Criptología asimétrica

Criptografía asimétrica

- Términos equivalentes: Cifrado
  - asimétrico
  - de clave pública
  - de dos claves
- Basada en teoría de complejidad y teoría de números

Problemas polinómicos
Problemas NP
Problemas NP-completos

Aritmética modular
Números primos
Factorización
Logaritmos discretos

- Requisitos:
  - Cada entidad tiene dos claves fáciles de generar computacionalmente:
    - Pública: conocida por cualquiera
    - Privada: secreta

    Una usada para cifrado y otra para descifrado
  - Computacionalmente sencillo cifrar y descifrar texto cuando se conoce clave correspondiente
  - Computacionalmente imposible determinar clave privada si sólo se conoce algoritmo y clave pública
  - Computacionalmente imposible determinar texto claro si sólo se conoce algoritmo y clave de cifrado es la pública
- Algoritmos: RSA, ElGamal, curvas elípticas

RSA

- De Rivest, Samir y Adleman en 1977 (¿previo?)
- Patentado en 1982 en EEUU
  - Prohibida exportación y uso salvo fines no comerciales

Generación de claves

1. Seleccionar dos nº primos grandes p, q ----------→ $p=17,\ q=11$

   $n=p.q$ ----------→ $n = pq = 17\times11=187$

   $\varnothing(n)=(p-1)(q-1)$ ----------→ $\varnothing(n)=(p-1)(q-1)=16\times10=160$

2. Elegir un número entero e tal que

   $1<e<\varnothing(n),\ mcd(e,\varnothing(n))=1$ ----→ $mcd(e,160)=1$, $e=7$

   (e y $\varnothing(n)$ primos entre sí)

3. Calcular/elegir un número d tal que

   $0\le d\le n$ y $e.d \equiv 1 \bmod \varnothing(n)$ -----→ $d.e=1 \bmod 160$ y $d<160$

   $d=23$ $(23\times7=161=1\times160+1)$

clave pública: (e,n) ----------→ $(7,187)$

clave privada: (d,n) o (d,p,q) ----------→ $(23,187)$

RSA (cont.)

Procedimiento

- Cifrado:

  Texto claro: $1<M<n$

  Texto cifrado: $C = M^e \pmod n$

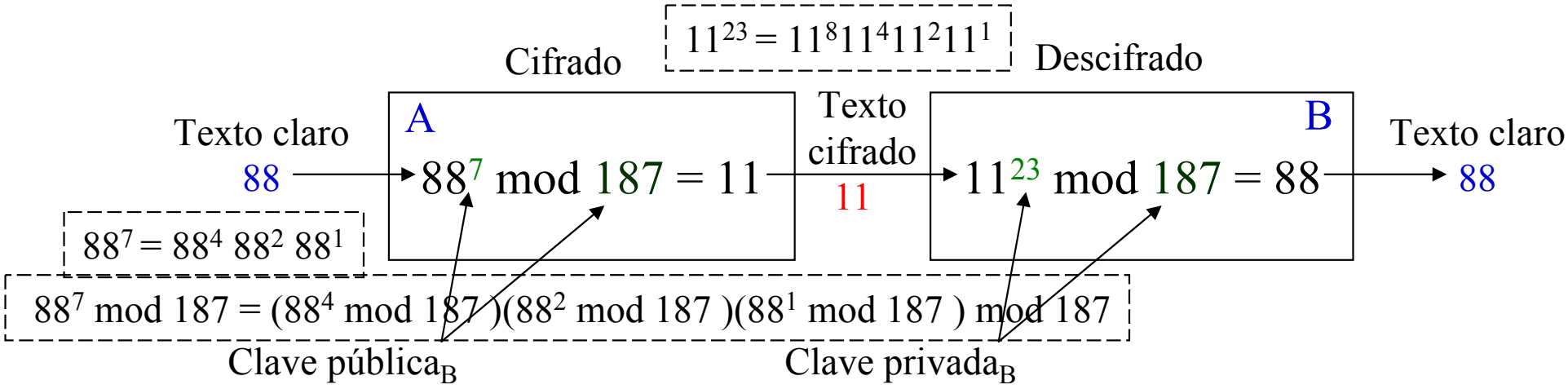
clave pública de receptor

- Descifrado:

