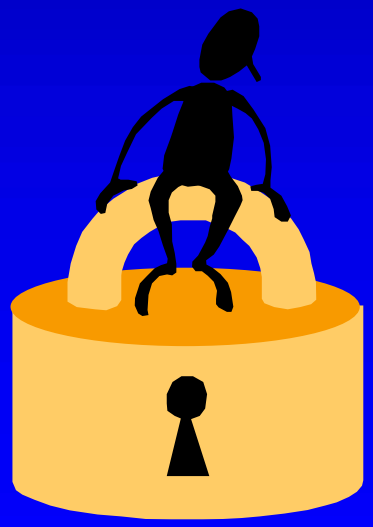
Criptografía

Versión original :Miquel Soriano
E-mail: soriano@mat.upc.es
Dpto. Matemática Aplicada y Telemática (UPC)
Actualización: Diego Ponce
E-mail: diego.ponce@ucuenca.edu.ec
Maestría de Telemática (UC)

Indice

- Criptografía
  - Introducción
  - Criptografía clásica
  - Cifrado simétrico: Métodos en flujo y en bloque
  - Cifrado asimétrico
  - Funciones de hash
  - Firma digital
  - Gestión de claves
  - Certificados
  - Esteganografía



Introducción

- Criptologia: cryptos (oculto) + logos (tratado)
- **Criptología**:Criptografía y Criptoanálisis
  - **Criptografía**: Comunicaciones en un *ambiente:*
    - Hostil
    - Vulnerable
    - Desconfianza mutua entre los comunicantes
  - **Criptoanálisis**: Violación de los sistemas criptográficos

Objetivos de la Criptografía

- Privacidad. Un usuario no autorizado no puede conocer el contenido
- Autenticidad. El destinatario tiene la certeza de que la comunicación proviene del origen supuesto.
- Verificabilidad. El destinatario tiene la capacidad de demostrar ante terceros el contenido y procedencia de una comunicación.

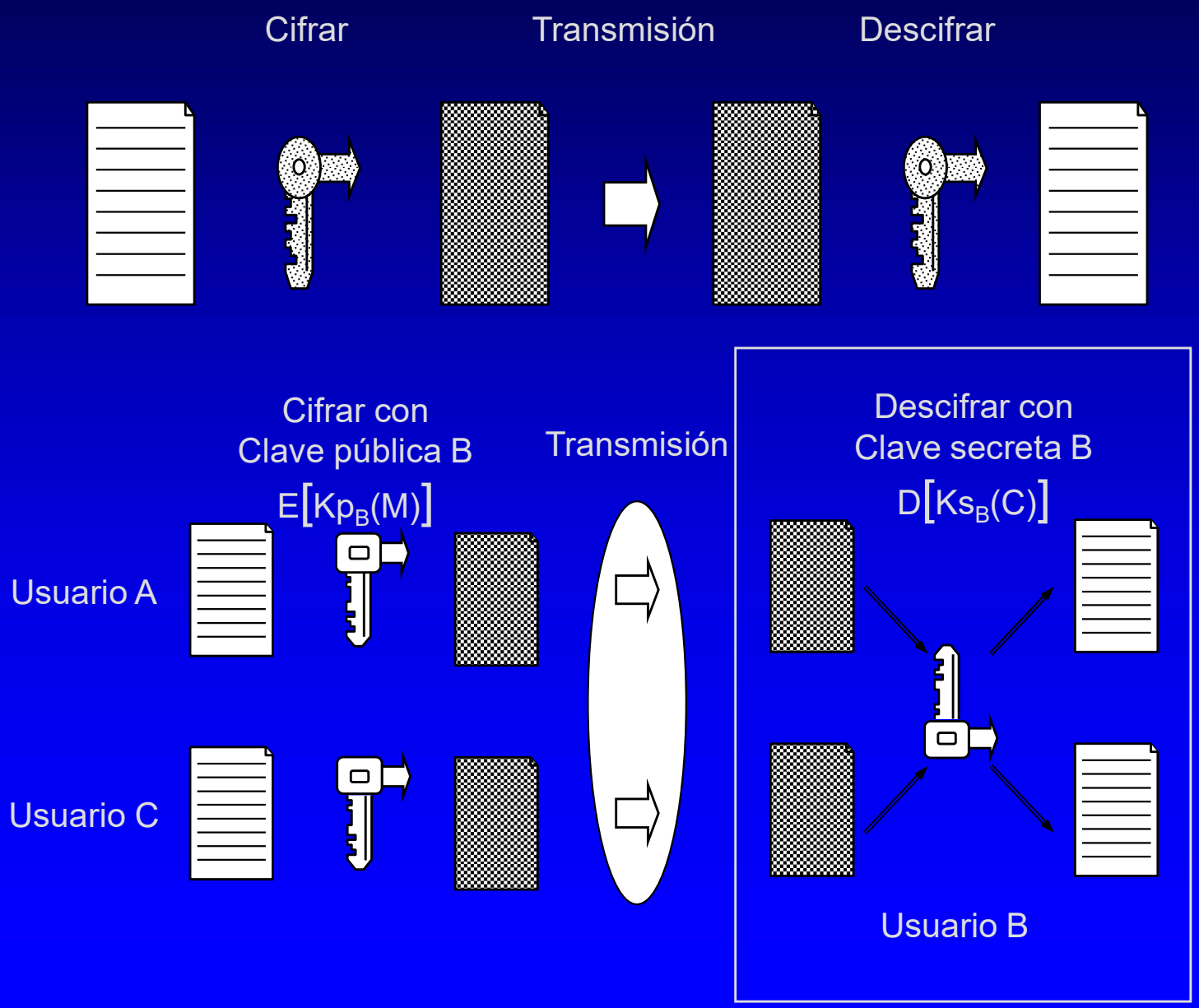
Herramientas criptograficas

- Algoritmos
- Protocolos
- Funciones de hash
- Certificados
- Autoridades de certificación
- Listas distribución

Clasificación algoritmos

- En función del tipo de claves
  - Algoritmos simétricos. Las claves utilizada para cifrar/descifrar son iguales o bien se pueden deducir fácilmente una a partir de la otra
  - Algoritmos asimétricos. Una de las claves es pública, mientras que la otra es secreta.

Cifrado simétrico vs. asimétrico



Clasificación

- En función del bloque
  - Cifrado en flujo. El tamaño del criptograma coincide con el del texto original
  - Cifrado en bloque. Se divide el texto en bloques de tamaño fijo. Si faltan caracteres, se debe añadir un relleno.

Clasificación

Órdenes de Magnitud

- Ejemplo. Potencia de cálculo: 10^{18} IPS
- Clave 128 bits:
  - Tiempo: **diez billones de años**
  - Memoria: **10^{38} bytes**
- Clave 256 bits:
  - Tiempo: **$4\ 10^{69}$ años**
  - Memoria: **10^{77} bytes**

Toda la materia de licúe: $10^{64}(2^{212})$años

Átomos en el Universo: $10^{77}(2^{256})$

Criptoanálisis: un primer contacto

- Tipología de los ataques:
  - Texto Cifrado Conocido
  - Parejas Texto Claro-Texto Cifrado
  - Texto Claro Escogido (Fijo y Adaptativo)
  - Texto Cifrado Escogido
  - "*La Clave o La Vida*"

Objetivos Criptoanalíticos

- Ruptura Total del Algoritmo
  - Puede descifrarse cualquier criptograma cifrado con cualquier clave
- Ruptura Total de la Clave
  - Se encuentra la clave de cifrado
- Ruptura Parcial
  - Se descifra un criptograma en concreto (sin hallar la clave)
- Información Parcial
  - Se obtiene algún conocimiento sobre el texto claro a partir del criptograma

Concepto de seguridad

- Seguridad incondicional. (Ej. Vernam)
  - Atacantes con tiempo y recursos ilimitados.
- Seguridad computacional. (Ej. RSA)
  - Atacantes con tiempo y recursos computacionales limitados.
- Seguridad probable. (Ej. DES, IDEA)
  - No se puede demostrar su integridad, pero el sistema aún no ha sido violado.
- Seguridad condicional. (Ej. Vignere)
  - El enemigo carece de medios para atacarlos

