Introducción a la Criptografía

Elvira Mayordomo Cámara

Octubre 2012

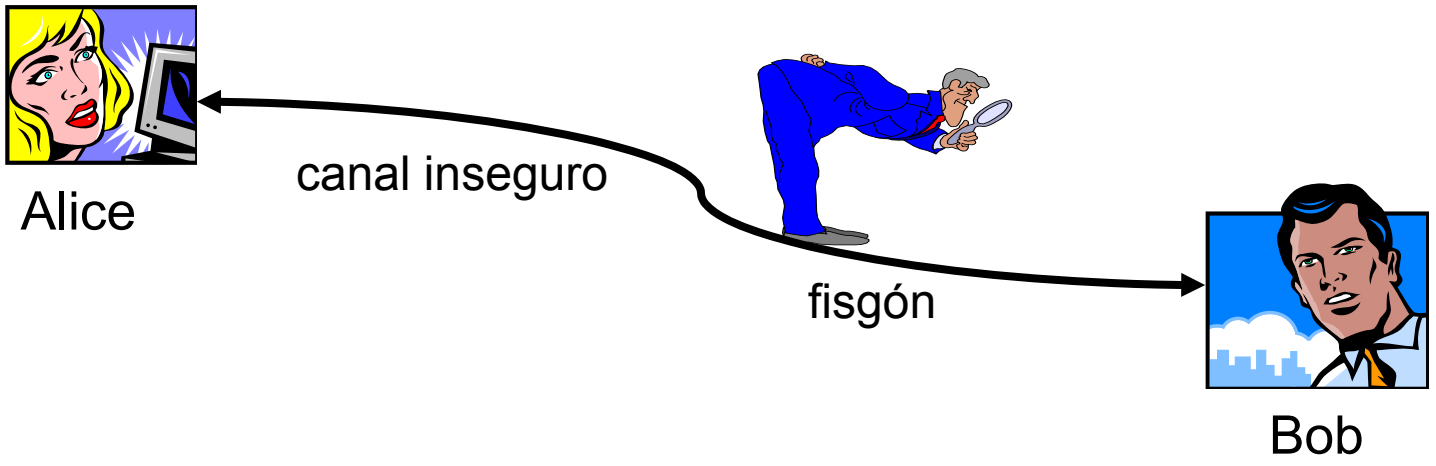
Bibliografía

- Bruce Schneier: “Applied Cryptography”. Wiley, 1996
- N. Ferguson, B. Schneier, T. Kohno: “Cryptography Engineering”. Wiley, 2010
- Carlo Blundo: Apuntes de “Introduction to Cryptography”. 1st International School On Foundations Of Security Analysis And Design, Bertinoro, 2000
- Estándares NIST: http://csrc.nist.gov/

  http://csrc.nist.gov/publications/PubsFIPS.html

¿Qué es Criptografía?

- Una forma de mantener la comunicación privada



- Esto era cierto en el pasado
- Criptografía es mucho más que “escritura secreta”

La Criptografía es sobre ...

Comunicación en presencia de enemigos activos

- Alice y Bob intercambian mensajes
- Eve controla el canal de comunicación
  - Escucha
  - Bloquea la comunicación
  - Manda su propio mensaje
  - Responde a un mensaje interceptado
  - Modifica el mensaje
  - ...

El objetivo que se busca es ...

- Supongamos que Alice le manda a Bob un mensaje M
  - Privacidad/Confidencialidad
    - M se mantiene secreto
  - Integridad de datos
    - M no se altera
  - Autentificación, identificación
    - Bob sabe quién mandó M
  - No repudiación
    - Alice no puede negar haber mandado el mensaje M

- Alice y Bob hacen todo esto usando Criptografía

PRIMERA PARTE

Veremos 5 primitivas criptográficas:

- Criptografía de clave privada
- Criptografía de clave pública
- Firma digital
- Funciones hash
- Intercambio de claves

Se quedan en el tintero ...

- Autentificación de mensajes
  - MAC (código de autentificación de mensajes)
- Autentificación de tiempo
  - “Timestamping”

Intro: Criptología

- Criptografía
  - Hacer:
    - Cifradores (también llamados codificadores y “encriptadores”)
    - Algoritmos de firma
    - Funciones hash
    - ...
- Criptoanálisis
  - Romper:
    - Todos los anteriores

Intro: Definición básica

Sistema criptográfico = <M, C, K, E, D>

- M es el conjunto de mensajes posibles m
- C es el conjunto de posibles mensajes cifrados c (también llamados mensajes “encriptados”)
- K es el conjunto de claves posibles k
- E es el conjunto de funciones de cifrado e

  e(m,k) es un mensaje cifrado

  e: M x K → C
- D es el conjunto de funciones de descifrado d

  d(c,k) es un mensaje

  d: C x K → M

Intro: Sistema Criptográfico

Un sistema criptográfico es una forma de transformar un mensaje m en un mensaje cifrado c bajo el control de una clave secreta k

- Cifrar un mensaje
  - c= e(m,k)
- Descifrar un mensaje
  - m= d(c,k)

- m= d(e(m,k),k)

Intro: Principio de Kerckhoffs

- La seguridad de un sistema criptográfico debe depender SÓLO de que la clave sea secreta y NO de que el algoritmo de cifrado sea secreto

- A. Kerckhoffs: “La Criptographie Militaire” (1883)

Intro: Medida de la seguridad

- Seguridad incondicional
  - Los enemigos tienen poder de cálculo ilimitado
  - Para romper un sistema criptográfico hay que probar todas las claves posibles
- Seguridad computacional
  - Los enemigos tienen poder de cálculo limitado
    - En general, pueden resolver sólo problemas resolubles en tiempo polinómico

Intro: Seguridad incondicional

- El conocimiento del mensaje cifrado no da ninguna información sobre el mensaje original

  Pr(mensaje=m | mensaje cifrado=c) = Pr(mensaje=m)

- Asumiendo que:
  - La clave tiene 120 bits y tenemos un procesador corriendo a 3GHz
  - Probamos una clave en cada ciclo
  - Para probar todas las claves necesitamos

    ≈ 140.000.000 siglos

Intro: Seguridad computacional

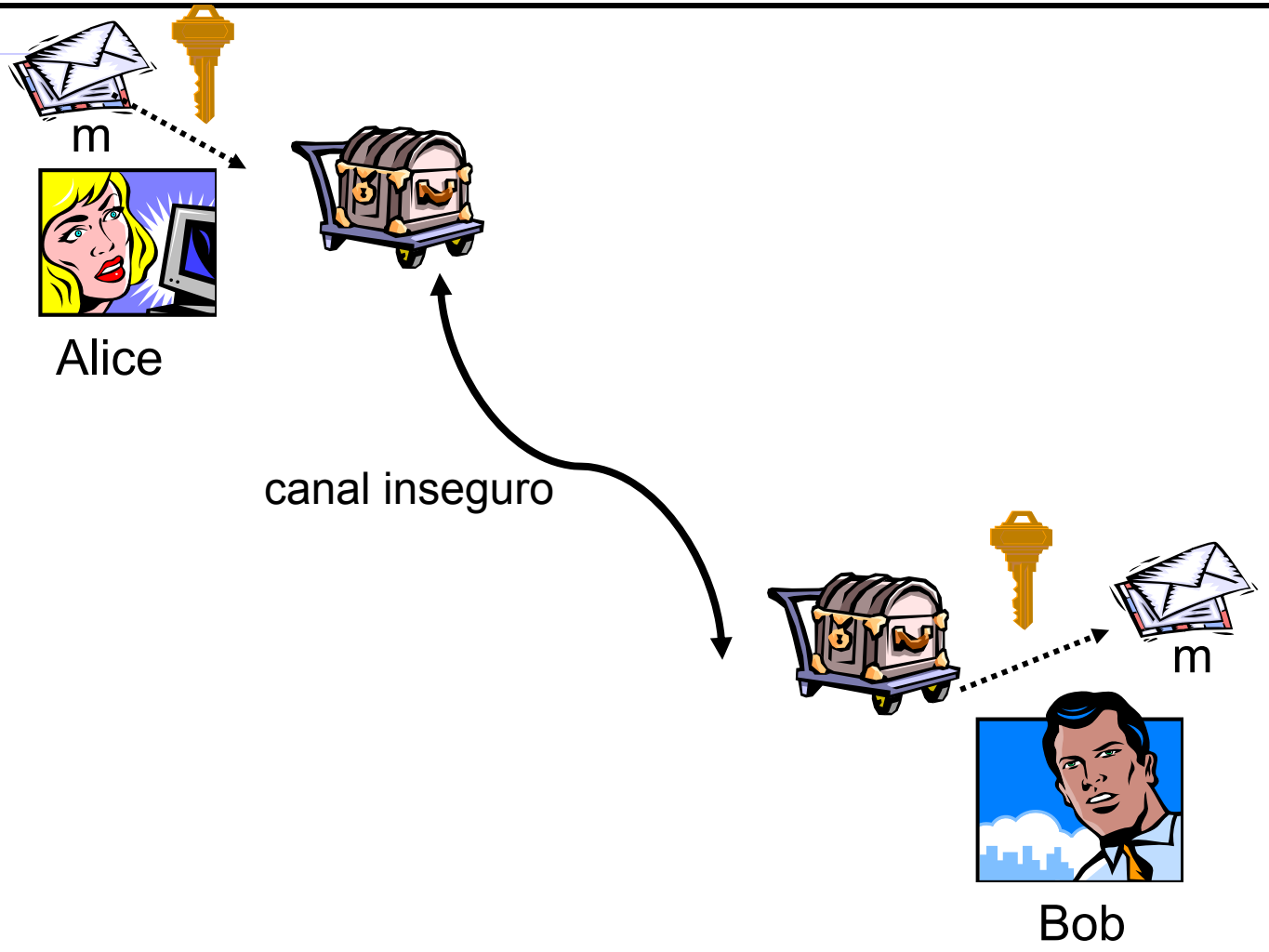
- La dificultad de vencer a un sistema criptográfico se puede ver que es tanta como resolver un problema que se supone muy difícil
  - Factorización de un entero, cálculo del logaritmo discreto

- Problemas fáciles
  - Encontrar el máximo de n números: tiempo O(n)
  - Ordenar n elementos: tiempo O(n log n)
- Un problema difícil:
  - Factorizar N (de n bits): tiempo $2^{O(\text{raíz}(n \log n))}$

C. privada: Sistemas criptográficos de clave simétrica

- Llamados de clave privada, de clave simétrica, de clave secreta
- Alice y Bob conocen la misma clave k
- Hay dos tipos:
  - Cifrador “a chorro”: mensajes cifrados carácter a carácter
  - Cifrador por bloques: mensajes primero divididos en bloques, después cifrados

C. privada: Gráficamente ...



C. privada: Protocolo

(Suponemos que Alice y Bob conocen la misma clave k)

1. Alice cifra su mensaje m usando k: c=e(m,k)
2. Alice manda c a Bob
3. Bob recibe c. Bob descifra usando k: d(c,k)=m

C. privada: Por supuesto

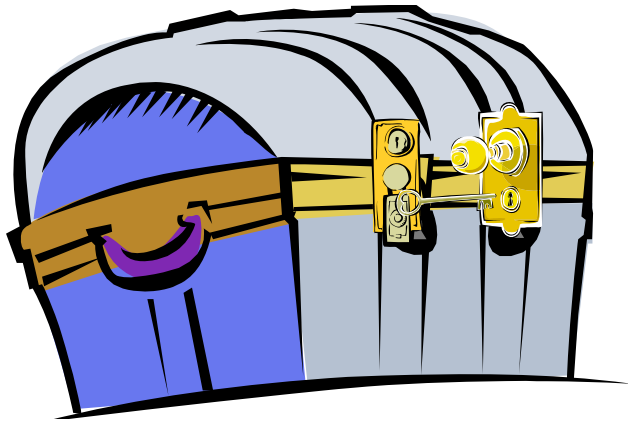
- Alice y Bob confían el uno en el otro, ya que no pueden hacer nada contra:
  - Uno de los dos vendiendo su clave a otro
  - Uno de los dos haciendo públicos los mensajes

C. privada: Problemas

- Transporte de las claves
  - Las claves tienen que ser transportadas manteniendo la confidencialidad: caro
- Número de claves
  - Para tener seguridad cada pareja tiene que tener una clave distinta
  - Para n personas se tienen que construir y distribuir n(n-1)/2 claves
- Tienes que saber con quién estás hablando