  Texto cifrado: C

  Texto claro: $M = C^d \pmod n$

clave privada de receptor



RSA. Criptoanálisis

- Ataques matemáticos:
  - Factorizar $n (=p.q)$ para obtener $\varnothing(n)$ y después d
  - Determinar $\varnothing(n)$ directamente y luego d
  - Encontrar d directamente

  Equivalentes a factorización: por ahora seguro para nº de 1024 bits
- Ataques basados en el tiempo
  - Inferir tamaño de operando según tiempo que tardan operaciones
  - RSA aprovecha tiempo que tarda exponenciación
  - Contramedidas:
    - tiempo de exponenciación constante, retardos aleatorios, …
- Fuerza bruta:
  - No factible si tamaño de números es grande: 1024 bits (300 decimales)

  (1994: con 1600 computadoras y clave de 129 decimales cifrado roto en 8 meses

  1999: 130 dígitos decimales (512 bits) roto con algoritmo GNFS)

ElGamal

- Taher ElGamal en 1985
- No patentado

Generación de claves

- Elegir un primo p grande (puede ser conocido y compartido)
- Elegir raíz primitiva g, $1<g<p$ (puede ser conocido y compartido)
- Elegir x, $x<p$
- Calcular $y=g^x \bmod p$
- Pública: p, g, y privada: x

Procedimiento

- Cifrado:
  - Elegir número aleatorio k, k y $p-1$ primos entre sí
  - $C(M,k) = (a,b)$, donde
    $a = g^k \bmod p$
    $b = My^k \bmod p$
- Descifrado:
  - $D(a,b) = b/a^x \bmod p = M$

Tamaño de cifrado doble que claro

clave pública de receptor

clave pública de receptor

Comparación entre simétricos y asimétricos

Criptosistemas simétricos

- Ventajas:
  - Muchos algoritmos:
    - IDEA, CAST, 3DES, RC4, Blowfish, RC5 y 6, AES, etc.
  - Implementación eficiente en software y hardware
  - Rápidos o muy rápidos
- Inconvenientes:
  - Necesario secreto previo compartido
    - Comunicación segura de claves (procedimiento y canal)
  - Número de claves crece con el cuadrado del número de comunicantes:
    - n comunicantes → n(n-1)/2 claves

Criptosistemas asimétricos

- Ventajas:
  - Implementación eficiente en software y, algo menos, en hardware
  - No necesario secreto previo compartido
  - Número de claves crece linealmente con número de comunicantes
- Inconvenientes:
  - Pocos algoritmos
    - RSA, ElGamal, Curvas Elípticas
  - Muy lentos y costosos computacionalmente

Funciones hash

Funciones hash

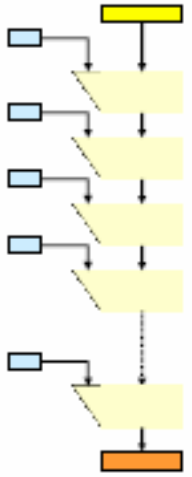
Valor representa original de manera concisa

REQUISITOS de una función hash H para usarla en criptosistemas:

- Se puede aplicar a entradas de cualquier tamaño
- Produce una salida de longitud fija
- H(x) es fácil de calcular
- Es de sentido único:
  - Dado H(x), es computacionalmente imposible encontrar x
- Es resistente a colisiones:
  - Dado x, es computacionalmente imposible encontrar x'distinta, tal que H(x)=H(x') (débil)
  - Es computacionalmente imposible encontrar x y x' distintos, tales que H(x)=H(x') (fuerte)

PROCEDIMIENTO

- Basado en funciones de compresión con entrada de longitud fija
  - Texto procesado en bloques (puede necesitarse relleno)
  - Salida a entrada de siguiente (puede necesitarse vector de inicialización)
- Algoritmos: familia MD, familia SHA, RIPEMD-160, ...
- Criptoanálisis: Seguridad depende de probabilidad de colisión



Imágenes de [Dávila, 2005]