Nivel de Seguridad

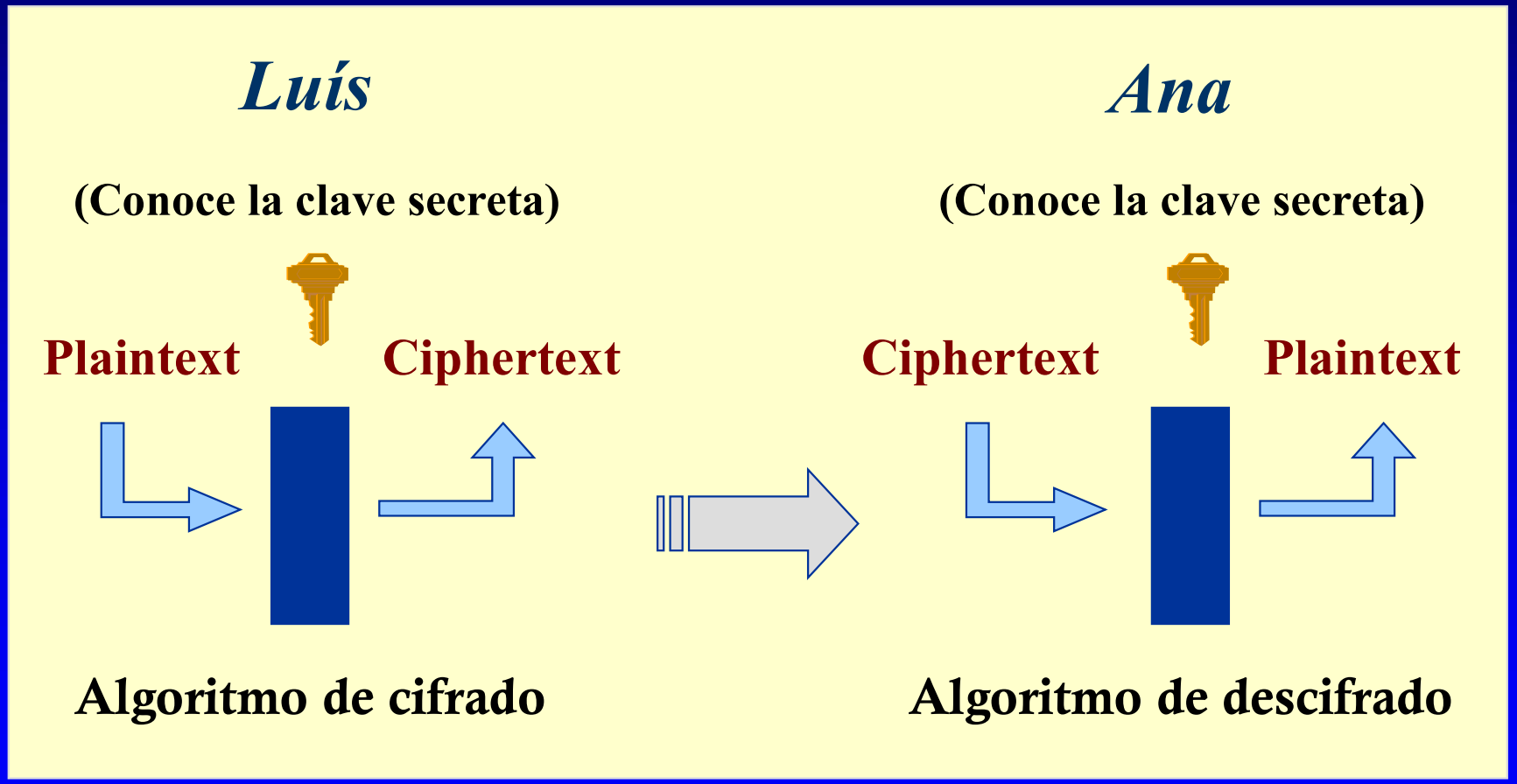
Conceptos fundamentales

- El *tiempo de cobertura* ha de ser superior al *periodo de validez* de la información
- El *coste de la seguridad* ha de ser inferior al *perjuicio de la violación*

Criptografía clásica

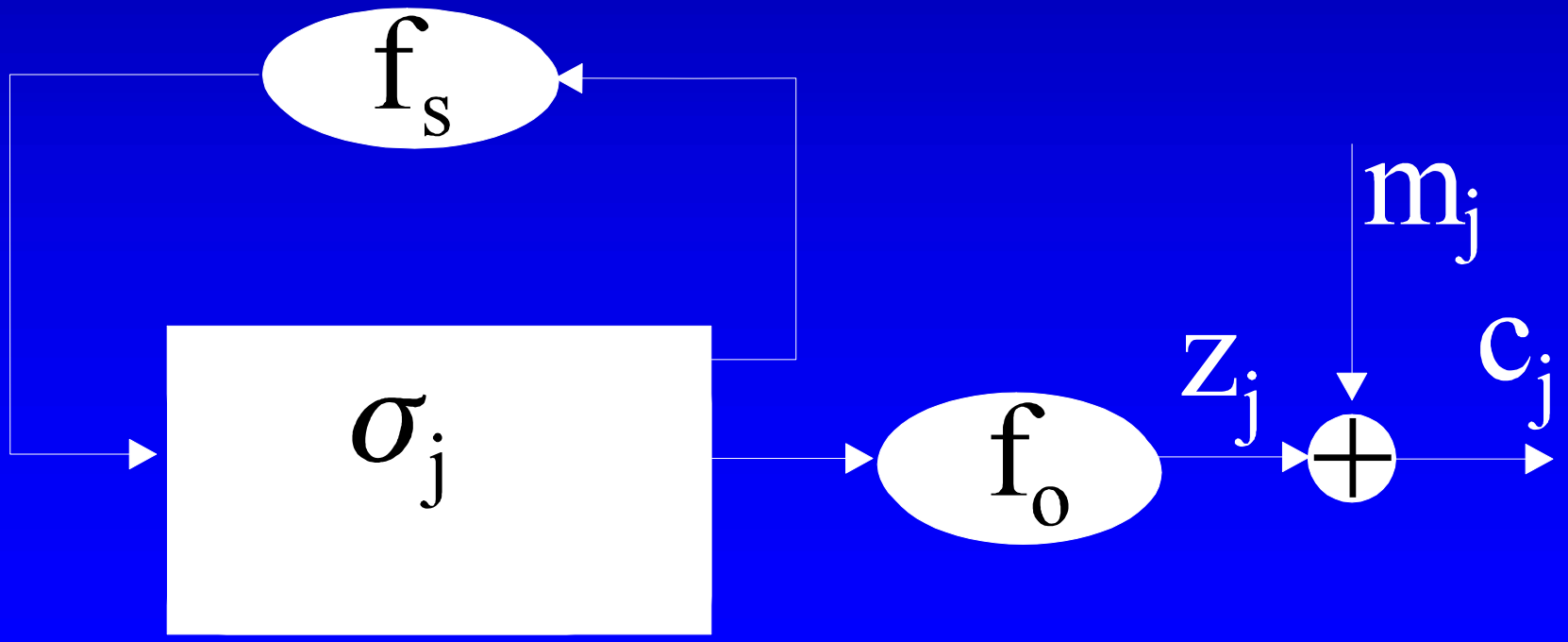
- Ejemplo Escítalo lacedemonia (s. V a.C)
- Principios de sustitución y transposició
- Sustitución monoalfabética
  - Cifrado de César
  - Método de Playfair
- Sustitución polialfabética
  - Cifrado de Vignère. La clave varía a lo largo de la sucesión. Roto en 1863 por el método Kasiski
  - Cifrado de Vernam. El único seguro incondicionalmente.