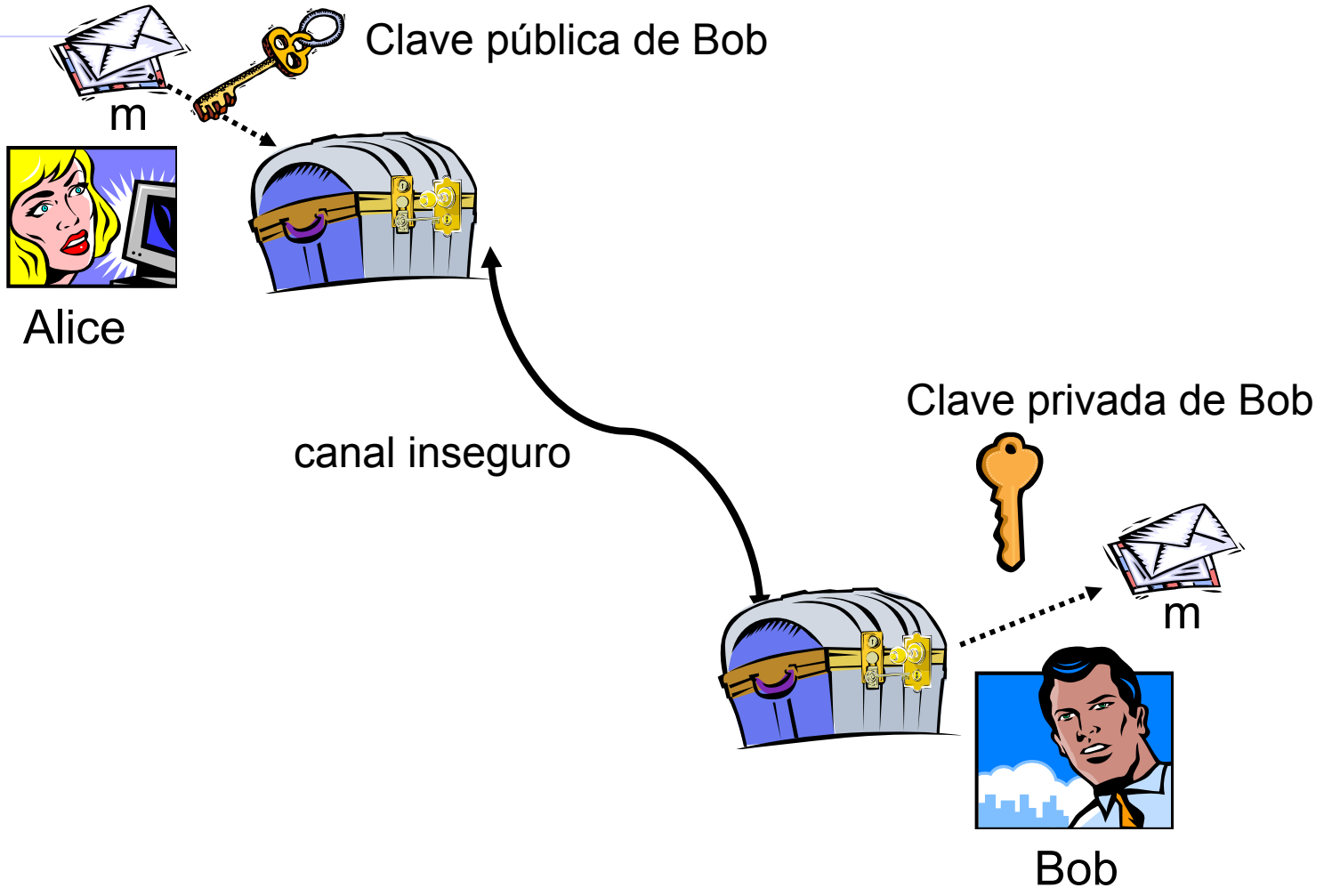
C. pública: Sistemas criptográficos de clave pública

- Llamados de clave pública, de clave asimétrica
- Usa un tipo especial de cofre con dos cerraduras
  - Con una llave (pública) cerramos el cofre
  - Con la otra (privada) abrimos el cofre



Clave privada ≠ Clave pública

C. pública: Gráficamente ...



C. pública: Protocolo

Bob tiene su clave privada $B_{privada}$ y todos tienen la clave pública de Bob, $B_{pública}$

1. Alice cifra su mensaje m usando la clave pública de Bob: $c = e(m, B_{pública})$
2. Alice manda c a Bob
3. Bob recibe c. Bob descifra usando su clave privada $B_{privada}$: $d(c, B_{privada}) = m$

C. pública: Propiedades

- Sólo el destinatario puede leer el mensaje
- Sólo hay que almacenar una clave
- Cualquiera puede cifrar con la clave pública
- No son necesarios canales seguros para comunicar la clave privada

C. pública: Funcionamiento

- Cada usuario genera su par (clave pública, clave privada) y publica la clave pública en un servidor de claves
  - Key Certification Authority o Key Distribution Center (KDC)

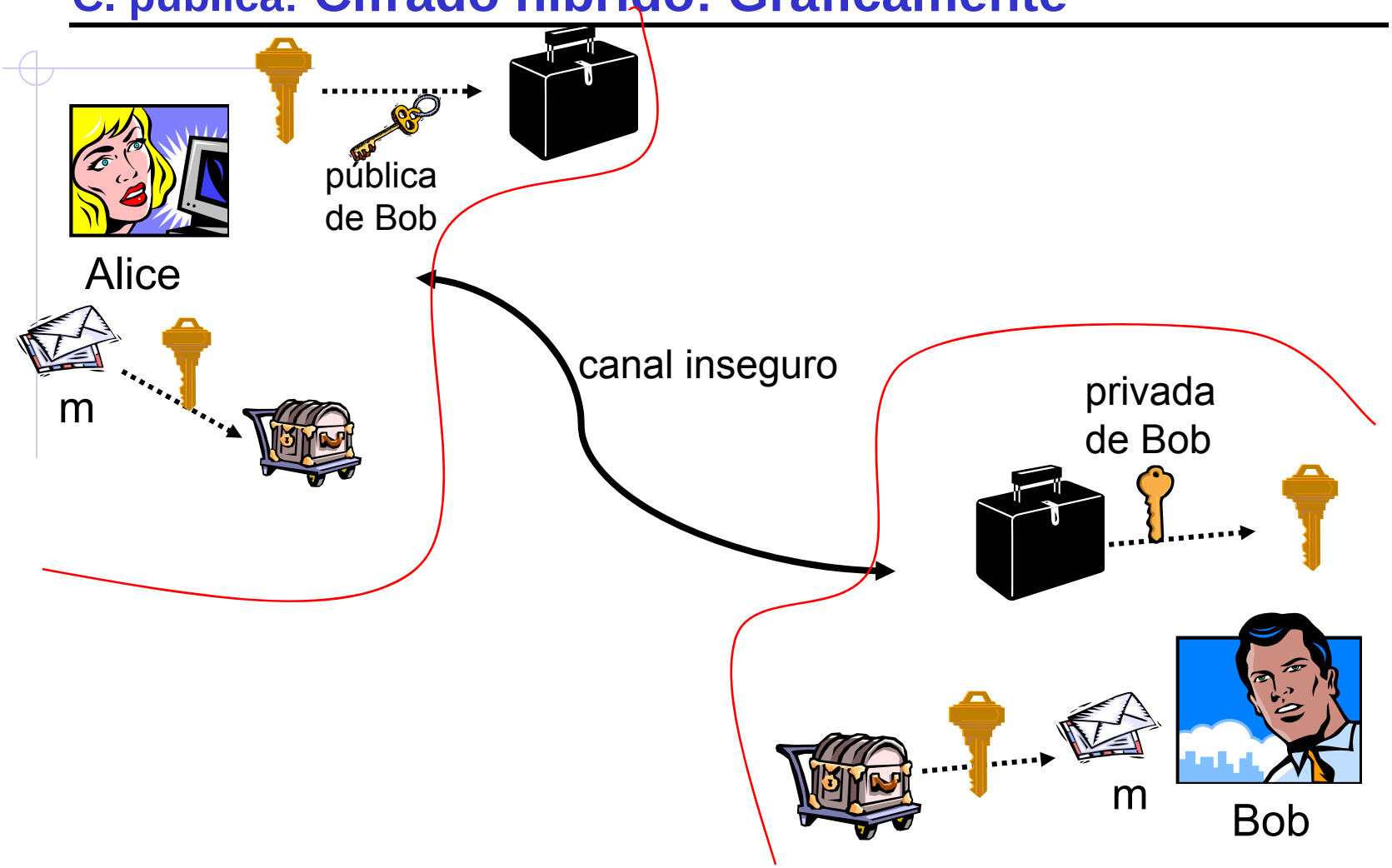
C. pública: Problemas

- La clave privada debe mantenerse privada
- Alice debe estar segura de que está usando la clave pública de Bob
- Debe ser fácil obtener las claves públicas
- Debería ser (prácticamente) imposible sacar la clave privada a partir de la clave pública
- **Cifrado y descifrado son más lentos que en los sistemas de clave secreta**

C. pública: Cifrado híbrido

- Los sistemas de clave secreta son mucho más rápidos que los de clave pública
- Muchas veces se usa una combinación:
  - El sistema de clave pública se usa para compartir una clave secreta S que sólo se usa una vez
  - El sistema de clave secreta usa S para cifrar el mensaje

C. pública: Cifrado híbrido: Gráficamente



C. pública: Protocolo de cifrado híbrido

Bob tiene su clave privada $B_{privada}$ y todos tienen la clave pública de Bob, $B_{pública}$

1. Alice genera una clave secreta S
2. Alice cifra su mensaje m usando S: cm = e1(m,S)
3. Alice cifra S usando la clave pública de Bob: cs=e2(S,$B_{pública}$)
4. Alice manda [cm,cs] a Bob
5. Bob recibe [cm,cs].
6. Bob descifra S usando su clave privada $B_{privada}$: d2(cs, $B_{privada}$)=S
7. Bob descifra m usando S: d1(cm,S)

Firma digital: ¿Qué es la firma digital?

- Alice le manda un mensaje a Bob usando un sistema de clave pública
  - Nadie puede leer el mensaje de Alice a Bob pero cualquiera podría haberlo mandado
  - ¿Cómo sabe Bob que se lo ha mandado Alice?
- Solución:
  - Alicia firma sus mensajes

Firma digital: Propiedades

- Sólo el usuario legítimo puede firmar su documento
- Nadie podrá falsificar una firma
- Cualquiera puede verificar una firma digital
- Una copia de una firma digital es igual a la original
  - Esto no es cierto para la firma escrita convencional

Firma digital: Más propiedades

- No se puede reutilizar una firma
- No se puede modificar una firma
- No se puede negar haber firmado un documento
- No se puede alterar un documento después de haberlo firmado

Firma digital: Esquema

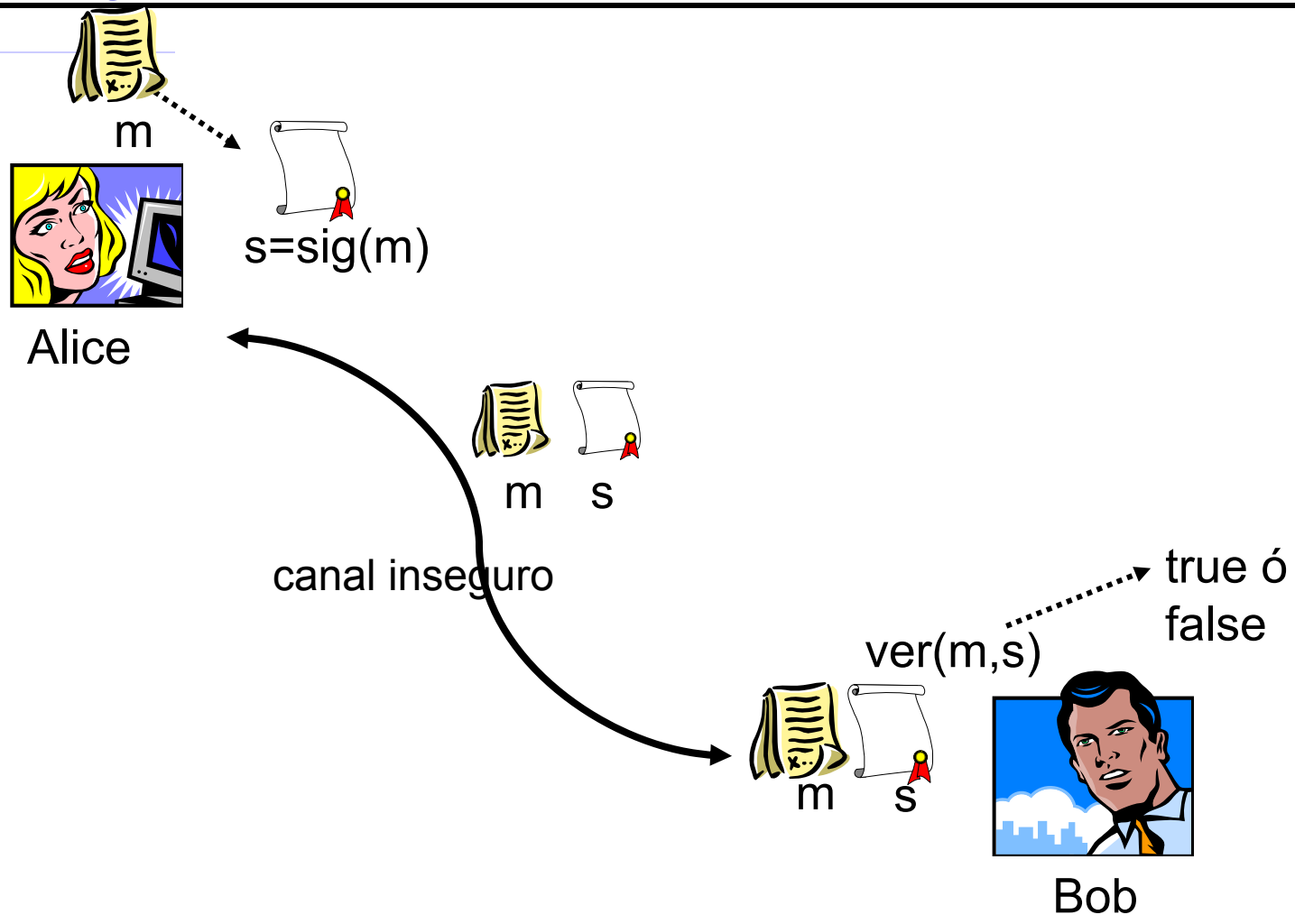
Un esquema de firma = <M, S, K, SA, V>

- M es el conjunto de mensajes posibles m
- S es el conjunto de posibles firmas
- K es el conjunto de claves posibles k
- SA es el conjunto de algoritmos de firma sig
  sig(m,k) es una firma
  sig: M x K → S
- V es el conjunto de algoritmos de verificación ver
  ver(m,s,k) es un booleano
  ver: M x S x K → {true, false}

Firma digital: Propiedades

$$ver(m,s,k)=\begin{cases} \text{true si } s=sig(m,k) \\ \text{false si } s\neq sig(m,k) \end{cases}$$

Firma digital: Gráficamente ...