MD5

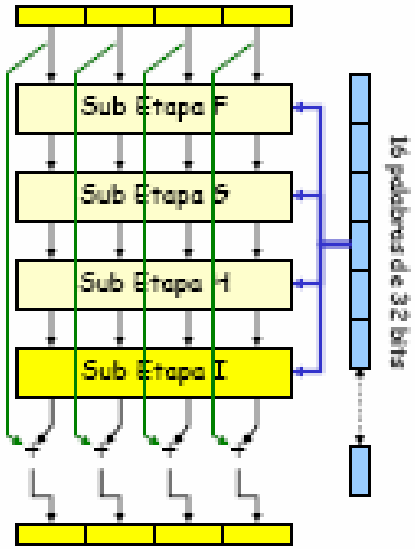
- Ronald Rivest diseñó MD4 en 1990
  - Procesa por etapas con bloques de 512 bits (16 x 32 bits)
  - Salida son 4 palabras de 32 bits (128 bits)

  Criptoanálisis parciales: sentimiento general de poca seguridad
- Ronald Rivest mejoró MD4 con MD5 en 1991 (RFC1321)
  - Entrada y salida como MD4

  FUNCIONAMIENTO
  - Rellenar texto para que longitud sea 448 mod 512
  - Añadir 64 bits (con longitud en bits de original)
  - Inicializar 4 variables (128 bits) en registro (`123456`, `89abcdef`, `fedcba98`, `76543210`)
  - Procesar en grupos de 16 palabras (512 bits):
    - 4 etapas intermedias de 16 operaciones de bit con bloques y registro
  - Valor de salida = Concatenación de valor final de variables

  Parte de muchas aplicaciones: IPsec, PGP, …

  Criptoanálisis: en Crypto2004 anuncio de ruptura de familia MD (y otros)



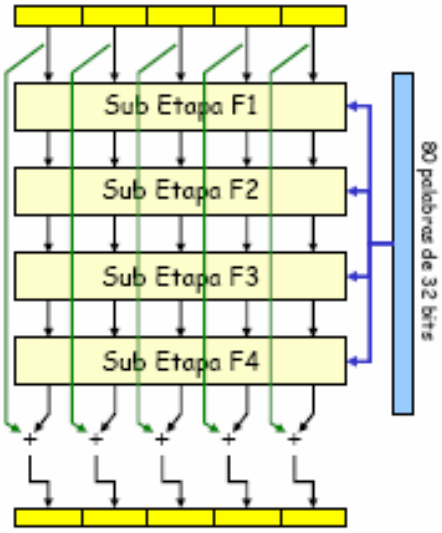
Imágenes de [Dávila, 2005]

Secure Hash Algoritm (SHA)

- Diseñado por NIST y NSA en 1993
  - Basado en MD4 y MD5
  - Revisado en 1995: SHA-1
  - Revisión FIPS 180-2: SHA-256, SHA-384, SHA-512

FUNCIONAMIENTO

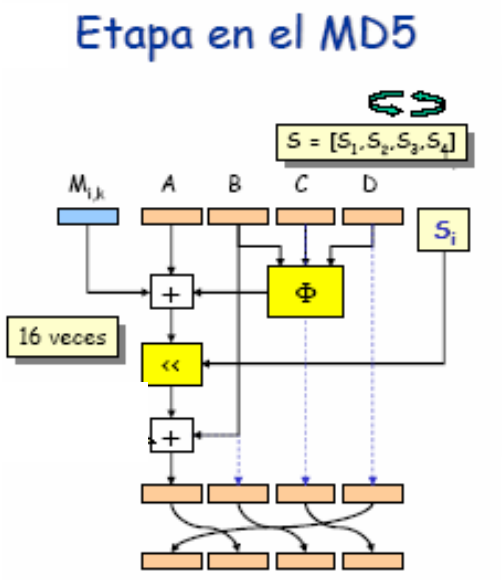
- Entrada de longitud < 2^{64} bits, salida de 160 bits
- Rellenar texto para que longitud sea 448 mod 512
- Añadir 64 bits (con longitud en bits de original)
- Inicializar 5 variables (160 bits) en registro
  - (`67452301,efcdab89,98badcfe,10325476,c3d2e1f0`)
- Procesar en grupos de 16 palabras (512 bits):
  - Expandir a 80 palabras mezclando y desplazando
  - 4 etapas de 20 operaciones de bit con bloques y registro
- Valor de salida = Concatenación de valor final de variables