Cifrado simétrico

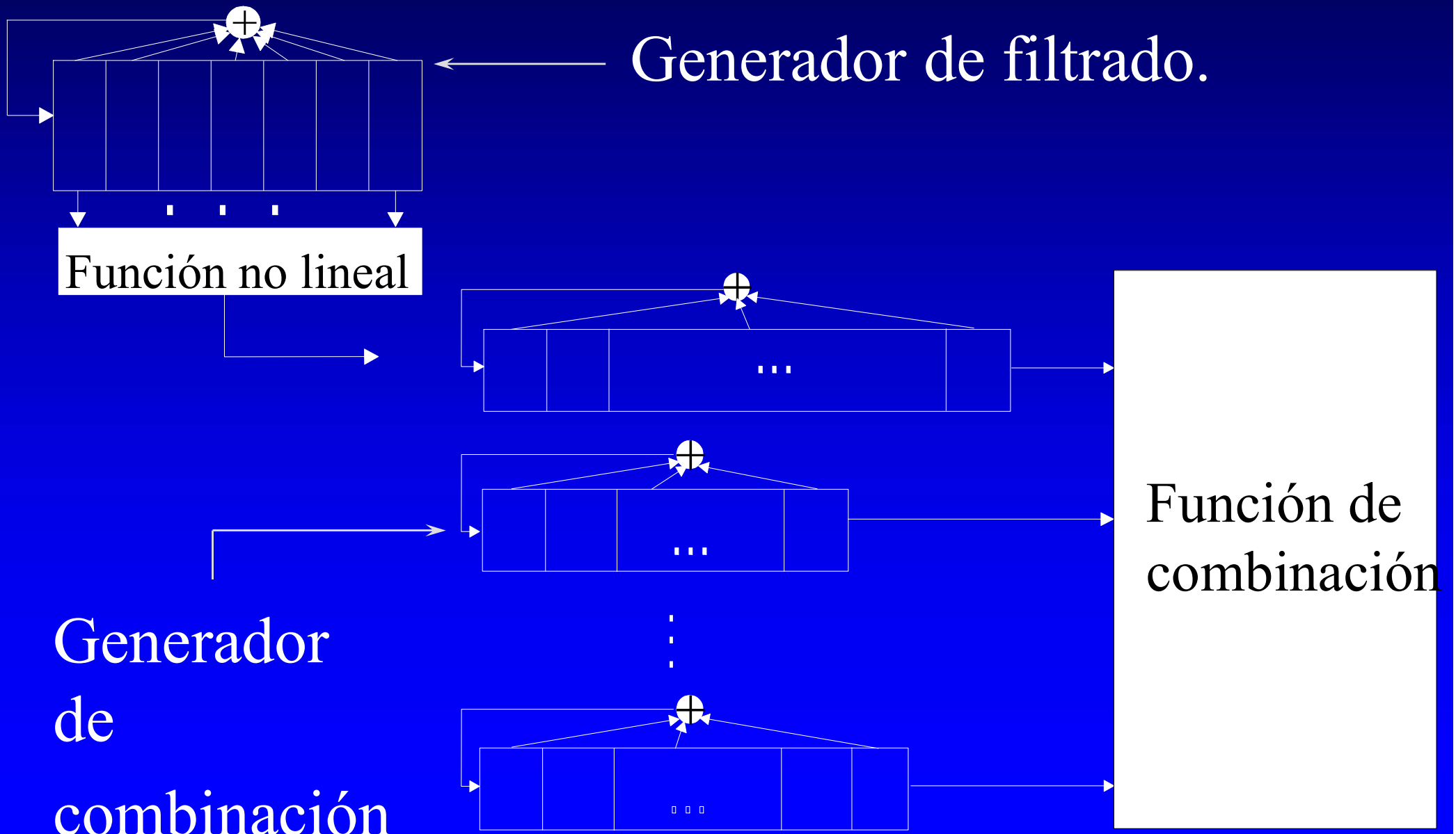


Cifrado en flujo. Estructura básica.

- ??$_j$: Estado (j).
- f_s: Función estado siguiente.
- f_o: Función salida.



Generadores de filtrado y combinación



Generadores Pseudoaleatorios

- Características Estadísticas Similares a las Fuentes Aleatorias.
- Secuencia fácilmente reproducible en Receptor (Determinismo)
- Realización Sencilla y Económica

Postulados de aleatoriedad de Golomb

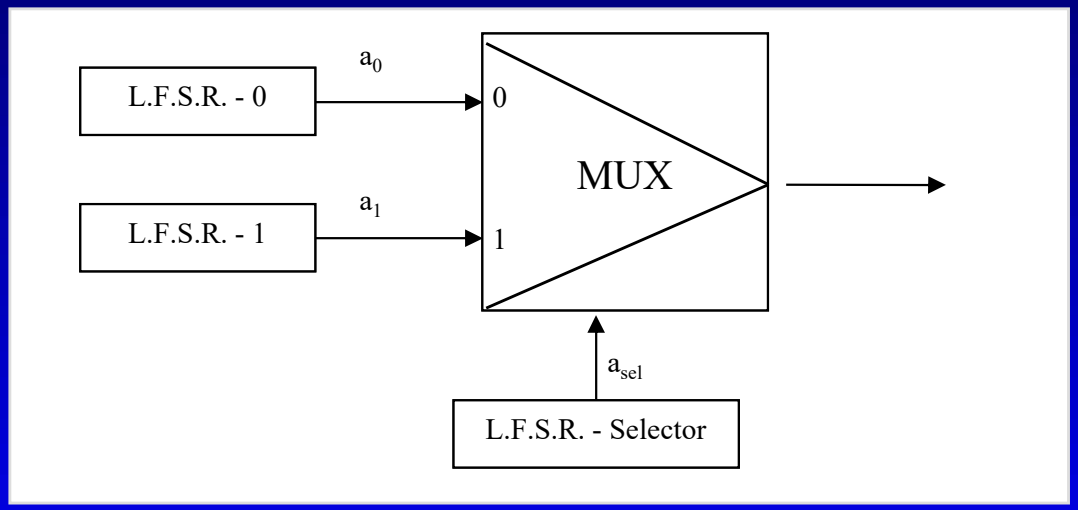
- Equidistribución
- Test de carreras
- Autocorrelación bivaluada

Registros de Desplazamiento Realimentados

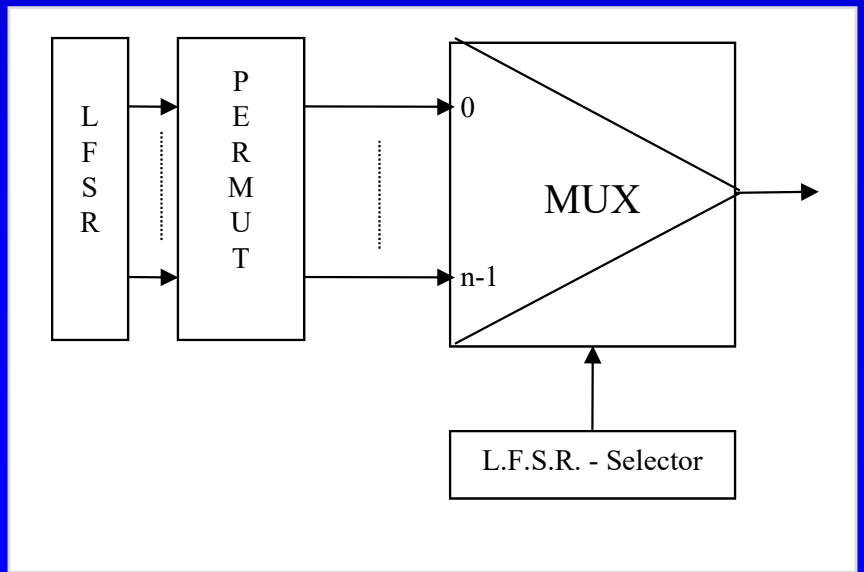
- LFSR Registros de Desplazamiento Linealmente Realimentados
- FCSR Registros de Desplazamiento Realimentados con Acarreo
- NLFSR Registros de Desplazamiento Realimentados de forma no lineal

