKey

38

AES - Advanced Encryption Standard

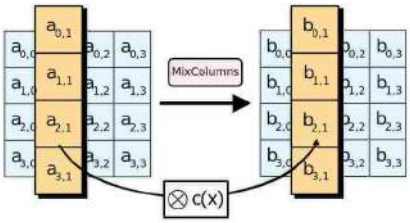


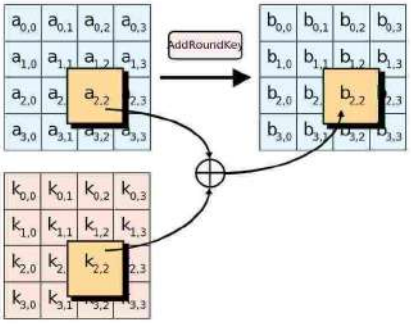


Imágenes: Dominio público, vía Wikimedia Commons.

39

AES - Advanced Encryption Standard

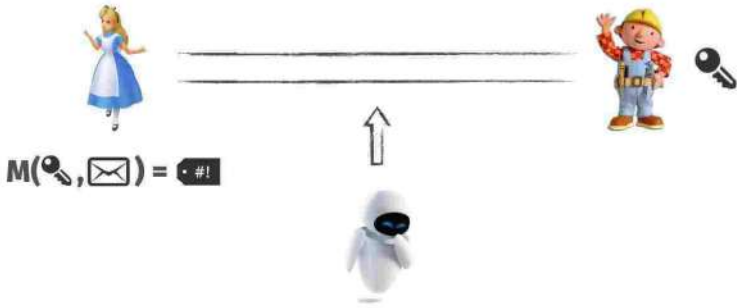




Imágenes: Dominio público, vía Wikimedia Commons.

40

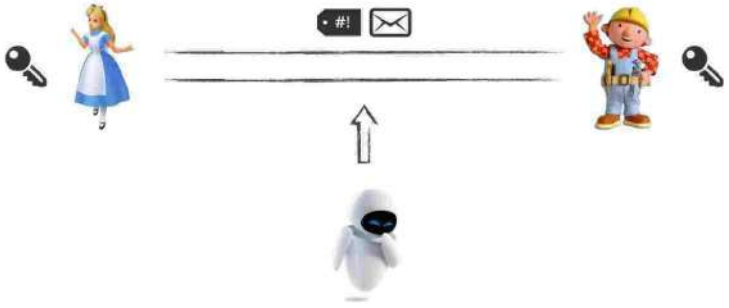
Autenticación



M( ) : generación de código

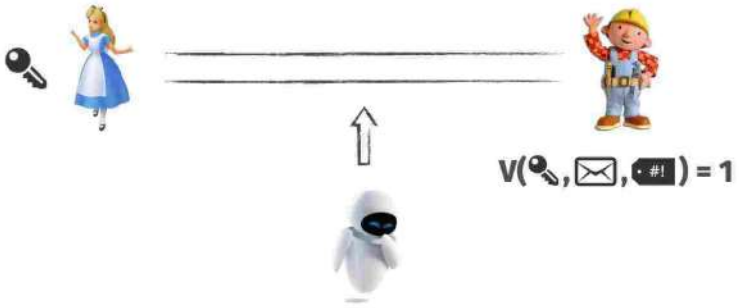
41

Autenticación



42

Autenticación



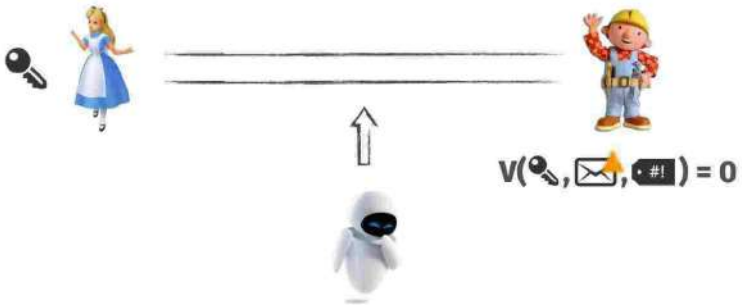
V( ) : algoritmo de verificación

43

Autenticación

44

Autenticación



45

Autenticación

- Permite verificar integridad de la información
- No ofrece privacidad
- Cifrado autenticado: ofrece privacidad + integridad

46

Código de autenticación de mensaje

- G( ): Generación de llave (probabilista)
  - Salida: llave k
- M( ): Algoritmo de generación de código (probabilista)
  - Entrada: llave k y mensaje p
  - Salida: código de autenticación t
- V( ): Algoritmo de verificación (determinista)
  - Entrada: llave k, mensaje p y código t
  - Salida: 1 si la verificación es correcta, 0 en otro caso