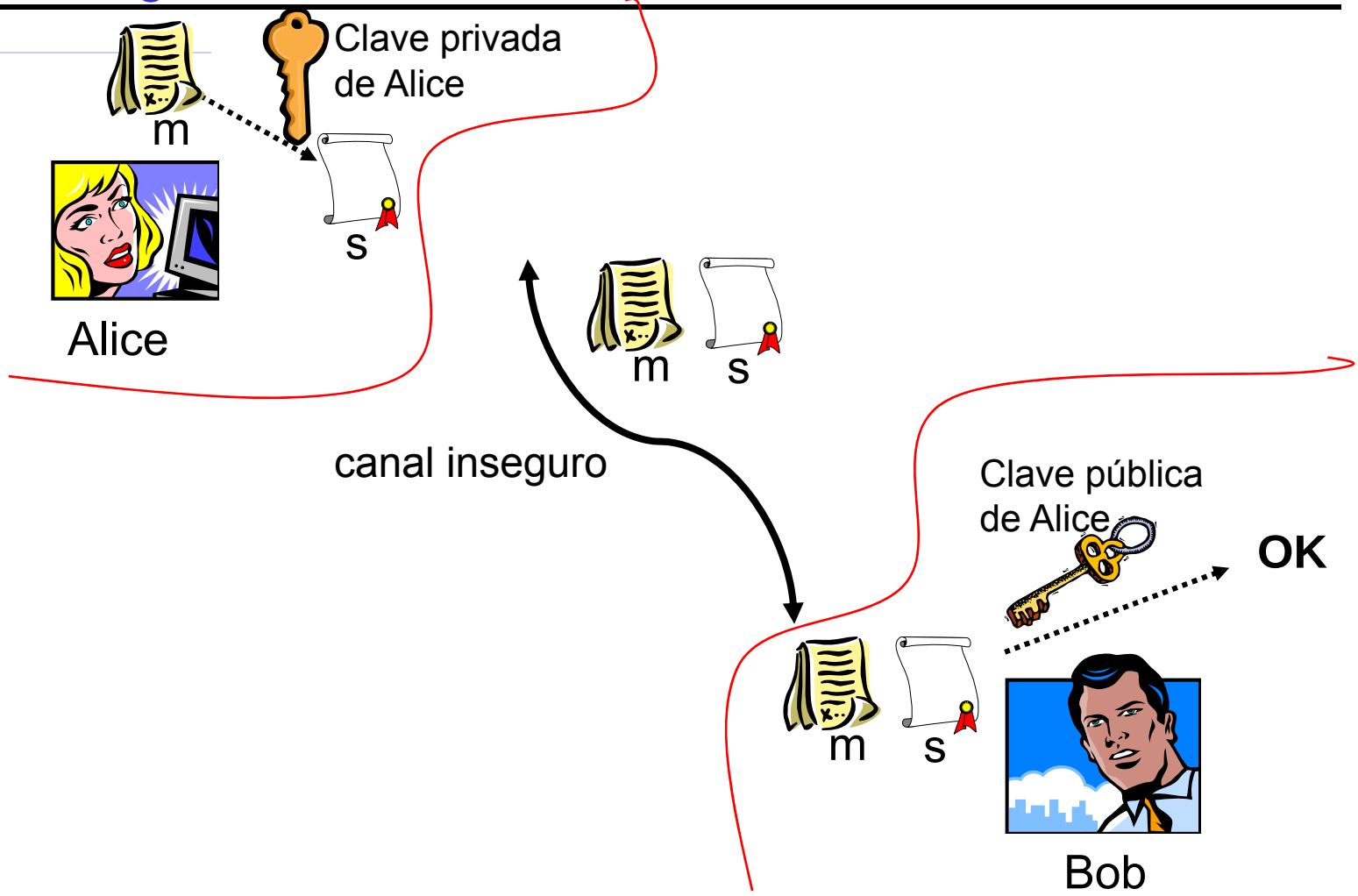
Firma digital: Protocolo básico

1. Alice firma su mensaje m : s=sig(m)
2. Alice manda [m,s] a Bob
3. Bob recibe [m,s]. Bob verifica s : ver(m,s)

Firma digital: Esquema de firma con clave pública

- Alice tiene dos claves ($A_{privada}$, $A_{pública}$)
- Alice firma usando $A_{privada}$
  - s= sig(m, $A_{privada}$)
  - Alice le manda a Bob (m,s)
- Bob verifica la firma usando $A_{pública}$
  - Bob calcula ver(m,s,$A_{pública}$)

Firma digital: Gráficamente ...



Firma digital: Protocolo de firma con clave pública

Alice tiene su clave privada $A_{privada}$ y todos tienen la clave pública de Alice, $A_{pública}$

1. Alice firma su mensaje m usando su clave privada: $s=sig(m, A_{privada})$
2. Alice manda [m,s] a Bob
3. Bob recibe [m,s].
4. Bob verifica s usando la clave pública de Alice: $ver(m,s,A_{pública})$

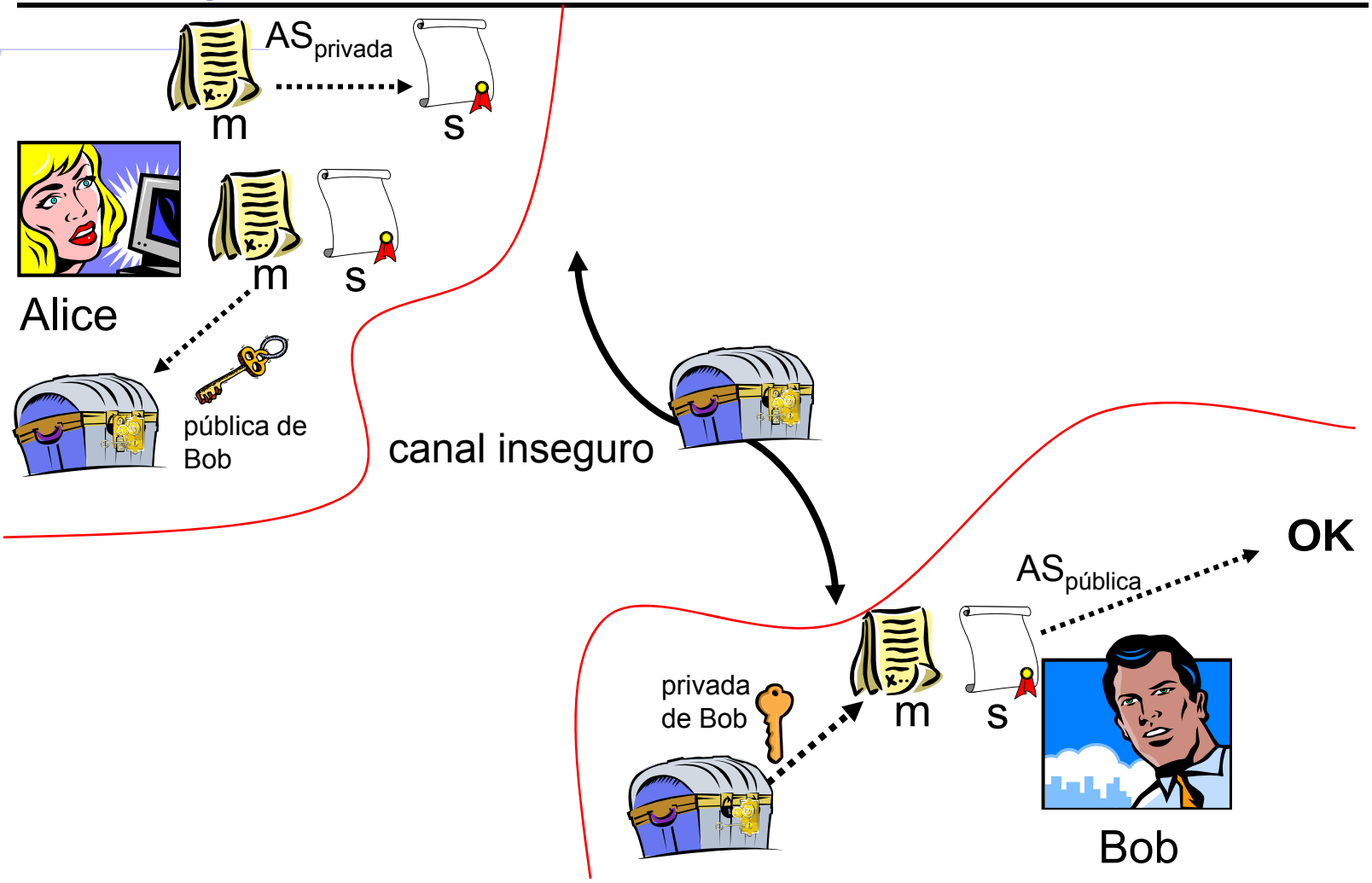
Firma digital: Problemas con el esquema anterior

- El tamaño de la firma puede ser proporcional al tamaño del mensaje
- El mensaje se manda sin cifrar
- La firma se puede reutilizar

Firma digital: Una solución simple

- Alice y Bob tienen dos pares de claves
  - Un par para cifrar ($AC_{privada}$, $AC_{pública}$)
  - Otro par para firmar ($AS_{privada}$, $AS_{pública}$)
- Alicia
  - Firma con $AS_{privada}$
  - Cifra [m, sig(m)] con $BC_{pública}$
- Bob
  - Descifra usando $BC_{privada}$
  - Verifica la firma usando $AS_{pública}$

Firma digital: Gráficamente ...



Firma digital: Protocolo de firma con clave pública

Alice tiene $AS_{privada}$, Bob tiene $BC_{privada}$ y todos tienen $AS_{pública}$, $BC_{pública}$

1. Alice firma su mensaje m usando su clave privada: s=sig(m, $AS_{privada}$)
2. Alice cifra [m,s] usando $BC_{pública}$:
   c=e([m,s], $BC_{pública}$)
3. Alice envía c a Bob
4. Bob descifra c usando su clave privada:
   d(c, $BC_{privada}$)= [m,s]
5. Bob verifica la firma ver(m,s,$AS_{pública}$)

Firma digital: Todavía hay problemas

- El tamaño de la firma puede ser proporcional al tamaño del mensaje
- El mensaje se manda sin cifrar ← solucionado
- La firma se puede reutilizar

Firma digital: La firma es reutilizable

- Bob puede mandar la firma de Alice a Oscar fingiendo ser Alice

Firma digital: Bob finge ser Alice

- Bob ha recibido e([m,sig(m, $AS_{privada}$)], $BC_{pública}$)
- Bob recupera [m,sig(m, $AS_{privada}$)] usando $BC_{privada}$
- Bob manda a Oscar

  e([m,sig(m, $AS_{privada}$)], $OS_{pública}$)

Firma digital: La firma es reutilizable

- Introducimos funciones hash para ayudarnos a construir firmas:
  - No reutilizables
  - Para mensajes largos

Funciones hash

- “Algoritmo de huella digital” o “transformación one-way”
- Las funciones hash transforman un mensaje de cualquier longitud en una cadena de longitud fija

- Dos propiedades:
  - One-way
  - Libre de colisiones

Funciones hash: One-way

- Dada la huella hash(m) es muy difícil adivinar m
- “muy difícil” quiere decir “computacionalmente imposible”

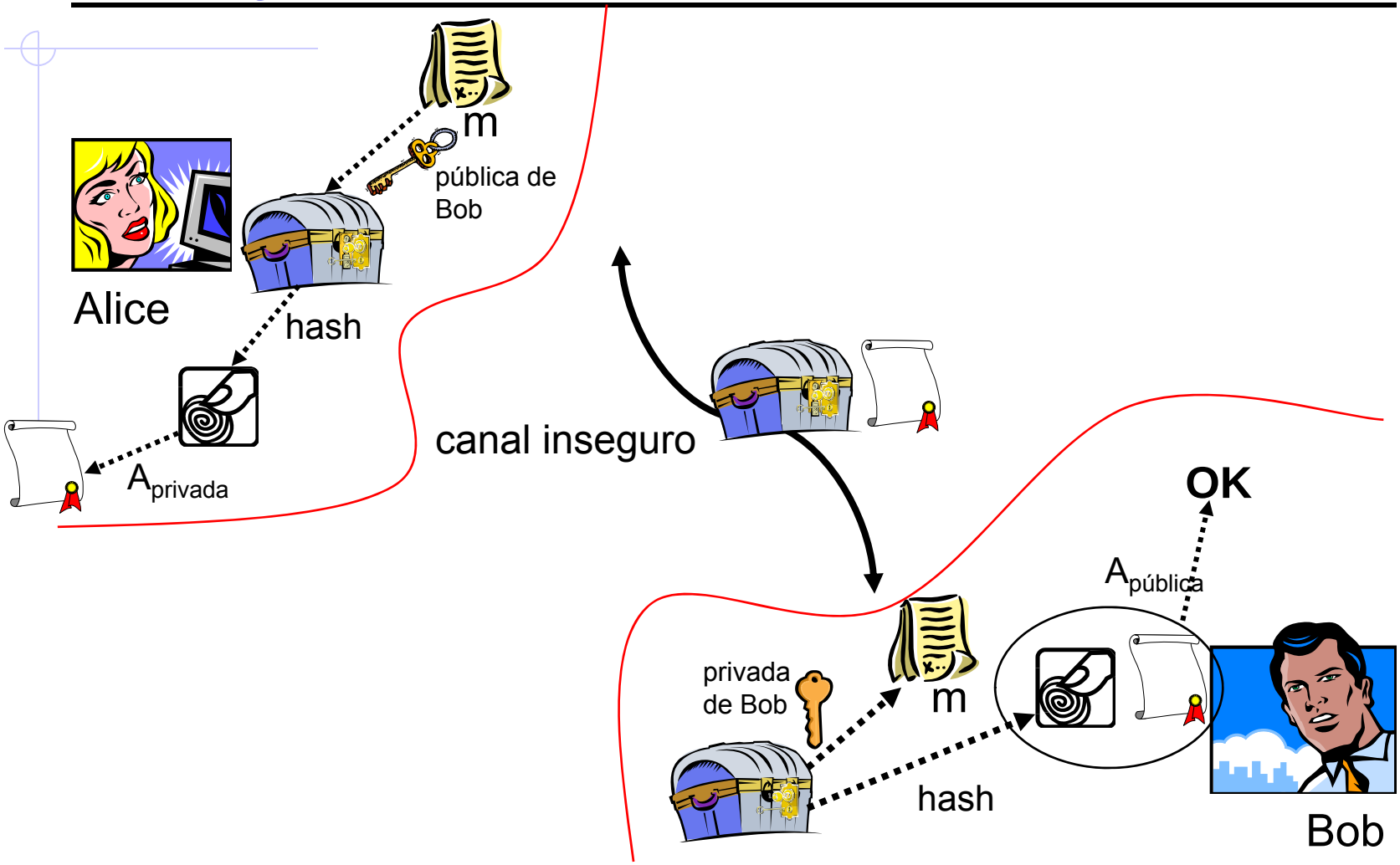
Funciones hash: Libre de colisiones

- Es muy difícil encontrar dos mensajes m1 y m2 den la misma huella hash(m1) = hash(m2)

Funciones hash: Solución para firmas reutilizables

- Alice calcula la huella de $e(m,B_{pública})$, es decir $h=hash(e(m,B_{pública}))$
- Alice le manda a Bob $e(m,B_{pública})$ y $sig(h,A_{privada})$

Firma digital: Gráficamente ...