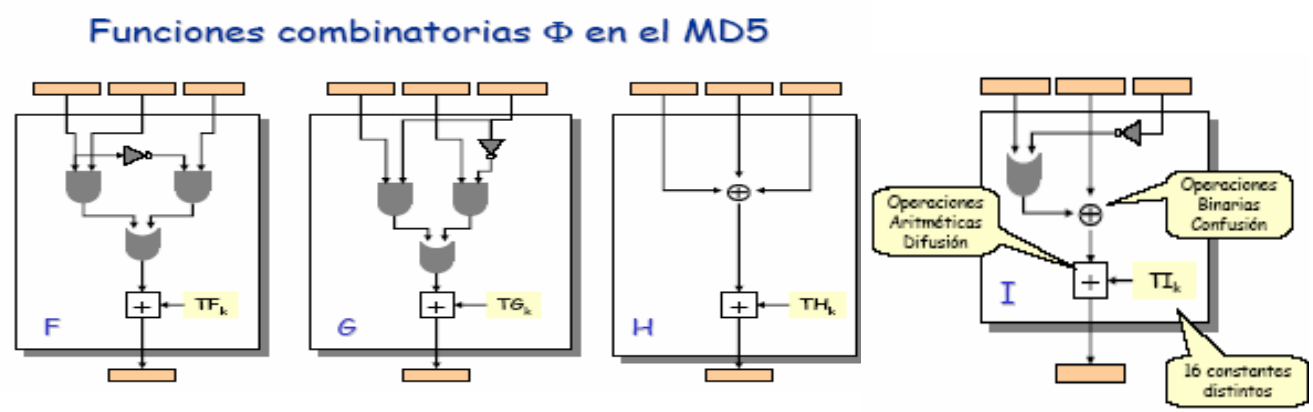


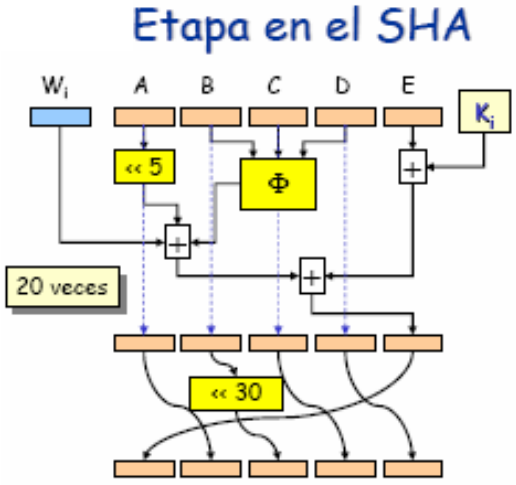
Criptoanálisis: en Crypto2004 anuncio de ruptura de SHA-0 y debilitamiento de SHA-1

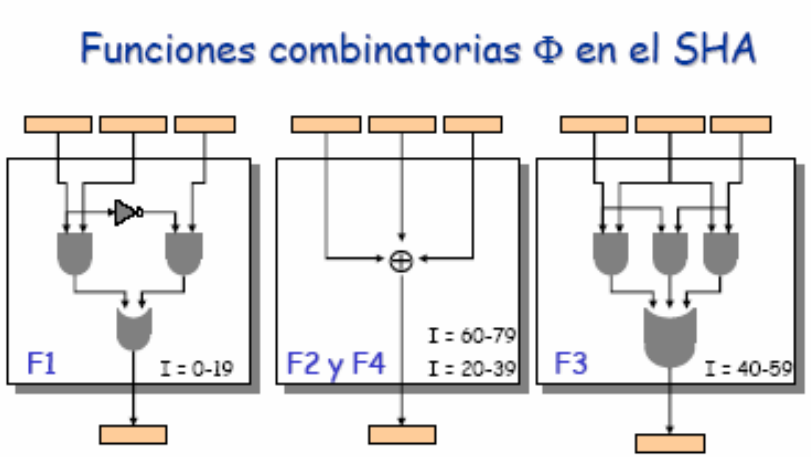
Imágenes de [Dávila, 2005]