Generadores

- Generador de Geffe

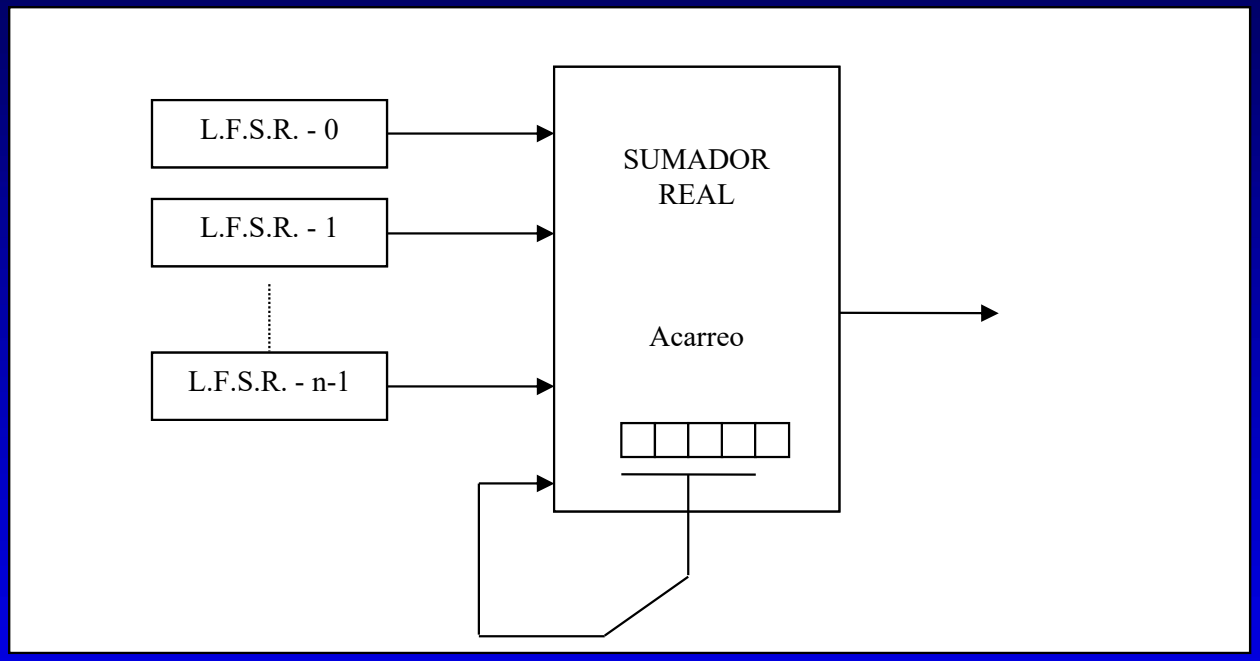


- Generador de Jennings

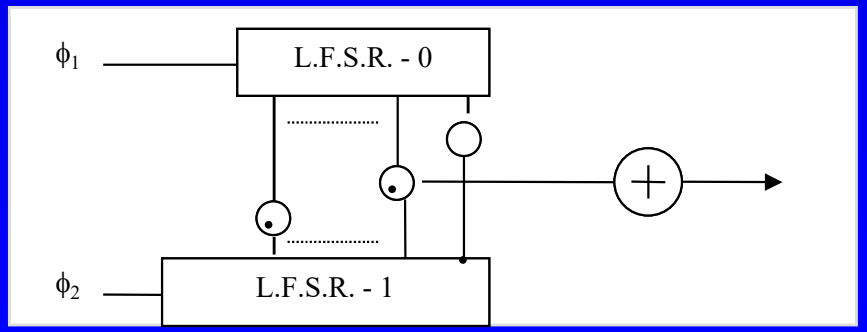


Generadores

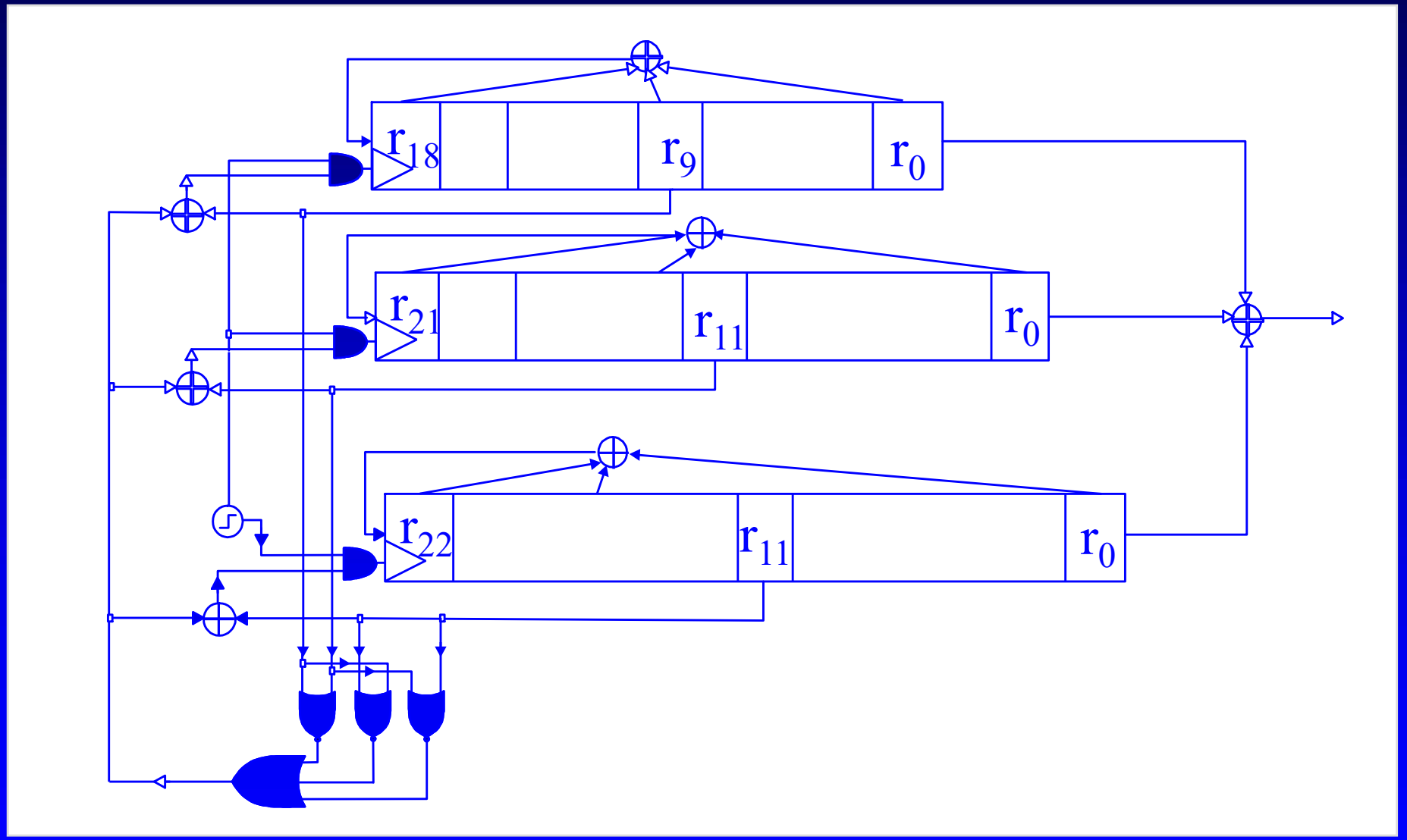
Sumador real de Rueppel



Multiclock Massey_Rueppel



Generadores (A5)



Cifrado en bloque simétrico

- Se cifra el mensaje original agrupando los símbolos en grupos (bloques) de dos o más elementos.
- Cada símbolo se cifra de manera dependiente de los otros que pertenecen al mismo bloque
- Dos bloques iguales se cifran de forma idéntica

Cifrado en bloque simétrico

- Transformación inicial
- Iteración r veces de una función criptográficamente débil.
- Transformación final
- Algoritmo de expansión de clave.

Transformación inicial

- Permite aleatorizar los datos de entrada para ocultar bloques. Carece de significación criptográfica si no depende de la clave (ej. DES)
- Dificultar algunos criptoanálisis (lineal o diferencial). (Ej. RC5, IDEA)

Iteraciones intermedias

- Funciones no lineales complicadas.
- Pueden ser unidireccionales (DES) o no (IDEA, RC5).
- Función no lineal
  - Una sola operación compleja
  - Sucesión de varias transformaciones simples.
- Se enlazan con $\oplus$ con datos procedentes de la transformación inicial o iteraciones precedentes.