Para todas las llaves k y mensajes p:

$$V(k, p, M(k, p)) = 1$$

47

Código de autenticación de mensaje

- Pueden ser construidos utilizando funciones pseudo-aleatorias
  - Por ejemplo, cifradores por bloque (en modo CBC)
- Otra opción es utilizar funciones hash

48

Cifrado autenticado

- Ofrece privacidad e integridad simultáneamente
- Puede construirse combinando un esquema de cifrado y código de autenticación
  - No cualquier combinación resulta en un esquema seguro

49

Cifrado autenticado

Cifrar y autenticar

c, t

$c = \mathsf{E}(k, p)$; $t = \mathsf{M}(k, p)$

$p = \mathsf{D}(k, c)$

$\mathsf{V}(k, t, p)$

- Inseguro (se puede filtrar información del mensaje)

50

Cifrado autenticado

Autenticar después cifrar

c, t

$t = \mathsf{M}(k, p)$

$c = \mathsf{E}(k, p \| t)$

$p \| t = \mathsf{D}(k, c)$

$\mathsf{V}(k, t, p)$

- Inseguro (el adversario puede utilizar errores en padding para descifrar el mensaje)

51

Cifrado autenticado

Cifrar después autenticar

c, t

$c = \mathsf{E}(k, p)$

$t = \mathsf{M}(k, c)$

Si $\mathsf{V}(k, t, c) = 1$

$p = \mathsf{D}(k, c)$

- Seguro si el esquema de autenticación es suficientemente seguro

52

Función hash

H() = 1e8390d

H() = a3599c7

H() = ddea2c3

H() = 000d3e2

53

Función hash

- Utilizadas en estructuras de datos (tabla hash)
- Función que recibe entradas de longitud arbitraria y produce salidas de longitud fija y corta
- Múltiples entradas generan la misma salida; una *colisión*
- Una función hash buena genera pocas colisiones
- En criptografía nos interesan las funciones hash *resistentes a colisiones*

54

Función hash resistente a colisiones

Una función hash H es *resistente a colisiones* si encontrar un par de entradas diferentes (x, x') tal que $H(x) = H(x')$ es computacionalmente difícil

55

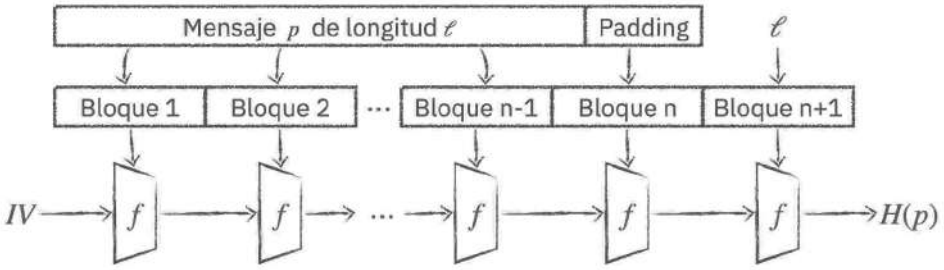
Función hash resistente a colisiones

- *Resistencia a segunda preimagen*:
  - Dado x, es difícil encontrar $x', x' \neq x$, tal que $H(x) = H(x')$
- *Resistencia a preimagen*:
  - Dado un valor y, es difícil encontrar x tal que $H(x) = y$
- Resistencia a colisiones incluye resistencia a segunda preimagen y preimagen

56

Construcción de funciones hash

Construcción Merkle-Damgård

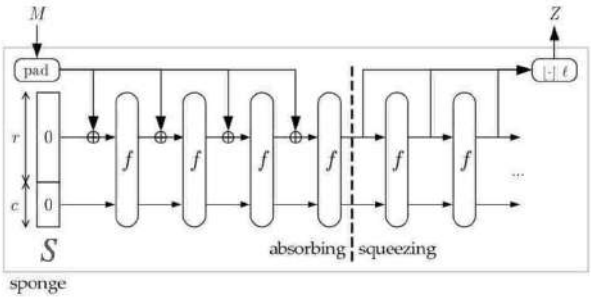


- f es una *función de compresión:* función hash con entradas de longitud fija

57

Construcción de funciones hash

Funciones esponja



- S es la memoria de tamaño $r + c$
- f es una permutación

Imagen: https://keccak.team/images/Sponge-150.png

58

Aplicaciones

- Huella digital
- Protección de contraseñas
- Arboles Merkle (red peer-to-peer, sistema de archivos)

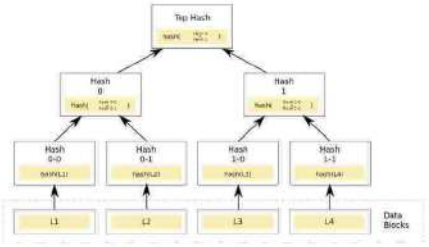


Imagen: Dominio público, via Wikimedia Commons.