Etapas intermedias









Imágenes de [Dávila, 2005]

Aplicaciones

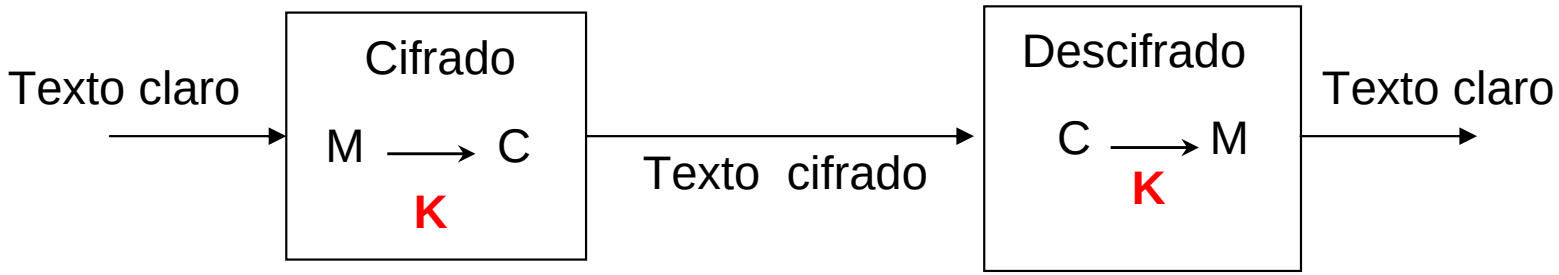
Confidencialidad

Integridad

Autenticación

Confidencialidad

- Critografía simétrica
  - Cifrado y descifrado con misma clave



  - En comunicación:
    - Cifrado de
      - mensaje completo: necesario descifrar en cada nodo de la red (clave secreta cada 2)
      - mensaje sin cabecera: sólo descifrar en destino

      } combinación

    - ¿Cómo conseguir que emisor y receptor compartan clave?

↓

Necesaria distribución segura de clave

Distribución de clave

Posibilidades

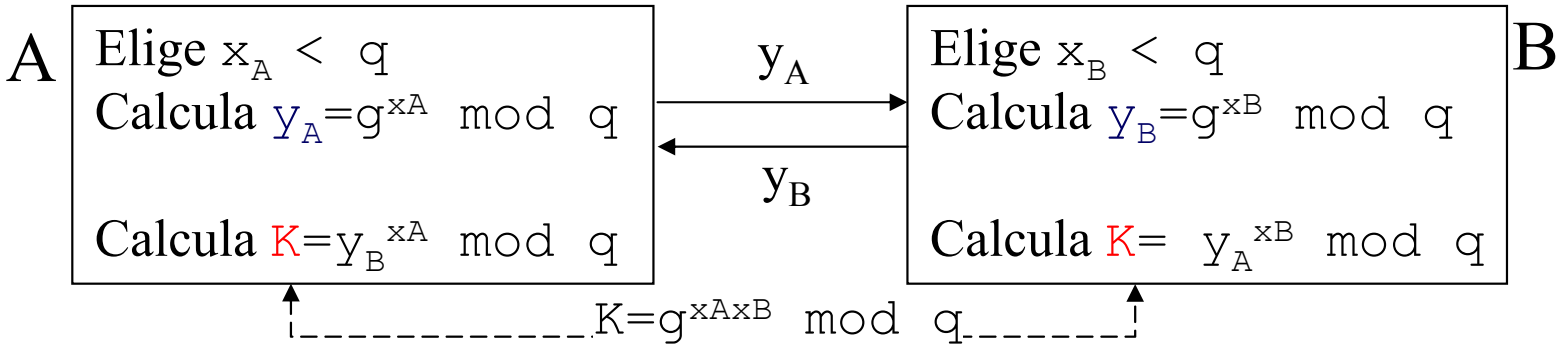
- Clave pasada físicamente
  - elegida por uno de los dos
  - elegida por tercero
- Clave cifrada (necesaria otra clave antes)
  - elegida por uno de los dos
  - elegida por tercero