Firma digital: Protocolo de firma no reutilizable

1. Alice cifra m usando $B_{pública}$:
   $c=e(m, B_{pública})$
2. Alice hashea c y lo firma usando su clave privada:
   $s=sig(h(c), A_{privada})$
3. Alice envía [c,s] a Bob
4. Bob descifra m usando su clave privada:
   $d(c, B_{privada})= m$
5. Bob hashea c y verifica la firma $ver(h(c),s,A_{pública})$

Funciones hash: Solución para firmas reutilizables

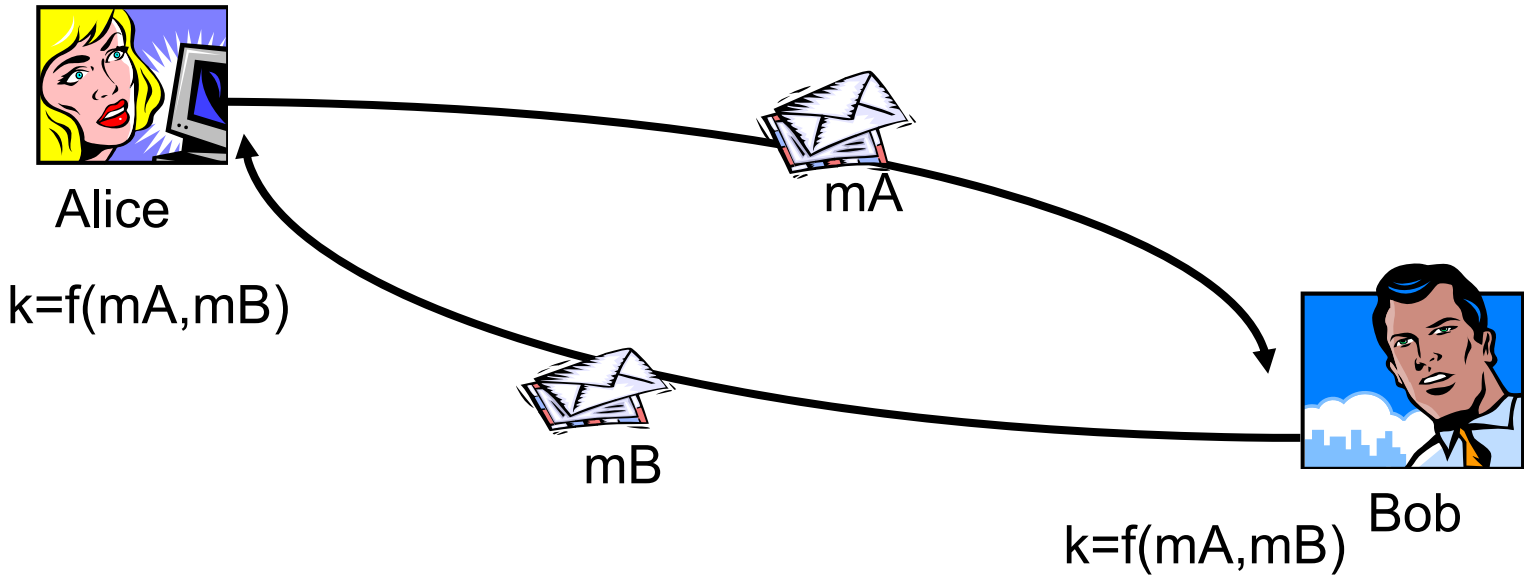
- Bob ya no puede reutilizar la firma (sólo sirve para hash(e(m,$B_{pública}$)))
- h es corto luego la firma es corta

Funciones hash: Solución para firmas reutilizables

- Todavía queda la posibilidad de que Bob diga que ha recibido el mismo mensaje varias veces (por ejemplo, un cheque electrónico)
- Eso se soluciona con timestamps

Intercambio de claves

- Es un protocolo mediante el cual 2 ó más partes establecen una clave secreta comunicándose por un canal público



Otra referencia (gratis)

Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone:
Handbook of Applied Cryptography

```
http://cplus.about.com/gi/dynamic/offsite.htm?zi=1/XJ/Ya&sdn=cplu
   s&cdn=compute&tm=9&f=11&su=p284.8.150.ip_&tt=14&bt=1&bts=0&zu=h
   ttp%3A//www.cacr.math.uwaterloo.ca/hac/index.html
```

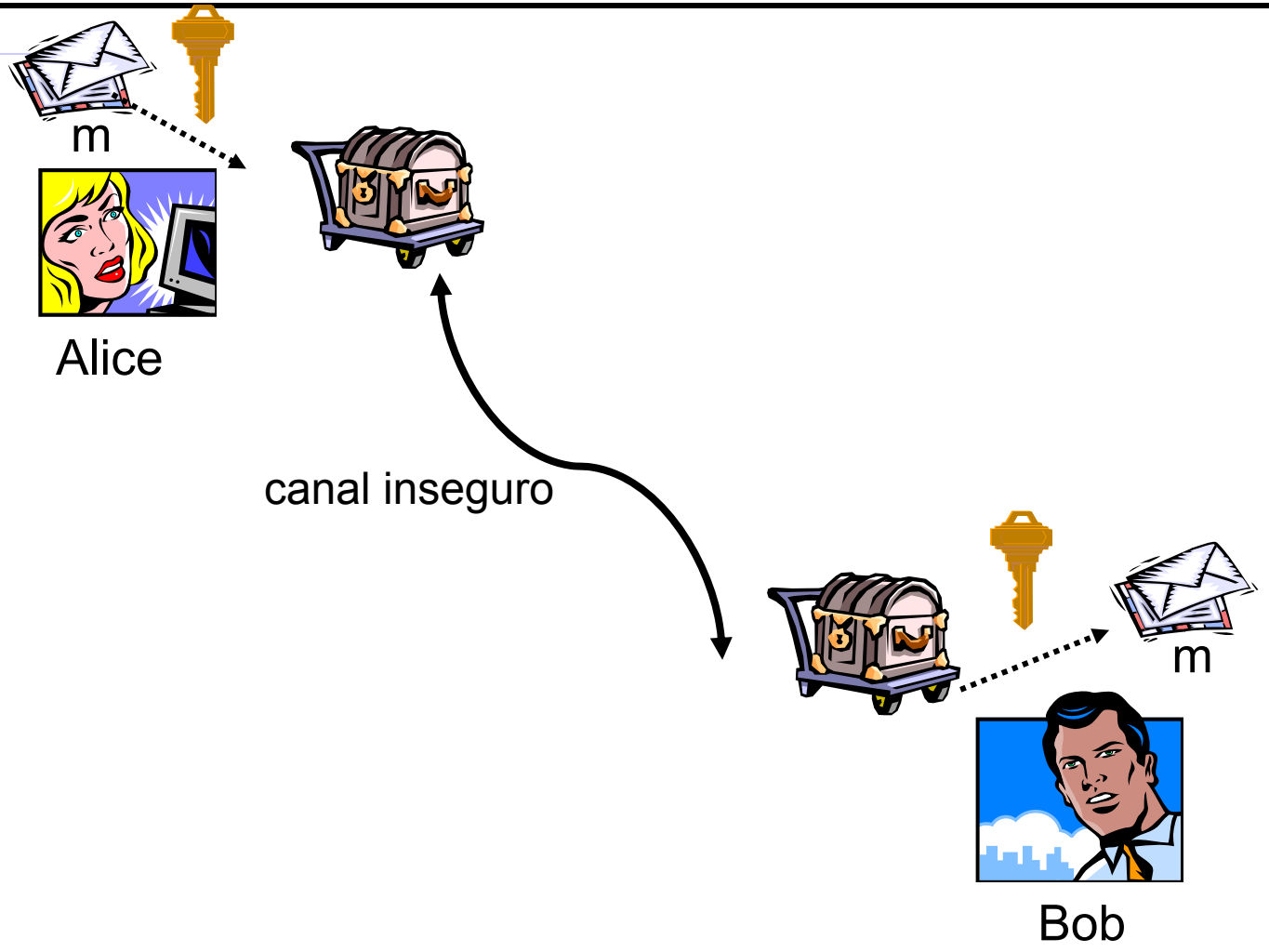
SEGUNDA PARTE

- Veremos sistemas criptográficos, esquemas de firma y funciones hash concretos
- Principalmente los estándares del NIST (National Institute of Standards and Technology)
- Son los estándares mínimos que el gobierno USA utiliza en todas sus agencias para comunicaciones no clasificadas
- Existen muchos más estándares (ANSI, ISO, por países, por ramas, ...)

SEGUNDA PARTE

- Sistemas criptográficos (todos de clave simétrica):
  - DES
  - Triple-DES
  - AES
- Esquemas de firma (todos de clave pública):
  - DSA
  - RSA
- Funciones hash:
  - SHA1 (SHA256, SHA384, SHA512, SHA224)
- Intercambio de claves:
  - Diffie-Hellman (no estándar)

Cifradores de clave simétrica



Cifradores de clave simétrica

Formados a base de

- Sustituciones
- Permutaciones

Dos técnicas clásicas de cifrado

El cifrado de César

- Las letras del mensaje se cambian por 3 más

```
A B C D E F G H I J K L M
D E F G H I J K L M N O P
```

- “ataque a media noche” se cifra como
  “dwdtxh d phgld qrfkh”

Cifradores de clave simétrica: Sustitución

El cifrado de César

- El salto puede ser cualquiera
- Cifrado y descifrado son igual de fáciles
- Demasiado fáciles ... Sólo hay 25 claves diferentes
  - El espacio de claves es demasiado pequeño

Cifradores de clave simétrica: Sustitución general

- Hay una tabla que muestra cómo cifrar cada letra del mensaje

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

- PERRO se cifra como YBUUE
- Hay 26! claves (>4×10^{26})
  - Mayor que el espacio de claves de DES (<10^{17})
- Esto elimina los ataques de fuerza bruta

Análisis estadístico

- El método anterior se puede romper fácilmente explotando las regularidades del lenguaje
- Primero contamos la frecuencia relativa de las letras en el mensaje cifrado
- Lo comparamos con la frecuencia estándar en el idioma
- Suficiente para romper mensajes largos

Análisis estadístico

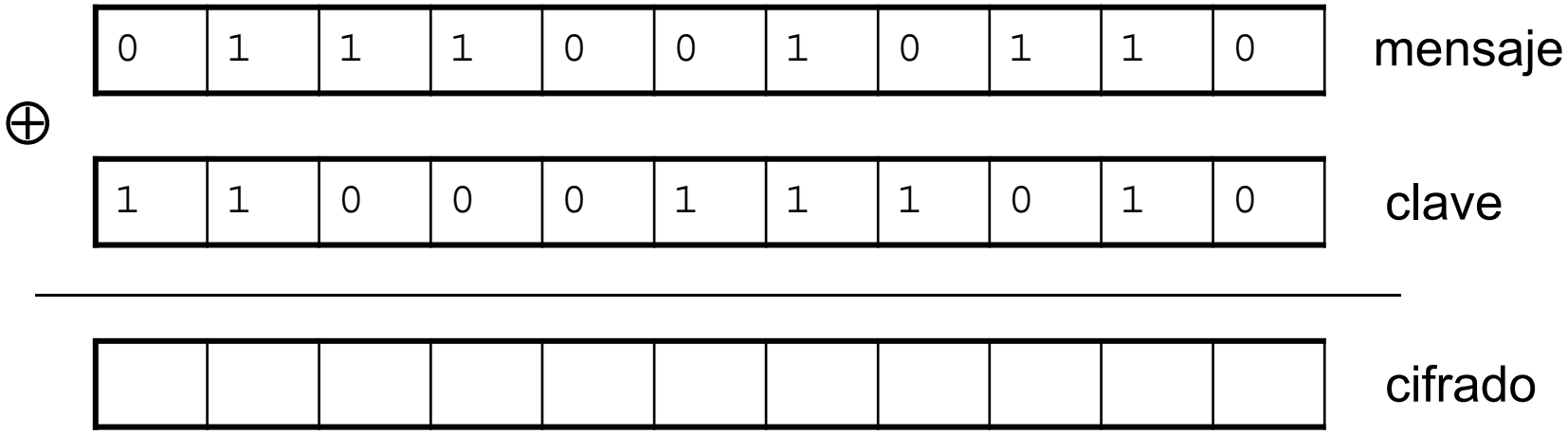
Frecuencias en inglés:

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
7,3	1,3	3,5	4,3	12,8	3,0	2,0	3,5	7,8	0,3	0,5	3,7	2,8	7,8	7,5	2,8	0,5	8,5	6,0	9,3	3,0	1,5	1,5	0,5	2,3	0,3

- La letra más frecuente corresponderá a la E, la siguiente la T ...
- También se puede usar qué letras van más a menudo juntas, etc

One-time pad

- Cogemos el mensaje en binario y hacemos el XOR con una clave binaria (que sólo usamos una vez)



Cifradores de clave simétrica: Sustitución one-time

- Simétrico
  - Cifrar $c = m \oplus k$
  - Descifrar $m = c \oplus k$
- Incondicionalmente seguro
  - Sólo podemos adivinar m
- Sólo una vez
  - Enviar la clave es tan costoso como enviar el mensaje

Cifradores de clave simétrica: Sustitución one-time

Usado por:

- Los espías rusos
- El teléfono rojo “Washington-Moscú”
- Los mensajes del Che a Castro

Cifradores de clave simétrica

Formados a base de

- Sustituciones
- Permutaciones

Cifradores de clave simétrica: Permutación

Permutación por columnas:

```
BUSH SE LIBRO DE LA GUARDIA NACIONAL POR UNA
   FIRMA DE SU PADRE

BUSHSELIBR
ODELAGUARD
IANACINALP
ORUNAFIRMA
DESUPADRE

BOIOD UDARE SENAUS HLANU SACAP EGIFA LUNID IAARR
   BRLME RDPA
```

Cifradores de clave simétrica

Formados a base de

- Sustituciones: para oscurecer la relación entre mensaje y mensaje cifrado
- Permutaciones: para oscurecer las redundancias en el mensaje (qué letra va después de cuál, etc)

Cifradores de clave simétrica: DES

- Es un cifrador por bloques: cada mensaje m se parte en bloques de 64 bits que se cifran por separado
- Se añaden bits al final para que la longitud del mensaje sea múltiplo de 64
- En lo siguiente veremos **cómo cifrar un bloque de 64 bits**

Cifradores de clave simétrica: DES

- Usa una clave secreta de 56 bits k
- Un cifrado está formada por **16 rondas**
- Cada ronda usa sustituciones y permutaciones
- Cada ronda usa una subclave distinta de 48 bits, k_i, obtenida permutando k y quedándote con parte

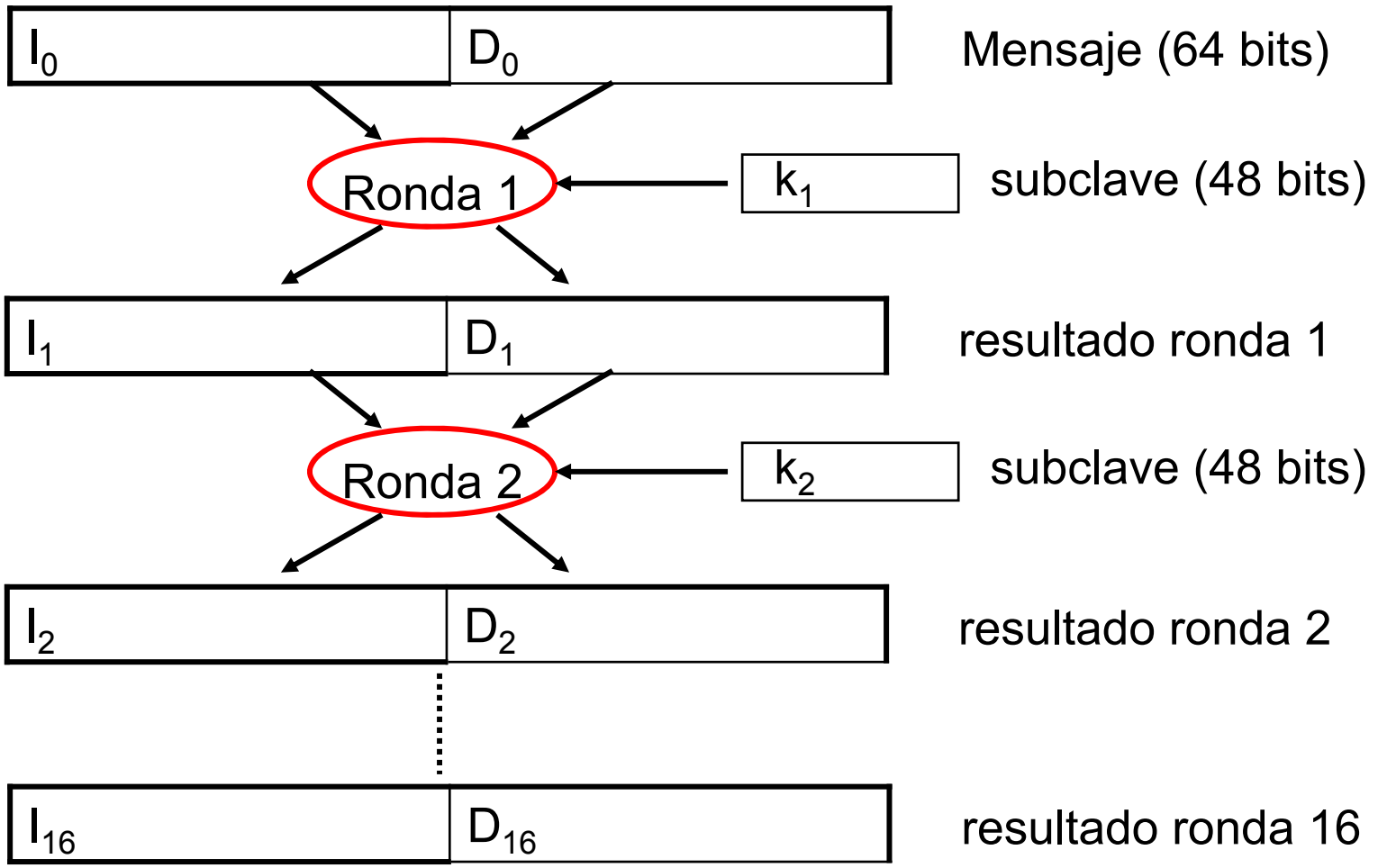
  $k_1, k_2, ..., k_{16}$

Cifradores de clave simétrica: DES

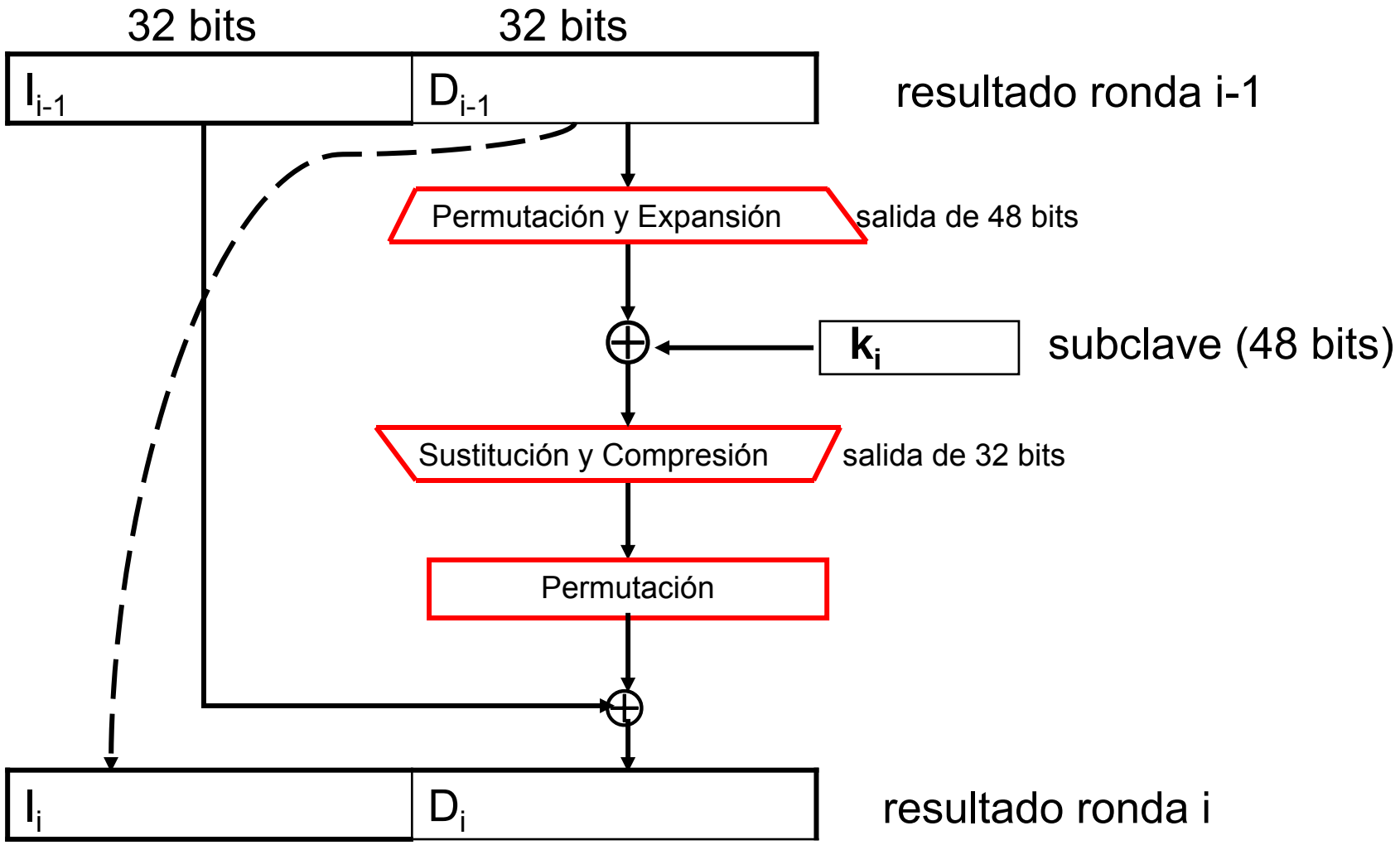
La ronda número i:

- Coge los 64 bits resultado de la ronda i-1
- Los parte en dos trozos de 32 bits, D_{i-1}, I_{i-1}
- Utiliza la subclave k_i

Cifradores de clave simétrica: DES



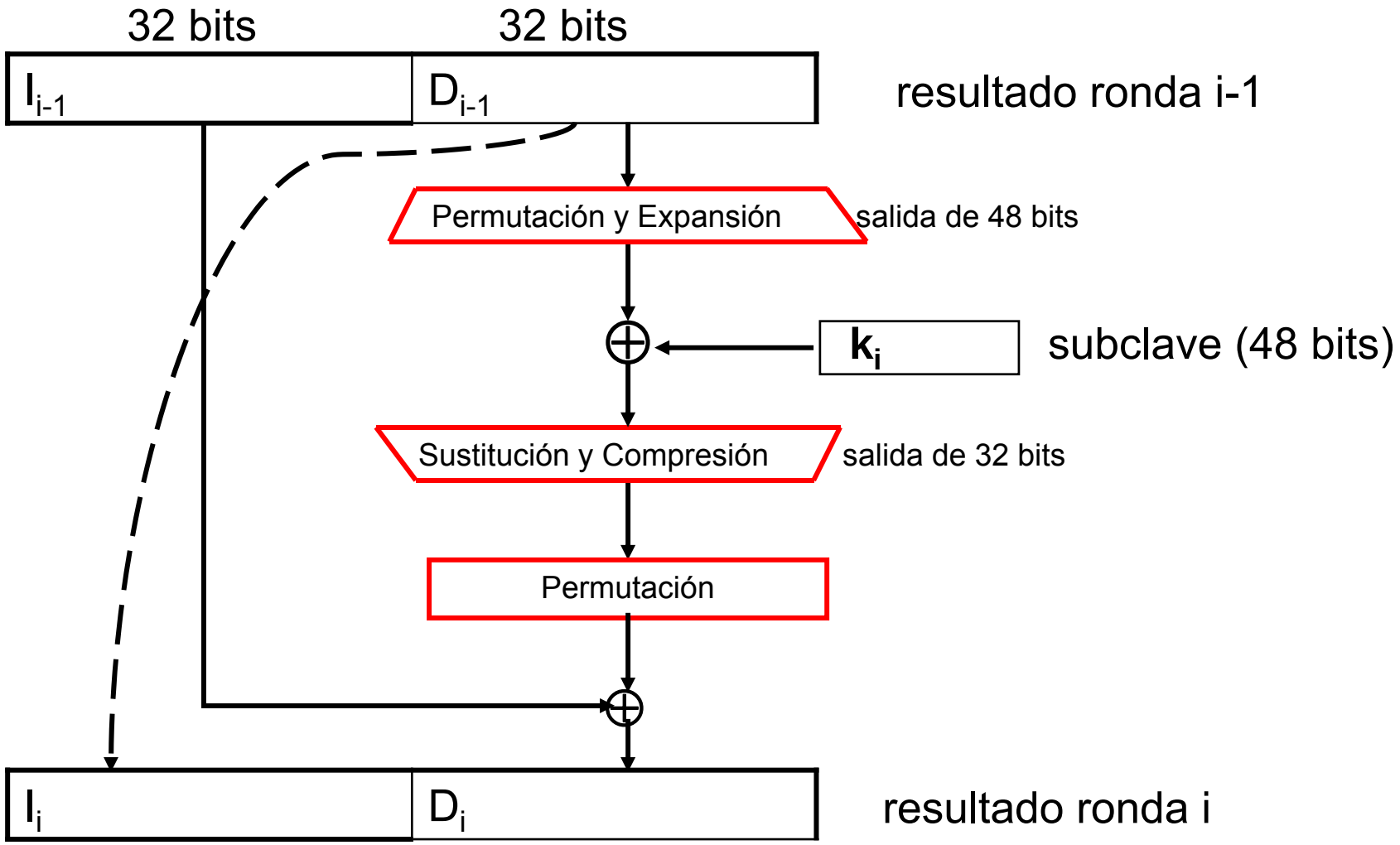
Cifradores de clave simétrica: una ronda de DES