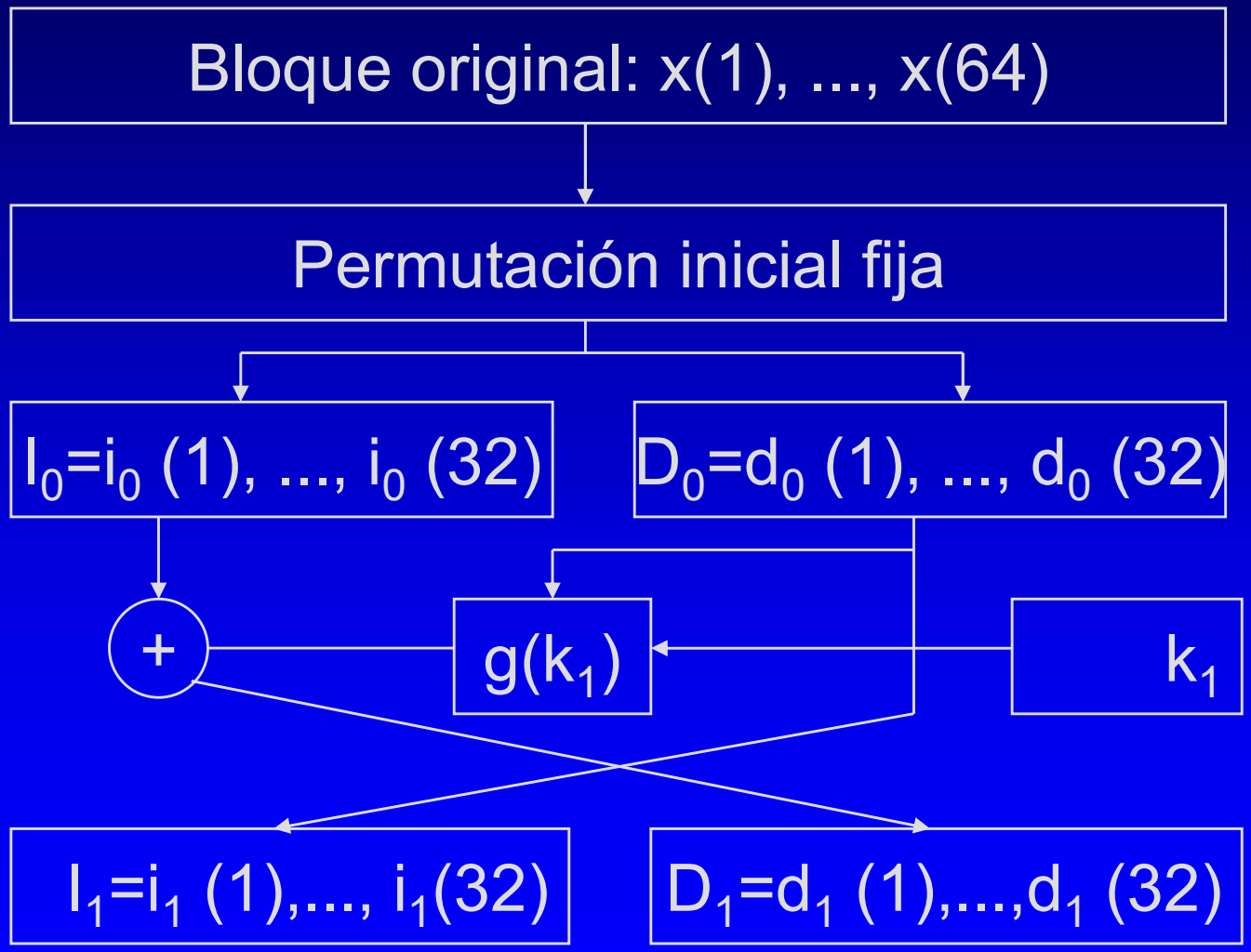
Iteraciones intermedias

- No han de formar grupo. No deben ser equivalentes al paso de una única iteración con una sola subclave.
- Transformación final. Permite que las operaciones de cifrado y descifrado sean simétricas

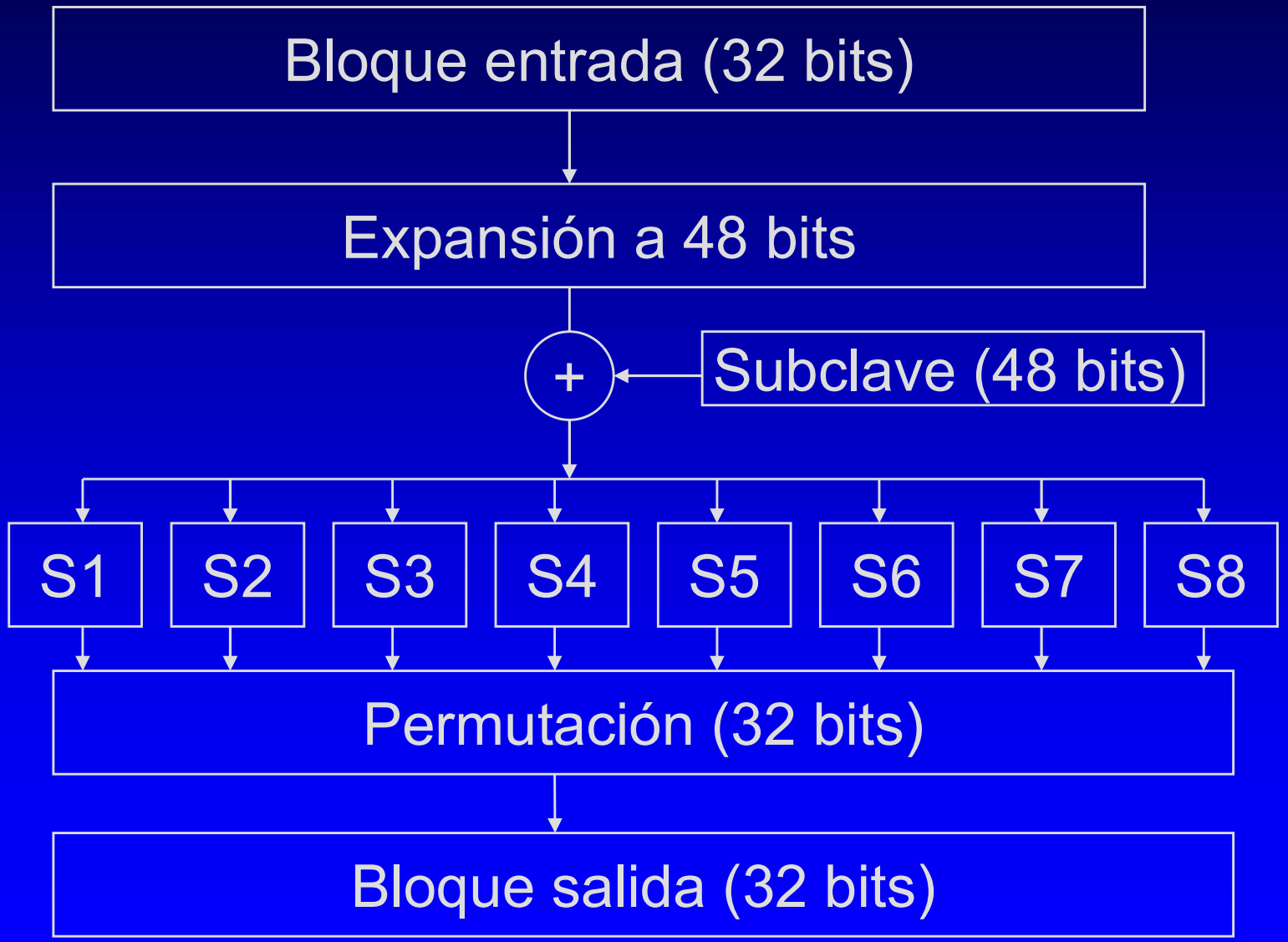
Cifrados de FEISTEL

- Dividen el bloque de datos en dos mitades, y en cada iteración se trabaja con una de las dos mitades.
- Ejemplos: LUCIFER, DES, LOKI, FEAL
- Algoritmo DES.
  - Claves 56 bits $\approx 7.2\ 10^6$ claves distintas
  - Ampliamente extendido.