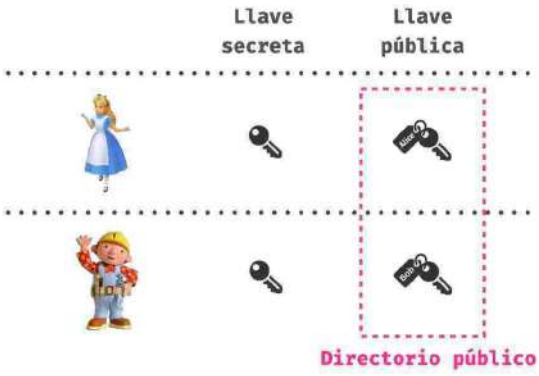
59

Ejemplos

- MD5 (insegura), SHA-1 (insegura)
- SHA-2 (Merkle-Damgård)
- SHA-3 / Keccak (esponja)
- BLAKE, CubeHash, Whirlpool, Argon2,...
- HMAC: autenticación de mensaje que utiliza una función hash criptográfica

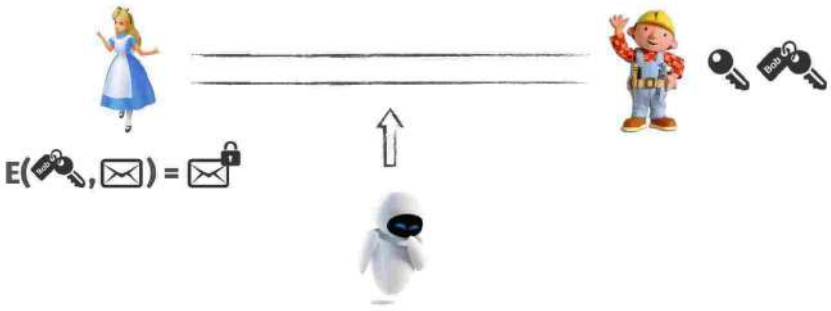
60

Criptografía asimétrica



61

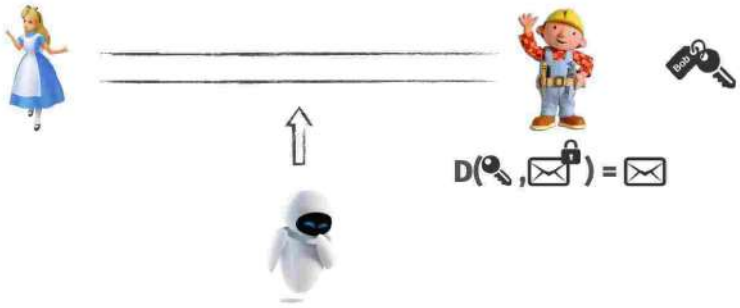
Cifrado asimétrico



E( ) : algoritmo de cifrado

62

Cifrado asimétrico



D( ) : algoritmo de descifrado

63

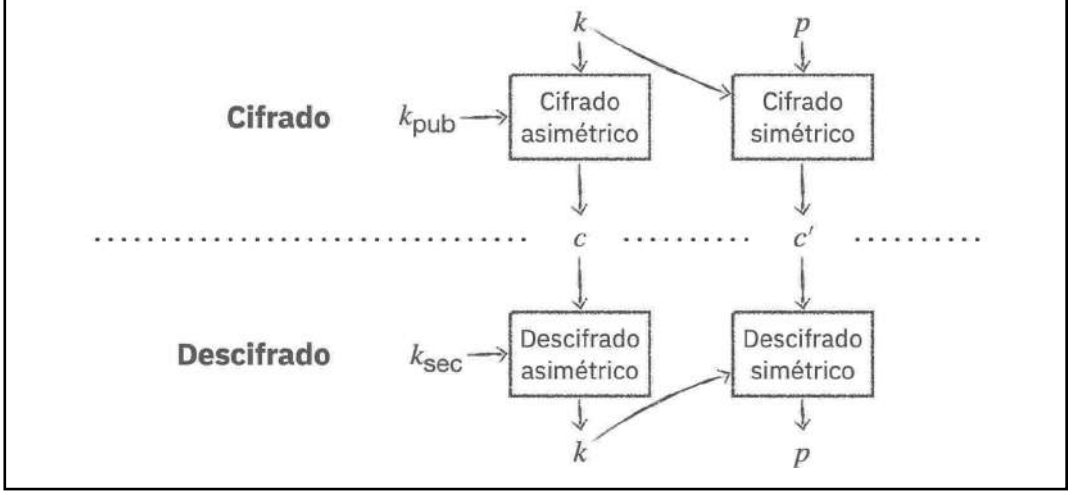
Cifrado simétrico vs cifrado asimétrico

Comparado con esquemas asimétricos ...

- cifrado y descifrado simétrico es mas rápido, y
- la longitud del texto cifrado simétrico es mas corto

64

Cifrado híbrido



65