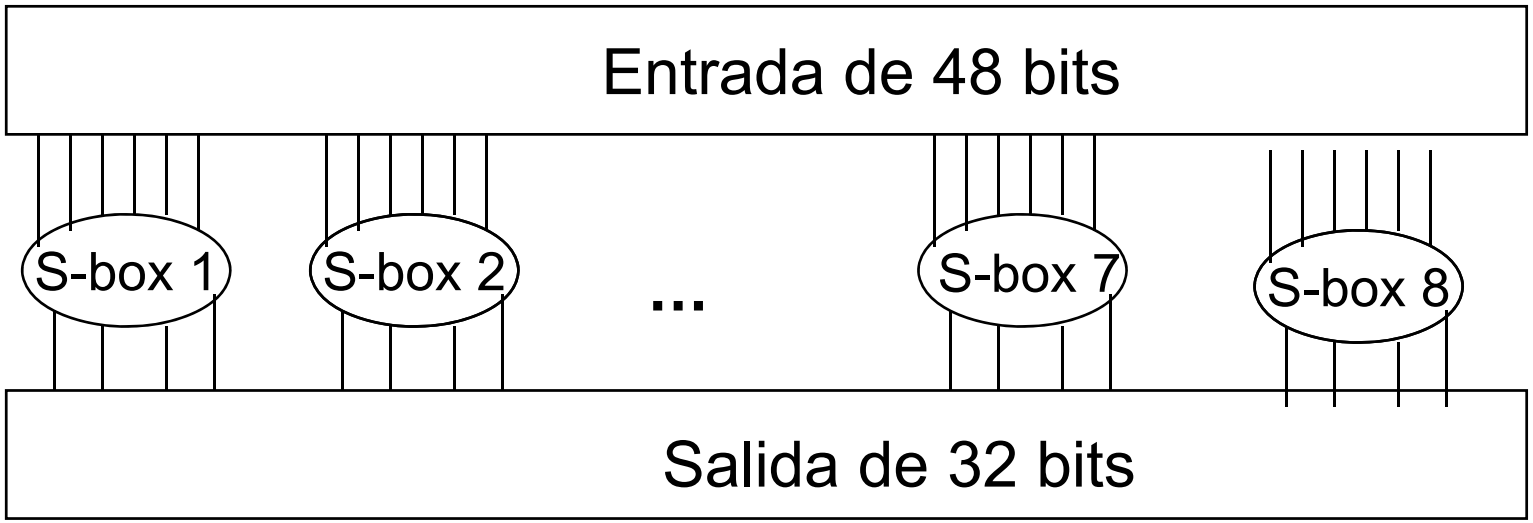
Permutación y Expansión

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Cifradores de clave simétrica: una ronda de DES



Sustitución y Compresión (S-Boxes)

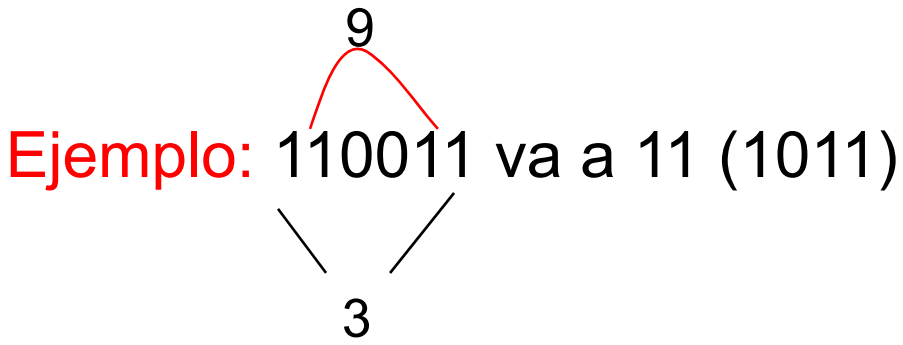


El misterio está en las S-boxes …

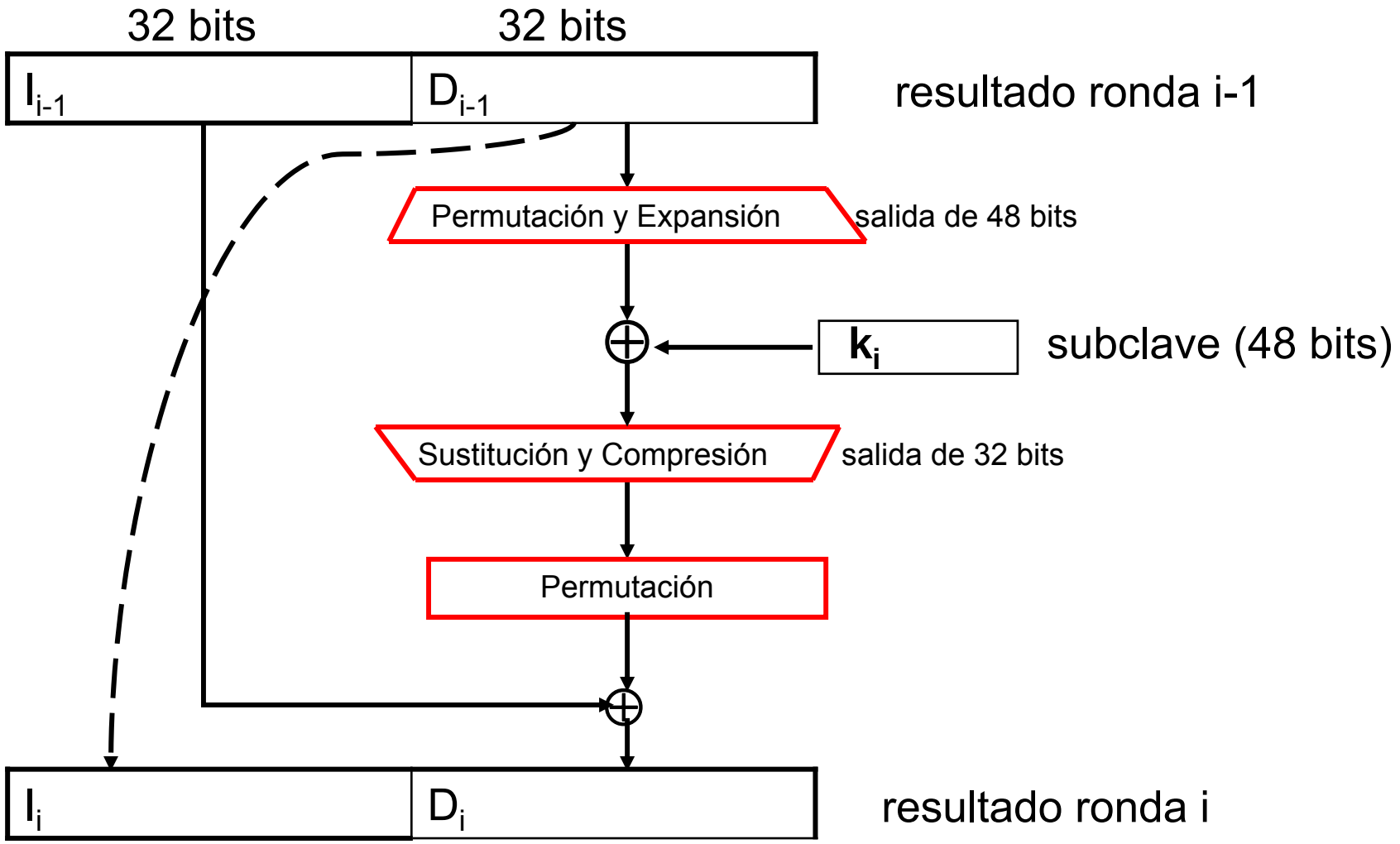
Sustitución y Compresión (S-Boxes)

S1

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



Cifradores de clave simétrica: una ronda de DES



Cifradores de clave simétrica: descifrar en DES

- Dado un mensaje cifrado (de 64 bits)
- Para descifrar se usa exactamente el mismo algoritmo
- Lo único que cambia es que ahora las subclaves se usan en orden contrario

  $k_{16}, k_{15}, ..., k_1$

Cifradores de clave simétrica: historia de DES

- El NIST lo propuso como estándar en 1977
- Lo propusieron como implementable en hard
- De hecho se equivocaron, pensaban que el estándar no revelaría suficiente para la implementación en soft
- Renovado cada 5 años desde entonces hasta 2005
- Desde 1981 se usa generalizadamente en el sector privado
- Se sabe que NSA (National Security Agence) podía romperlo sin problemas
- El resto pagando ($10.000 en 2006)

¿Por qué es/no es seguro DES?

- Sin saber la clave es como un “barajeo” perfecto de las palabras de 64 bits

- Probar todas las claves de 56 bits tarda:

Suponemos que podemos probar 10^9 claves por segundo:

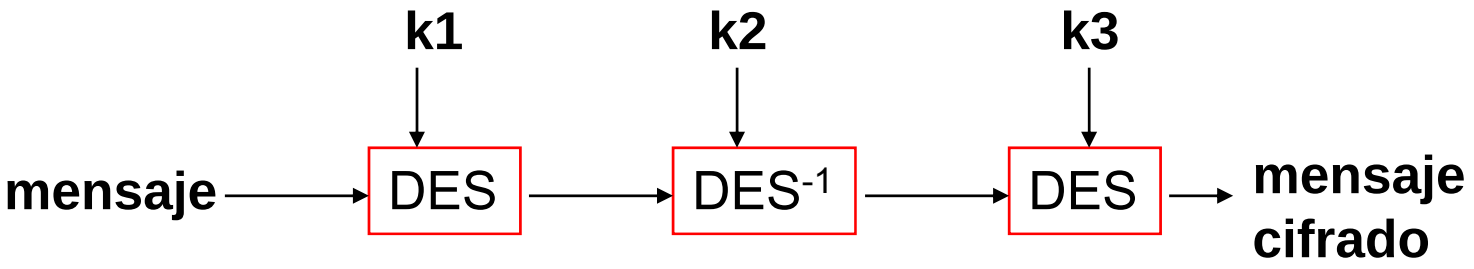
$$2^{56} = 10^{17}$$

Tardamos 10^8 segundos = 3 años

100% paralelizable

Cifradores de clave simétrica: Triple DES

- También estándar NIST (1999)
- Usa 3 claves DES



Cifradores de clave simétrica: AES

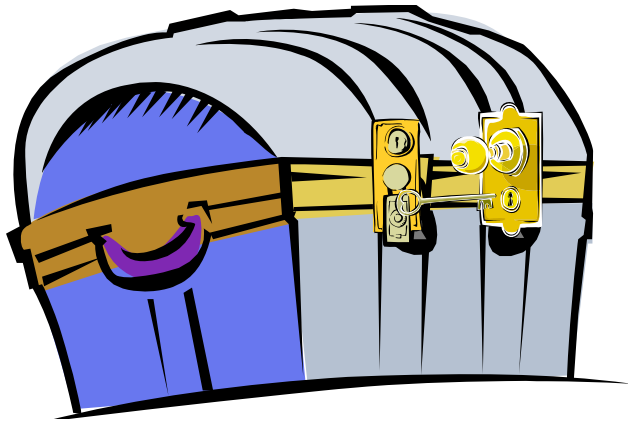
- También estándar NIST (2001)
- Alternativa preferida: mejor que DES
- También por bloques, también usa sustitución y permutación
- También implementable en hard y descifrado muy similar al cifrado
- Clave de 128, 192 ó 256 bits
- Bloques de 128 bits

Cifradores: Más sobre estándares

- Hay un cuarto estándar de cifrado NIST: Skipjack (1994), pero de descripción clasificada
- Fue muy criticado que NIST no tenga estándares de cifrado con clave pública
- Los estándares de firma digital (que son de clave pública) se pueden usar para cifrado. RSA sin modificación alguna