Mixtos

- Clave permanente (A)
  - Para distribuir claves de sesión
- Clave de sesión (S)
  - No vale al final de la sesión o tras cierto tiempo

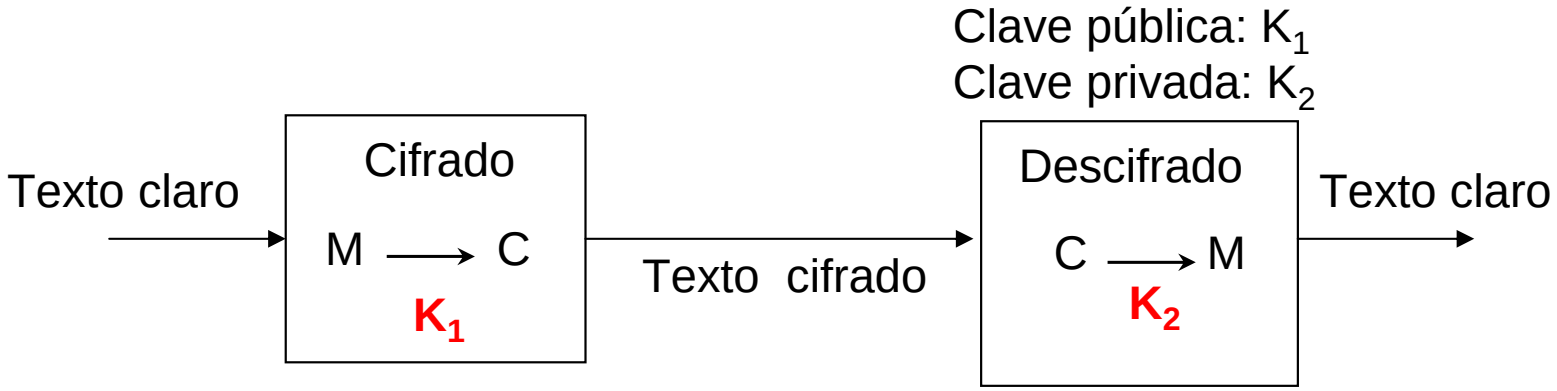
Algoritmo de Diffie-Hellman para intercambio de clave (1976)

- No se pasa secreto entre usuarios →escuchas no pueden interceptarlo
- Parámetros iniciales: q primo y g raíz primitiva de q (pueden conocerse)



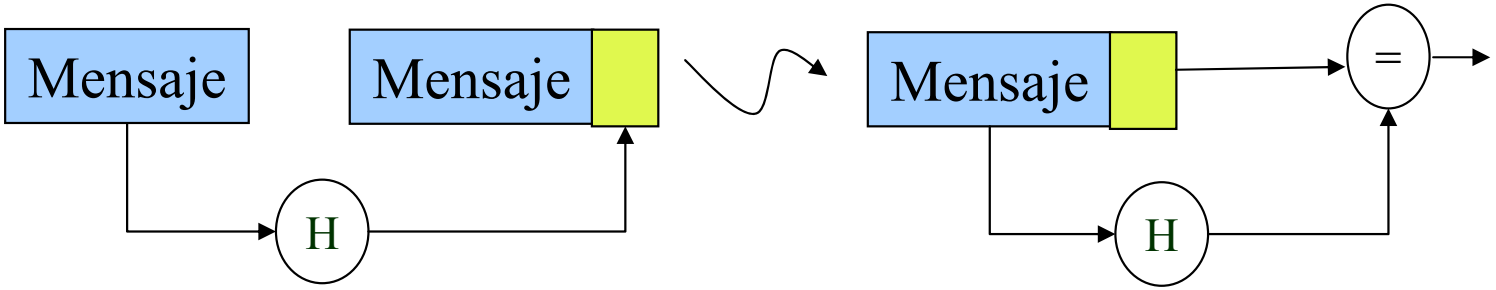
Confidencialidad (cont.)