Estructura del DES



Estructura transformación g



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**

Generación subclaves

- Permutación inicial fija

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

- División clave resultante en dos mitades
- Desplazamiento circular de cada mitad a la izquierda 1 o 2 bits

Vuelta	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
#bits	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Permutación

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

Regularidades del DES

- Complementariedad
  - $\overline{DES_k(m)} = DES_{\overline{k}}(\overline{m})$
  - Ejemplo:
  - m:0123456789abcdef
  - k: 133457799bbcdff1
  - $DES_k(m)$=85e813540f0ab405
  - $DES_{\overline{k}}(\overline{m})$=7a17ecabf0f54bfa

Regularidades del DES

- Claves débiles: $DES^2_k(m)=m$
  - 0101010101010101 fefefefefefefefe
  - e0e0e0e0f1f1f1f1 1f1f1f1f0e0e0e0e
- Claves semi-débiles $DES_{k1}[DES^2_{k2}(m)]=m$
  - Existen 6 parejas. Ejemplo:
  - 01fe01fe01fe01fe fe01fe01fe01fe01

Ataques al DES

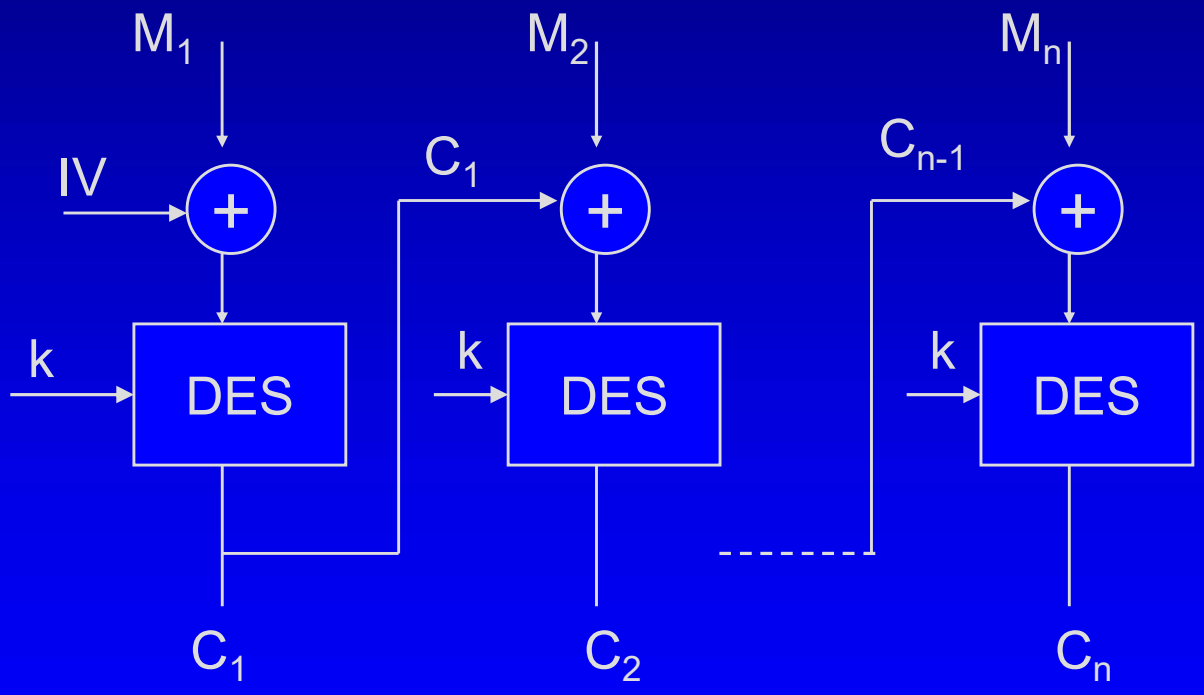
- Fuerza bruta.
- Ataque con texto en claro conocido
- Ataque con texto claro elegido
- Criptoanálisis diferencial
  - Comparaciones entre XOR de dos textos en claro elegidos y sus correspondientes criptogramas.
- Criptoanálisis lineal.
  - Obtención de un método lineal que represente la relación entre algunos bits del mensaje en claro, criptograma y clave

Modos de operación

- Cuatro modos básicos
  - ECB Electronic Code Book
  - CBC Cipher Block Chaining
  - CFB Cipher Feedback
  - OFB Output Feedback
- Otros métodos
  - PBC Plaintext Block Chaining
  - PFB Plaintext Feed Back

Ejemplo modos de funcionamiento

- Cifrado modo CBC



Modos de operación

MODO	CIFRADO	DESCIFRADO
ECB	C(n)=E[M(n),k]	M(n)=D[C(n),k]
CBC	C(n)=E[M(n)+C(n-1),k]	M(n)=C(n-1)+D[C(n),k]
CFB	C(n)=M(n)+E[C(n-1),k]	M(n)=C(n)+E[C(n-1),k]
PBC	C(n)=M(n-1)+E[M(n),k]	M(n)=D[M(n-1)+C(n),k]
PFB	C(n)=M(n)+E[M(n-1),k]	M(n)=C(n)+E[M(n-1),k]

Cifrados múltiples

- Cifrado de forma sucesiva un mismo bloque con diferentes claves DES. Aumenta el nivel de seguridad
- Dos claves ≠> clave equivalente 112 bits (ataque meet in the middle)
- Cifrado triple
  - Sin encadenamiento: EEE3, EEE2, EDE3, EDE2
  - Encadenamiento exterior EEE3-CBC, EDE3-CBC, EEE2-CBC, EDE2-CBC
  - Encadenamiento interior CBC-EEE3, CBC-EDE3, CBC-EEE2, CBC-EDE2

Otros algoritmos simétricos

- IDEA
  - Bloques 64 bits
  - Claves 128 bits
  - 8 vueltas .
  - Operaciones: XOR, sumas (mód 2^{16}) y multiplicaciones (mod $2^{16}+1$)
- RC2
- RC5
- BLOWFISH
- SAFER-64

Limitaciones métodos simétricos

- Secreto y autenticidad indisociables
- Verificación imposible sin un tercero
- Problema en la transmisión de claves y su crecimiento rápido para grupos con muchos usuarios
- En la mayoría, la seguridad es supuesta.

Cifrado en bloque clave pública

- Se definen dos algoritmos matemáticos E y D como operador asimétrico si y sólo si:
  - Son algoritmos públicos
  - Dependen de ciertas claves
  - Dado un mensaje M, verifican D(E(M))=M
  - Es difícil hallar D a partir de E
- Se basan en la utilización de ciertas funciones
  - Unidireccionales
  - Trampa

Cifrado en bloque clave pública

- Cada comunicante tiene dos claves: privada y pública (K_{sx}, K_{px}).
- Algoritmo público
- X calcula sus claves (K_{sx}, K_{px}) en tiempo polinómico.
- Y no puede calcular K_{sx} a partir de K_{px}.

Cifrado en bloque clave pública

Condiciones que deben satisfacerse.

Datos		Objetivo	Dificultad
(secreto)		(Ksx,Kpx)	fácil
(Kpx, M)		C	fácil
(Ksx,C)		M	fácil
Kpx		Ksx	difícil
(Kpx,C)		M	difícil

Definición protocolos asimétricos

	Cifrado	Descifrado
Confidencialidad	C=Ekpb(M)	M=DKsb(C)
Autenticidad y firma	C=Eksa(M)	M=DKpa(C)

Posibilidades criptografía asimétrica

- Confidencialidad y autenticidad pueden conseguirse por separado
- Firma digital posible sin intervención de un tercero
- Transmisión de claves públicas. Crecimiento lineal con número usuarios
- Seguridad computacional basada en la dificultad de funciones unidireccionales y trampa

Criptografía clave pública

- Función trampa
  - Cálculo de y=f(x) viable en tiempo polinómico
  - Cálculo de $x=f^{-1}(y)$ viable en tiempo polinómico si se conoce cierta información adicional
  - Cálculo de $x=f^{-1}(y)$ no viable en tiempo polinómico si se desconoce la información adicional
  - Cálculo información adicional no viable en tiempo polinómico

Métodos más extendidos

MÉTODO	BASADO EN ...
Diffie-Hellman	Logaritmo discreto
Massey-Omura	Logaritmo discreto
Elgamal	Logaritmo discreto
RSA	Factorización
Miller y otros	Logaritmo elíptico
Probabilisticos	Residuosidad cuadrática

Conceptos matemáticos básicos

- Congruencias
  - a≡b(c) o a=b (mod c) si a-b=kc (k entero)
    23 ≡3(5)
- Operaciones en aritmética modular
  - a op b mod(c) = [a mod c op b mod c] mod (c)