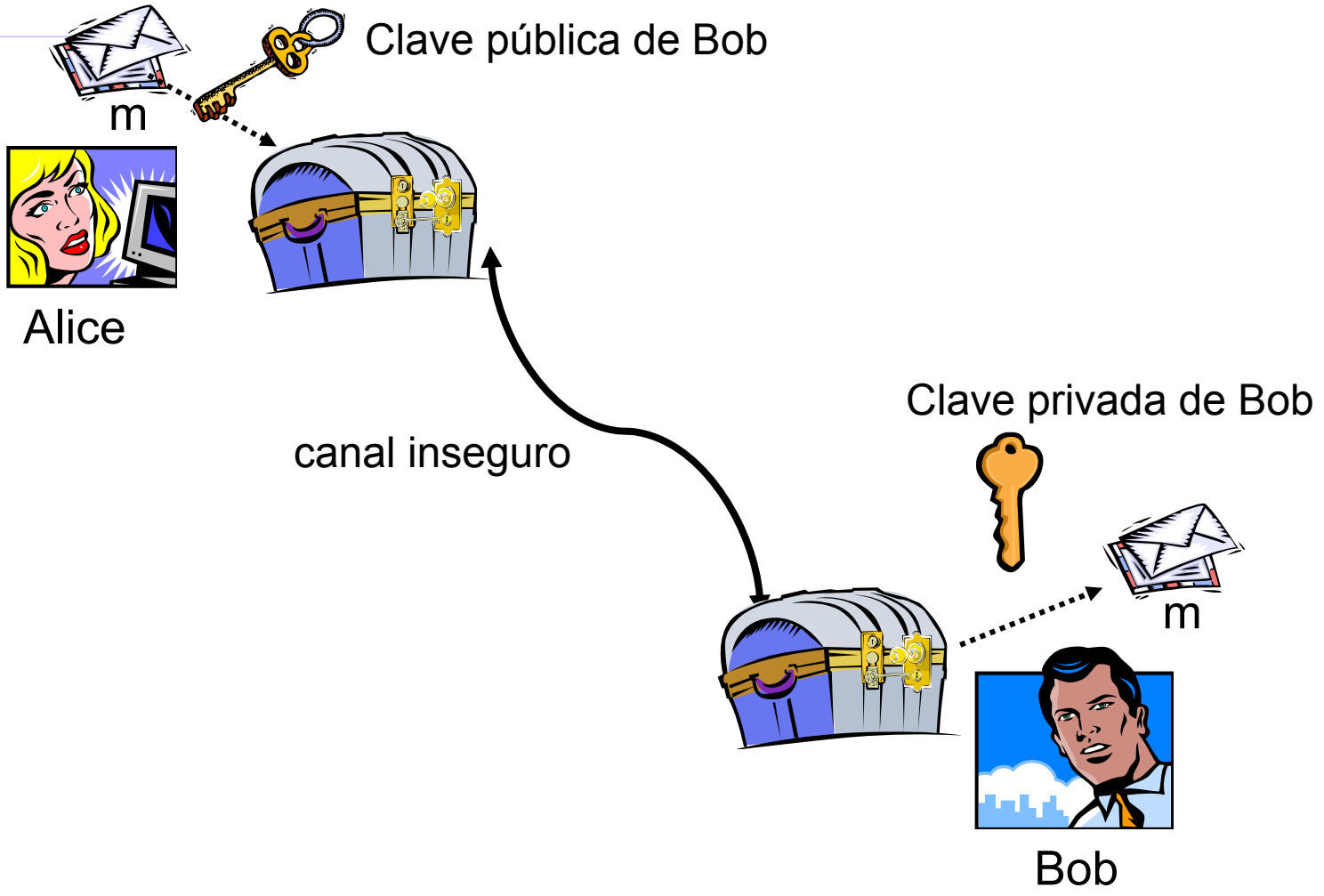
RSA como cifrador: clave pública

- Los sistemas de clave pública usan un tipo especial de cofre con dos cerraduras
  - Con una llave (pública) cerramos el cofre
  - Con la otra (privada) abrimos el cofre



Clave privada ≠ Clave pública

RSA como cifrador: clave pública



RSA como cifrador

- Propuesto en 1978 por Rivest, Shamir y Adelman
- No aceptado como estándar NIST (y sólo de firma) hasta 2000
- Su seguridad se basa en la dificultad de factorizar números grandes

RSA como cifrador

Generación de claves:

- Elegir dos números primos p y q (de aproximadamente el mismo número de bits y de al menos 512 bits cada uno)
- Elegir aleatoriamente e que cumpla mcd(e, (p-1)(q-1)) = 1
- Calcular d tal que $ed \equiv 1 \pmod{(p-1)(q-1)}$
- n=pq
- Clave pública n,e
- Clave privada d

OJO: mantened p y q secretos

MUY INTERESANTE: $(m^e)^d \equiv m \pmod{n}$ para cualquier m

RSA como cifrador

¿Por qué ...

- $(m^e)^d \equiv m \pmod{n}$ para cualquier m
  - Teoría de números ...

- hay que mantener p y q secretos?
  - Porque si no se podría encontrar d a partir de e
  - Por eso mismo si se factoriza n=pq se rompe RSA

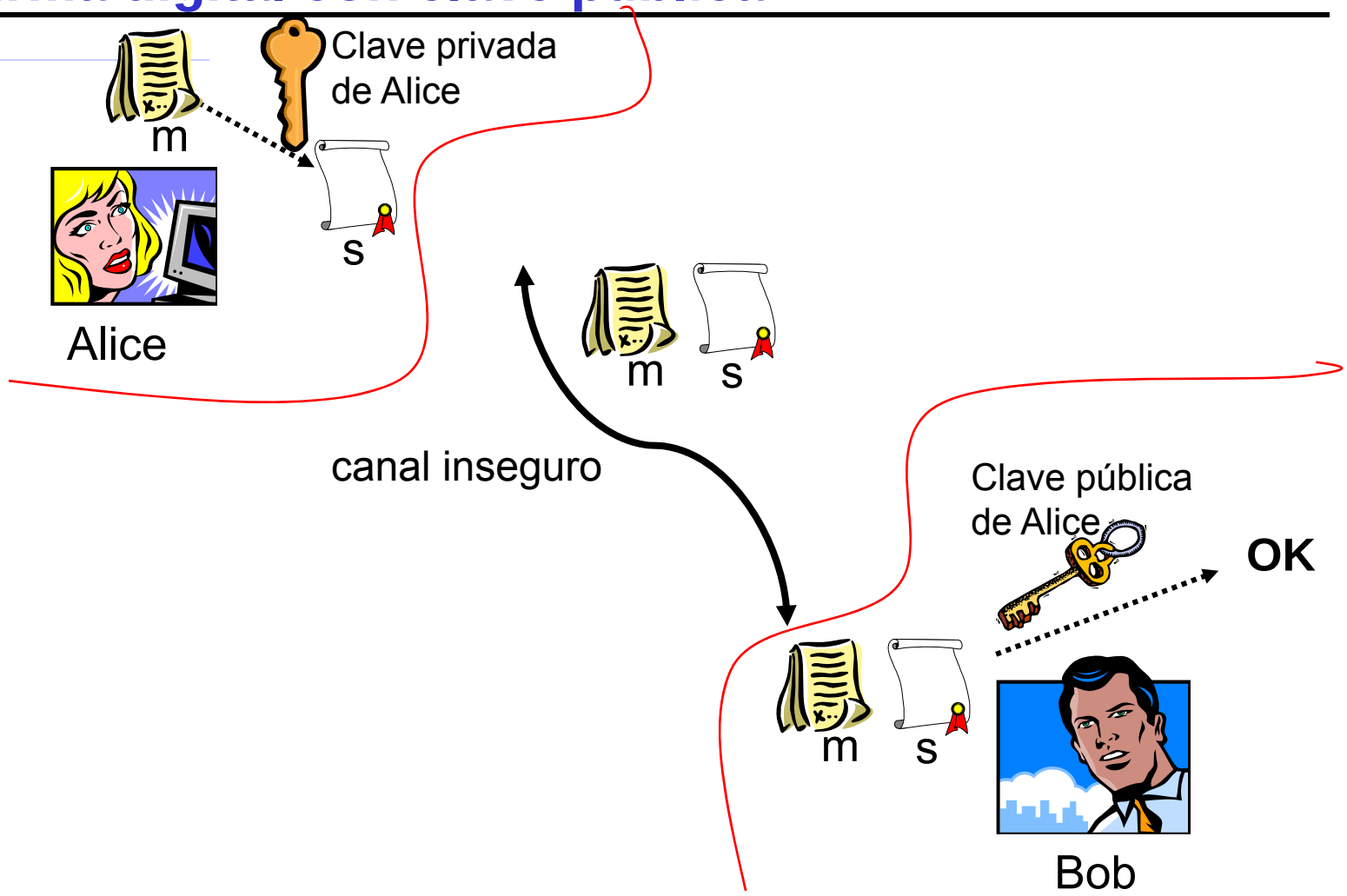
RSA como cifrador

- Clave pública n,e
- Clave privada d
- Cumplen que $(m^e)^d \equiv m \pmod n$

- Para cifrar un número m menor que n:
  - Cifrado: $c = m^e \bmod n$
  - Descifrado: $m = c^d \bmod n$

- Los números más grandes se parten en bloques menores que n

Firma digital con clave pública



Firma digital: Protocolo de firma con clave pública

Alice tiene su clave privada $A_{privada}$ y todos tienen la clave pública de Alice, $A_{pública}$

1. Alice firma su mensaje m usando su clave privada: $s=sig(m, A_{privada})$
2. Alice manda [m,s] a Bob
3. Bob recibe [m,s].
4. Bob verifica s usando la clave pública de Alice: $ver(m,s,A_{pública})$

RSA como firma digital

- Clave pública n,e
- Clave privada d
- Cumplen que $(m^e)^d \equiv m \pmod n$

- Para firmar un número m menor que n:
  - Firma: $s = m^d \bmod n$
  - Verificación: comprobar que $m = s^e \bmod n$

- Estándar NIST desde 2000

Para usar RSA

- Muy importante: no usar la misma clave para firma y para cifrado
- Un grupo de usuarios no debe usar un n común

DSA: otro algoritmo de firma digital estándar NIST

- Conocidos y reutilizados: p, q, g números (ver condiciones)
- Clave privada x (<q)
- Clave pública y (= g^x mod p)

- Para firmar un número m menor que q:
  - Firma: ciertas cuentas con p, q, g, x, m
  - Verificación: comprobaciones con p, q, g, y, m

IMPORTANTE: Si se puede calcular x a partir de y se rompe.

Claves de DSA

Clave pública

- p primo de entre 512 y 1024 cifras
- q factor primo de $p - 1$ de 160 bits
- $g = h^{(p-1)/q} \bmod p$ (con $h < p-1$, $g > 1$)
- $y = g^x \bmod p$

Clave privada

- $x < q$ de 160 bits

Cálculos de DSA

Firmar

- Elegir aleatoriamente $k < q$
- $s1 = (g^k \bmod p) \bmod q$
- $s2 = (m + s1*x)k^{-1} \bmod q$
- La firma es (s1,s2)

Verficar

- $w = (s2)^{-1} \pmod q$
- $u1 = m*w \pmod q$
- $u2 = s1*w \pmod q$
- $v = [g^{u1}*y^{u2} \bmod p] \bmod q$
- Firma válida si v = s1

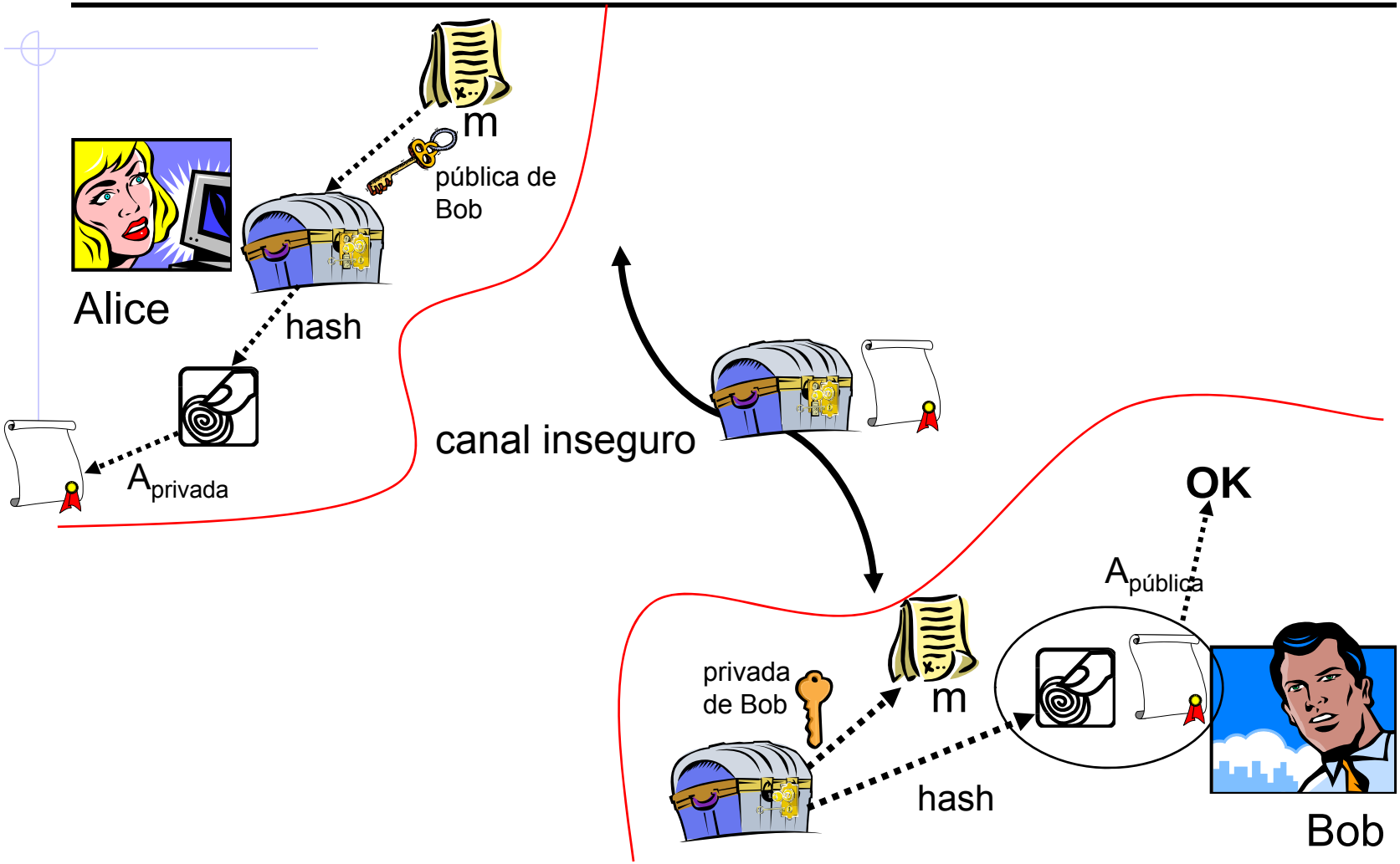
DSA versus RSA

- DSA fue el estándar NIST desde 1991
- Esto provocó muchas críticas porque RSA era el estándar “de facto”
- Ahora son estándar los dos (2000)
- DSA es más lento que RSA en verificación
- DSA es más seguro

Pero ...