- Criptografía asimétrica
  - Cifrado y descifrado con distinta clave
  - En comunicación:
    - Emisor cifra con clave pública de receptor
    - Receptor descifra con su clave privada



Integridad

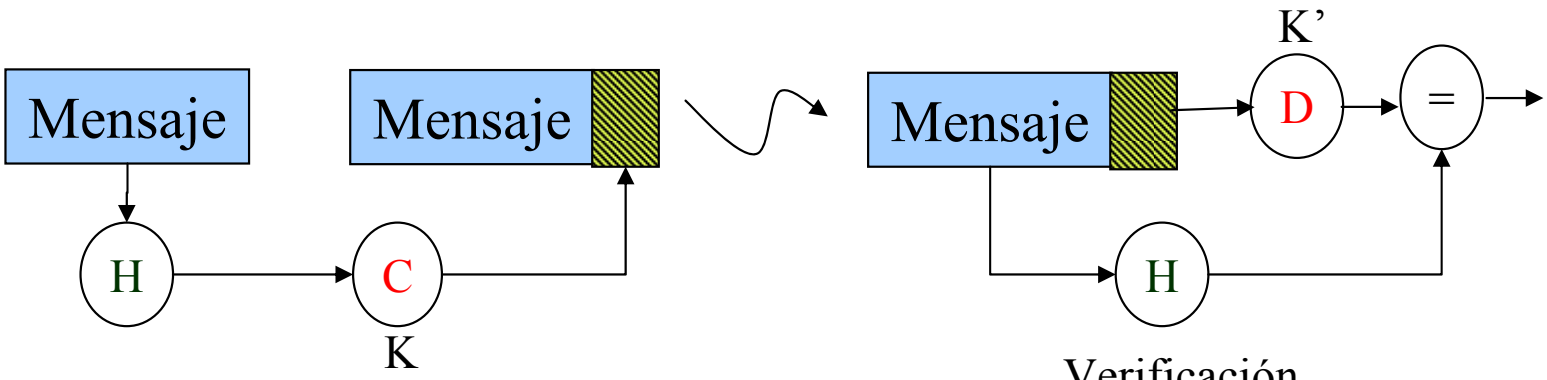
- Resumen hash



¿Seguridad frente a cambiazo?

Verificación

- Resumen hash (cifrado)

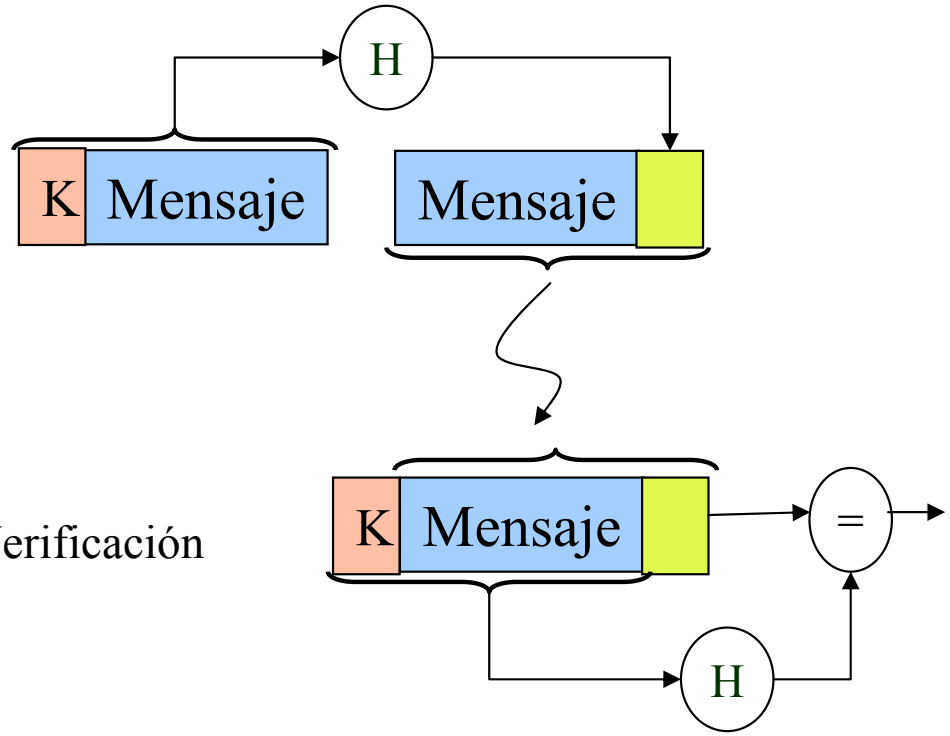


¿K=K'?