Conceptos matemáticos básicos

- Exponenciación modular

$$a^b \bmod(c) = \left[\prod_{i=1}^{b} a \bmod(c) \right] \bmod(c)$$

3^5 mod (7)

3*3 =9

9*9=81

81*3=243

243 mod (7)=5

3*3 mod(7)=2

2*2 mod(7)=4

4*3 mod(7)=5

Resultados intermedios acotados

Conceptos matemáticos básicos

- Función de Euler ϕ
- ϕ (N) Número de enteros i<N tales que mcd(i,N)=1
- Ej. ϕ (21) =12, dado que 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20 son co-primos con 21
- Propiedades
  - Si p es primo, ϕ (p) =p-1
  - Si n=p.q, (p, q primos), ϕ (n) = (p-1) (q-1)

Conceptos matemáticos básicos

- Teorema de Fermat: Sean a, b co-primos

$$a^{\phi(b)} = 1 \bmod (b)$$

- Inversos modulares
  - Dos enteros a y b, co-primos con c, son inversos módulo c cuando a*b mod(c)=1
  - Ej: 5 y 7 mod 17
- Propiedad
  - Todos los enteros tienen inverso módulo cualquier primo

Conceptos matemáticos básicos

- Cuando el módulo utilizado es primo, el conjunto de restos constituye el cuerpo de Galois (GF)
- Dado GF(p), un elemento e se denomina primitivo si $\{e^n\}$ describe a GF(p). Si p=7, 3 es un elemento primitivo

$3^0 \equiv 1$ $\quad 3^1 \equiv 3$ $\quad 3^2 \equiv 2$

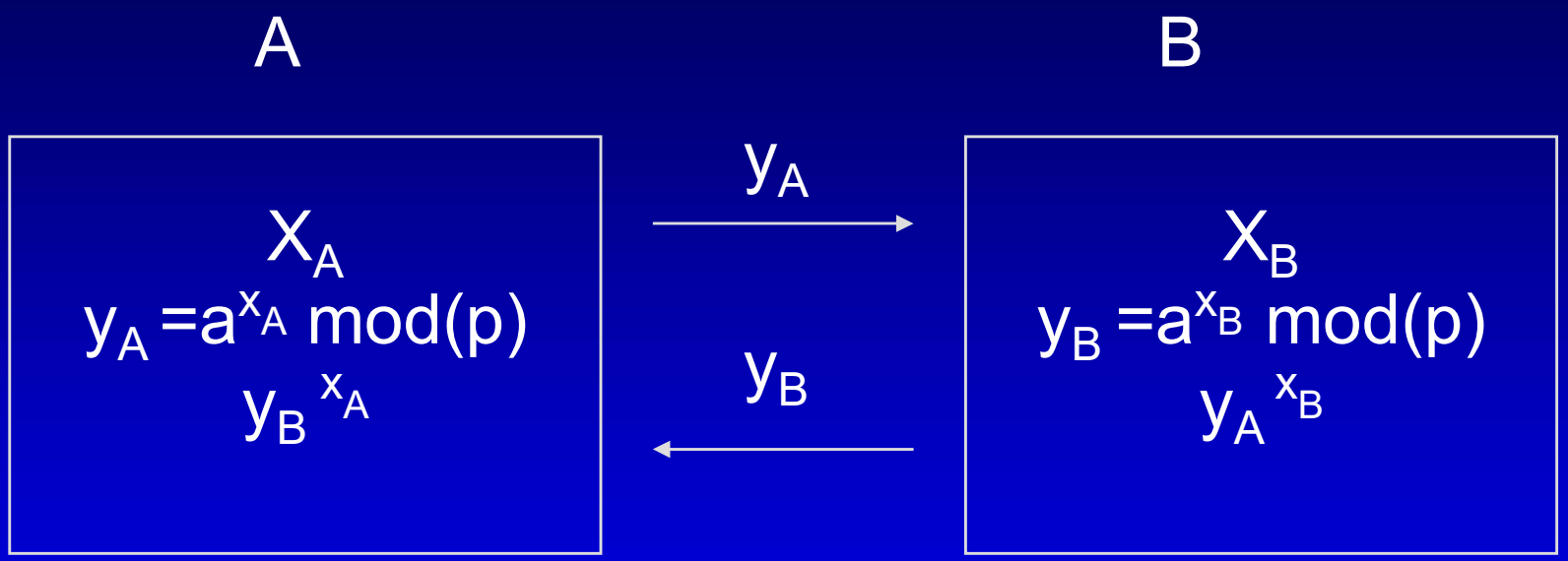
$3^3 \equiv 6$ $\quad 3^4 \equiv 4$ $\quad 3^5 \equiv 5$

- Puede definirse el logaritmo discreto en GF(p)

Método de Diffie-Hellman

- Método público de distribución de claves basado en la dificultad de obtener logaritmos discretos
- Construcción
  - Sea p un primo y a un elemento primitivo público
  - Cada comunicante realiza las siguientes operaciones
  - Elige x aleatorio y secreto
  - Calcula $y=a^x \bmod (p)$

Método de Diffie-Hellman



$$y_B{}^{x_A} = (a^{x_B})^{x_A} \bmod(p) \quad = \quad y_A{}^{x_B} = (a^{x_A})^{x_B} \bmod(p)$$

$$K_{AB} \quad = \quad K_{BA}$$

Método de Diffie-Hellman

- Fundamentos:
  - Se basa en que un intruso tendrá que calcular un logaritmo discreto para hallar K_{AB}, a partir de y_A e y_B
- Consideraciones adicionales
  - p ha de ser grande (> 200 cifras decimales)
  - Utilización de primos de Gordon, ya que dificultan el cálculo de logaritmos discretos

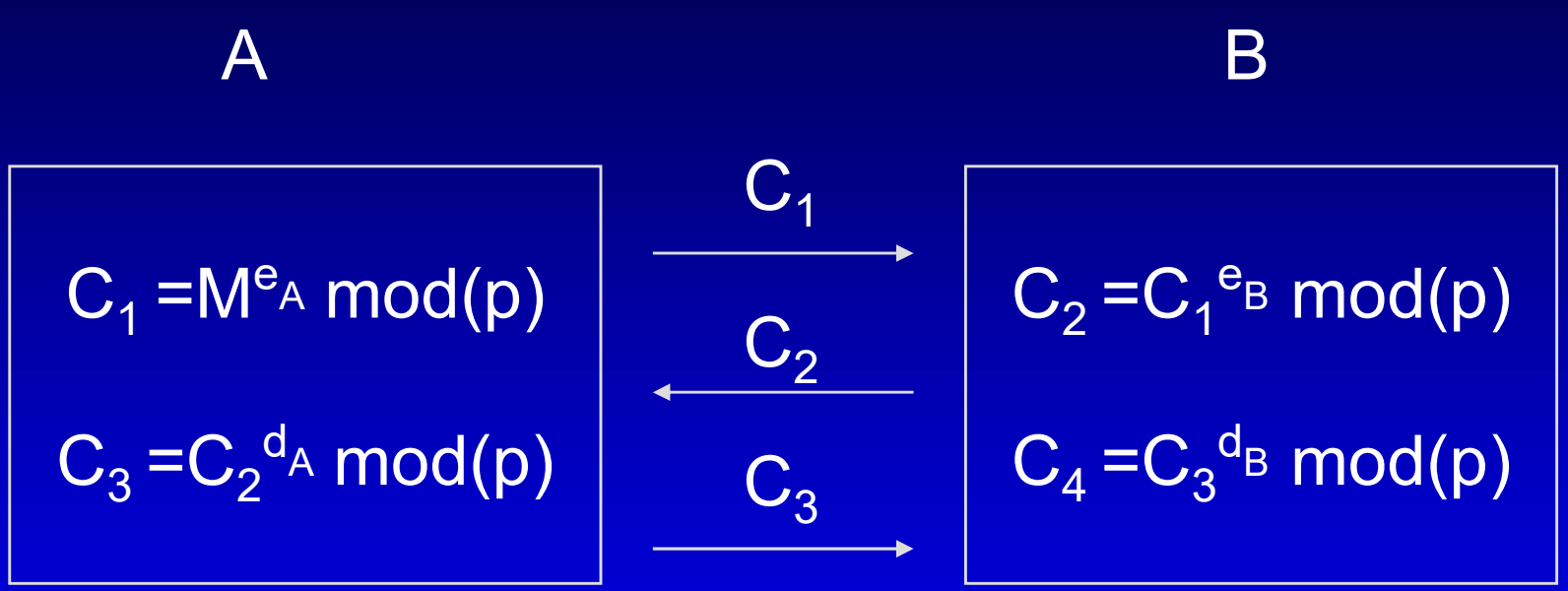
Método de Diffie-Hellman

- Consideraciones adicionales
  - El método posibilita un ataque activo por parte de un criptoanalista
  - No permite ni confidencialidad, autenticidad ni firma
  - La clave viene fijada
- Áreas de aplicación
  - Solución sencilla para transmisión de claves de sistemas simétricos cuando no existe conocimiento previo entre comunicantes