- Quedamos en que antes de firmar pasábamos los mensajes por la función hash

Gráficamente ...



Funciones hash o huellas

- Las funciones hash transforman un mensaje de cualquier longitud en una cadena de longitud fija (la "huella" del mensaje)

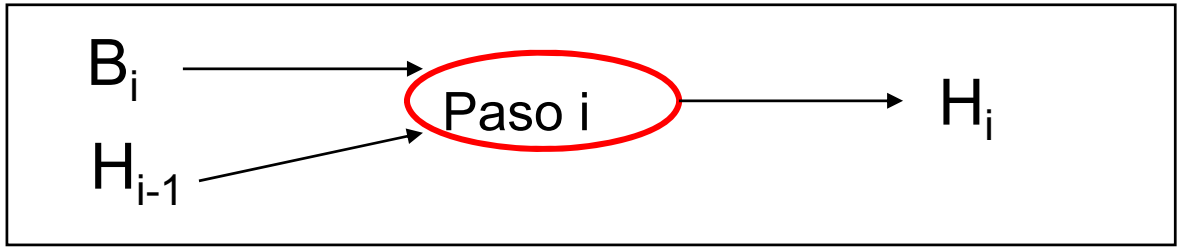
- Dos propiedades:
  - One-way: Dada la huella hash(m) es muy difícil adivinar m
  - Libre de colisiones: Es muy difícil encontrar dos mensajes m1 y m2 den la misma huella

    hash(m1) = hash(m2)

Funciones hash: estándar NIST (1995)

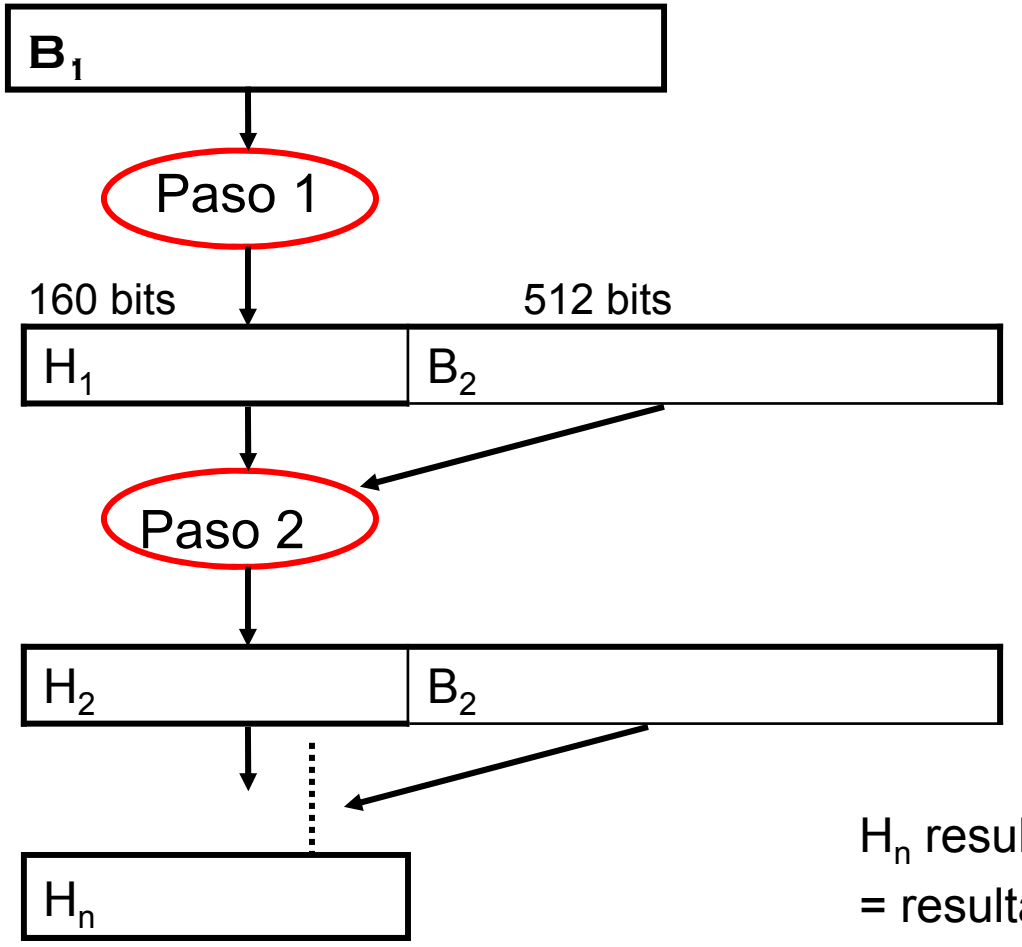
- SHA1 (también SHA256, SHA384, SHA512)
- La entrada se divide en bloques de 512 bits

  $M = B_1 B_2 \ldots B_n$



- El resultado final tiene 160 bits

Funciones hash: SHA1

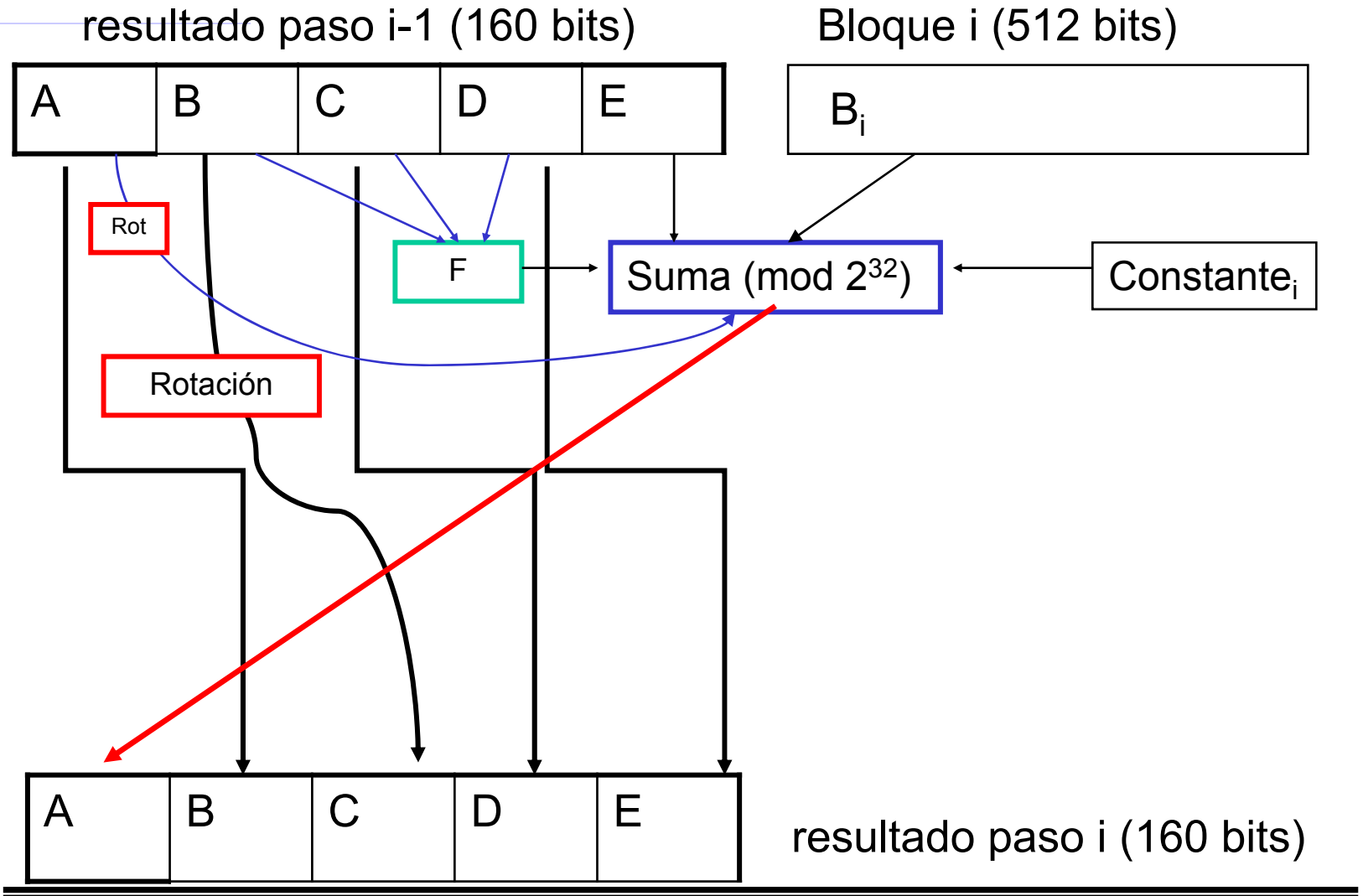


Bloque 1 (512 bits)

H_1 TIENE 160 BITS

H_n resultado paso n (160 bits)
= resultado final

Un paso de SHA1 de DES



Romper SHA1

Para encontrar una colisión como el hash da resultado de 160 bits queremos que sea necesario probar $2^{80} = 10^{24}$ parejas de mensajes aleatorios

Si podemos probar 10^9 mensajes por segundo, tardaríamos $2*10^{15}$ segundos = $6 * 10^7$ años

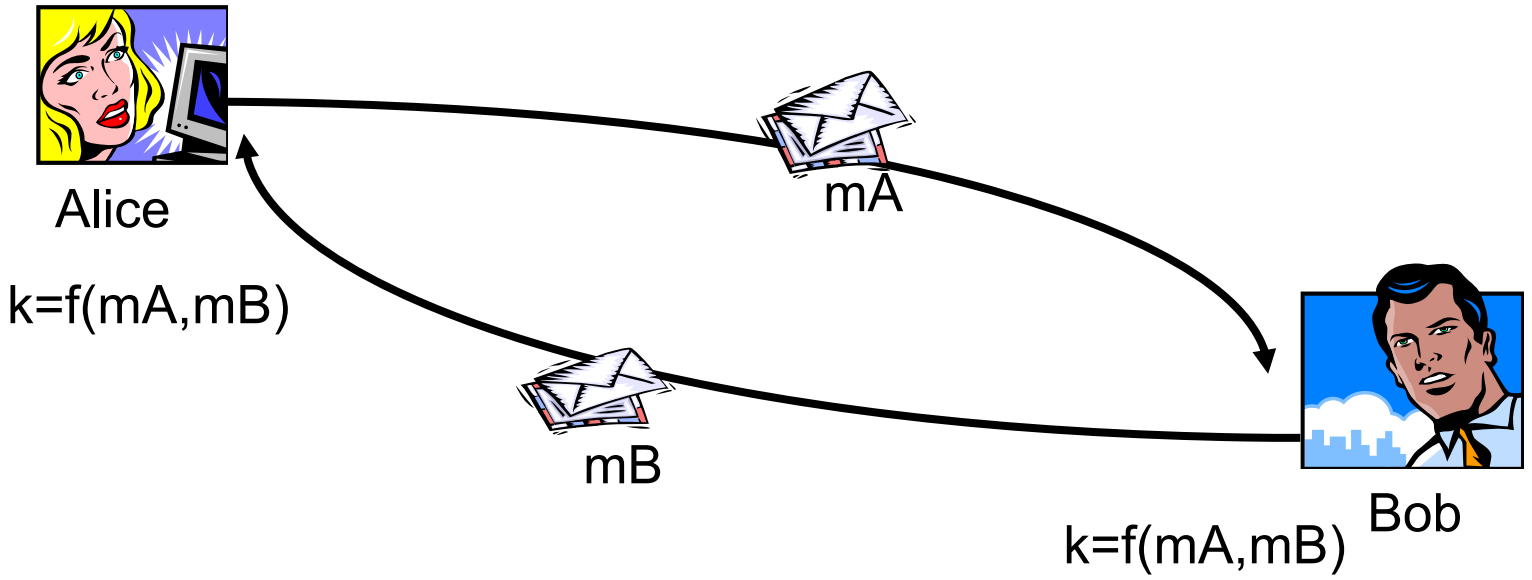
En febrero “rompieron” SHA1 reduciendo encontrar una colisión a probar $2^{69} = 10^{21}$ parejas de mensajes aleatorios

$2*10^{12}$ segundos = $6 * 10^4$ años

100% paralelizable

Intercambio de claves

- Es un protocolo mediante el cual 2 ó más partes establecen una clave secreta comunicándose por un canal público



Intercambio de claves

- No hay estándar NIST (hay algo bastante desfasado con clave privada)
- El protocolo de Diffie-Hellman (1976) es de los más usados
- Se basa en la dificultad de calcular x a partir de

  $y = g^x \bmod n$

Intercambio de claves: Diffie-Hellman

- Públicos:
  - p primo,
  - g con $\{g^i \mid 0 \leq i \leq p-2\} = \{1, 2, ..., p-1\}$
- Alice elige aleatoriamente x, calcula **X**= g^x mod p
- Bob elige aleatoriamente y, calcula **Y**= g^y mod p
- Alice manda **X** a Bob, Bob manda **Y** a Alice
- Alice calcula k= $\mathbf{Y}^x$ mod p
- Bob calcula k'= $\mathbf{X}^y$ mod p

k=k' ← ¡Están de acuerdo!

Final: Las 10 mayores amenazas a la seguridad

1. Errores en el software
2. Mala protección contra ataques anónimos
3. No hay donde guardar secretos
4. Malos generadores aleatorios
5. Mala elección de passwords
6. Confianza inmerecida en el canal de comunicación
7. Protocolos y sistemas mal entendidos
8. Mal asesoramiento sobre amenazas y riesgos
9. Hacer la seguridad cara y complicada de usar
10. Poca demanda de seguridad