Verificación

Integridad (cont.)

- MAC (*Mesage Authentication Code*):
  - Resumen que depende de clave
    - Ej., último bloque de salida de DES en modo CBC
  - HMAC: hash (de sentido único) como MAC



- Concatena clave y mensaje
- Posibilidades:

  $HMAC_K(X)=h(K \mid X)$

  $HMAC_K(X)=h(X \mid K)$

  $HMAC_K(X)=h(K|X|K)$

  $HMAC_K(X)=h(K1|X|K2)$

  $HMAC_K(X)=h(K1|h(K2|X))$

  (K1, K2 a partir de K)
- Ventajas:
  - Más rápido que cifrador
  - Sin patente

- Seguridad: la de algoritmo hash utilizado

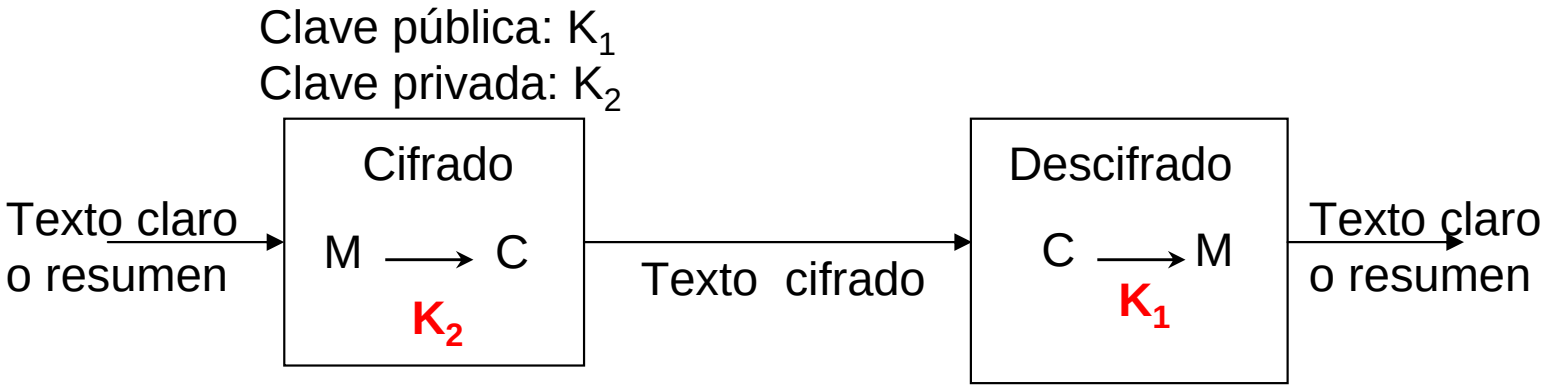
Autenticación

Criptografía simétrica:

– ¿Cuál de los dos lo ha hecho?

Criptografía asimétrica:

- Cifrar con clave privada y descifrar con clave pública de emisor



- Algoritmos de firma digital : El Gamal, DSA, ...
- Generalmente cifrado de resumen, no de mensaje completo
  – Más rápido

Autenticación (cont.)

DSA (*Digital Signature Algorithm*)

– Estándar NIST en 1991, basada en ElGamal y hash de sentido único

– No patentado

¿Vale para confidencialidad?

- Parámetros:
  - p primo,
    q primo (160 bits) y
    q divide $p-1$
  - g raíz primitiva, $y=g^x \bmod p$
  - x nº aleatorio entre 1 y q
  - Clave pública y, p, q, g
    clave privada x
  - M texto claro
  - H función hash
  - k clave aleatoria para una vez

- Firma:

  $r=(g^k \bmod p) \bmod q$

  $s=k^{-1}(H(M)+xr) \bmod q$

  $firma_x(M)=(r,s)$

- Verificación:

  Dados M y (r,s)

  $u1 = s^{-1} H(M) \bmod q$

  $u2 = s^{-1} r \bmod q$

  $v = (g^{u1} y^{u2} \bmod p) \bmod q$

  aceptar si $v=r$