Método de Massey-Omura

- Método público de cifrado basado en la dificultad de obtener logaritmos discretos, muy similar al anterior
- Construcción
  - Sea p un primo público
  - Cada comunicante realiza las siguientes operaciones
  - Elige e aleatorio, secreto y co-primo con p-1
  - Calcula d, inverso de e módulo p-1

Método de Massey-Omura



$$C_4 = C_3^{d_B} \bmod(p) = (M^{e_A})^{e_B})^{d_A})^{d_B} \bmod(p) = M$$

Método de Massey-Omura

- Fundamentos:
  - Se basa en que un intruso tendrá que calcular un logaritmo discreto para hallar M, a partir de C_1 sin conocer base ni exponente
  - Consideraciones adicionales
    - p ha de ser grande (> 200 cifras decimales)
    - Utilización de primos de Gordon,
    - El método posibilita un ataque activo
    - Permite confidencialidad, pero no autenticidad ni firma

Método de Massey-Omura

- Ejemplo

  p=17 e_A=13 d_A=5

  M=10 e_B=3 d_B=11

  C_1=11 C_2=5 C_3=14 C_4=10

- Áreas de aplicación
  - Solución sencilla para transmisión de claves de sistemas simétricos cuando no existe conocimiento previo entre comunicantes

Método de Elgamal

- Método público de cifrado basado en la dificultad de obtener logaritmos discretos.
- Construcción
  - Sea p un primo y e un elemento primitivo, públicos
  - Cada comunicante realiza las siguientes operaciones
  - Elige x aleatorio y secreto.
  - Calcula $y = e^x \bmod p$ y lo publica

Método de Elgamal.

Confidencialidad

A

x_A, $y_A = e^{x_A} \bmod(p)$

$K = y_B{}^{g} \bmod(p)$

$C_1 = e^{g} \bmod(p)$

$C_2 = K + M \bmod(p)$

y_B

C_1, C_2

B

x_B, $y_B = e^{x_B} \bmod(p)$

$C_1{}^{x_B} \bmod(p) = K$

$[C_2 - K] \bmod(p) = M$

$$C_1{}^{x_B} \bmod(p) = (e^{g})^{x_B} \bmod(p) = K$$

Método de Elgamal.

Autenticidad/Firma

A

¿R, S?

$e^m = y_A{}^R R^S \bmod(p)$

$R = e^g \bmod(p)$

$e^m = e^{R*X_A} * e^{g*S} \bmod(p)$

$m = x_A * R + g * S \bmod (p-1)$

R, S →

B

Comprueba la igualdad

?

$e^m = y_A{}^R R^S \bmod(p)$

Si es cierta, (R,S) es la autentificación o la firma del mensaje m

Método de Elgalmal

- Fundamentos:
  - Se basa en que un intruso tendrá que calcular un logaritmo discreto para hallar g o x_A
  - Consideraciones adicionales
    - p ha de ser grande (> 200 cifras decimales)
    - Utilización de primos de Gordon,
    - Permite confidencialidad, autenticidad y firma
    - Seguridad parametrizable (según p)
    - Tamaño cifrado es doble que el del mensaje
    - Cada usuario puede tener su propio entorno

Método de Elgamal

- Ejemplo

  p=17, e=3 $\quad x_A=4 \rightarrow y_A=3^4=13 \pmod{17}$

  $x_B=7 \rightarrow y_B=3^7=11 \pmod{17}$

  Confidencialidad

  M=14, g=9, $K=11^9=6$, $C_1=3^9=14$, $C_2=14+6=3$

  $C_1^7=14^7=6$, M=3-6=14

  Autenticidad/Firma

  m=10, g=11, $\rightarrow R=3^{11}=7$

  $10=4*7+11*S \pmod{16} \rightarrow S=10$

  $3^{10}=8$, $13^7 * 7^{10}=4*2=8$ √

Método RSA

- Método público de cifrado basado en la dificultad de factorizar números grandes.
- Construcción
  - Cada comunicante realiza las siguientes operaciones
  - Elige p, q primos aleatorios y secretos
  - Calcula n=p.q y lo publica
  - Elige e aleatorio y co-primo con $\phi(n)$ y lo publica
  - Calcula d, inverso de e mod $\phi(n)$, secreto

Método R.S.A.

Confidencialidad

A

d_A, p_A, q_A, e_A, n_A

$C=M^{e_B} \bmod(n_B)$

C →

B

d_B, p_B, q_B, e_B, n_B

$C^{d_B} \bmod(n_B)=M$

$$C^{d_B} \bmod(n_B)=(M^{e_B})^{d_B} \bmod(n_B) = M^{k.\phi(n)+1} \bmod(n) =M$$

Método R.S.A.

Autenticidad/Firma

A

d_A, p_A, q_A, e_A, n_A

$C=M^{d_A} \bmod(n_A)$

C →

B

d_B, p_B, q_B, e_B, n_B

$C^{e_A} \bmod(n_A)=M$

$$C^{e_A} \bmod(n_A)=(M^{d_A})^{e_A} \bmod(n_A) = M^{k.\phi(n)+1} \bmod(n_A) =M$$

Método de R.S.A.

- Fundamentos:
  - Se basa en la dificultad de obtener d a partir de e, dado que su cálculo se basa en el conocimiento de $\phi(n)$, requiriéndose la factorización de n si se desconocen p y q
  - Consideraciones adicionales
    - p y q han de ser grande (> 100 cifras decimales)
    - Permite confidencialidad, autenticidad y firma
    - Tamaño cifrado es el del mensaje

Método de R.S.A.

- Ejemplo: B transmite a A un M cifrado.

  $p_A = 61$, $q_A = 47$ $\rightarrow n_A = p_A * q_A = 2867$

  $e_A = 247$ $\rightarrow d_A = 247^{-1} \bmod 2760 = 2503$

  $\phi(n_A) = 2760$

  Confidencialidad

  M= 1575

  $C = 1575^{247} = 2085 \pmod{2867}$

  $C^{2503} = 2085^{2503} = 1575 \pmod{2867} = M$

  Autenticidad/Firma: A se autentifica ante B.

  Idéntica, permutando e y d

Algoritmo RSA

- Elección p, q
- a) p, q no deben ser muy próximos

  Supongamos p>q y próximos

$$\frac{(p+q)^2}{4} - n = \frac{(p-q)^2}{4}$$

  Se prueban valores x enteros> $n^{0.5}$ hasta que uno de ellos verifique que x^2-n sea cuadrado perfecto

- b) (p-1) y (q-1) deben contener factores primos grandes
- c) mcd (p-1, q-1) debe ser pequeño

Función de hash: definición

- Función de hash
  - Una función de hash(H) transforma un mensaje de entrada (M) de longitud variable y lo convierte en un string de salida de longitud fija, H(M).

Requerimientos básicos

- Los requerimientos básicos para una función criptográfica de hash:
  - La entrada puede tener cualquier longitud
  - La salida tiene una longitud fija
  - H(x) es relativamente fácil de calcular para cualquier x dado
  - H(x) es en un sentido: dado un valor de hash (h) es computacionalmente imposible encontrar algún mensaje de entrada x tal que H(x) = h.
  - H(x) no presenta colisiones: es computacionalmente imposible encontrar dos mensajes cualesquiera x e y tales que H(X) =H(Y).

Función de compresión

$$h_i = f(M_i, h_{i-1})$$

Funciones de hash:

- Las funciones de hash más utilizadas
  - MD2
  - MD4
  - MD5
  - SHA

Funciones de hash (I)

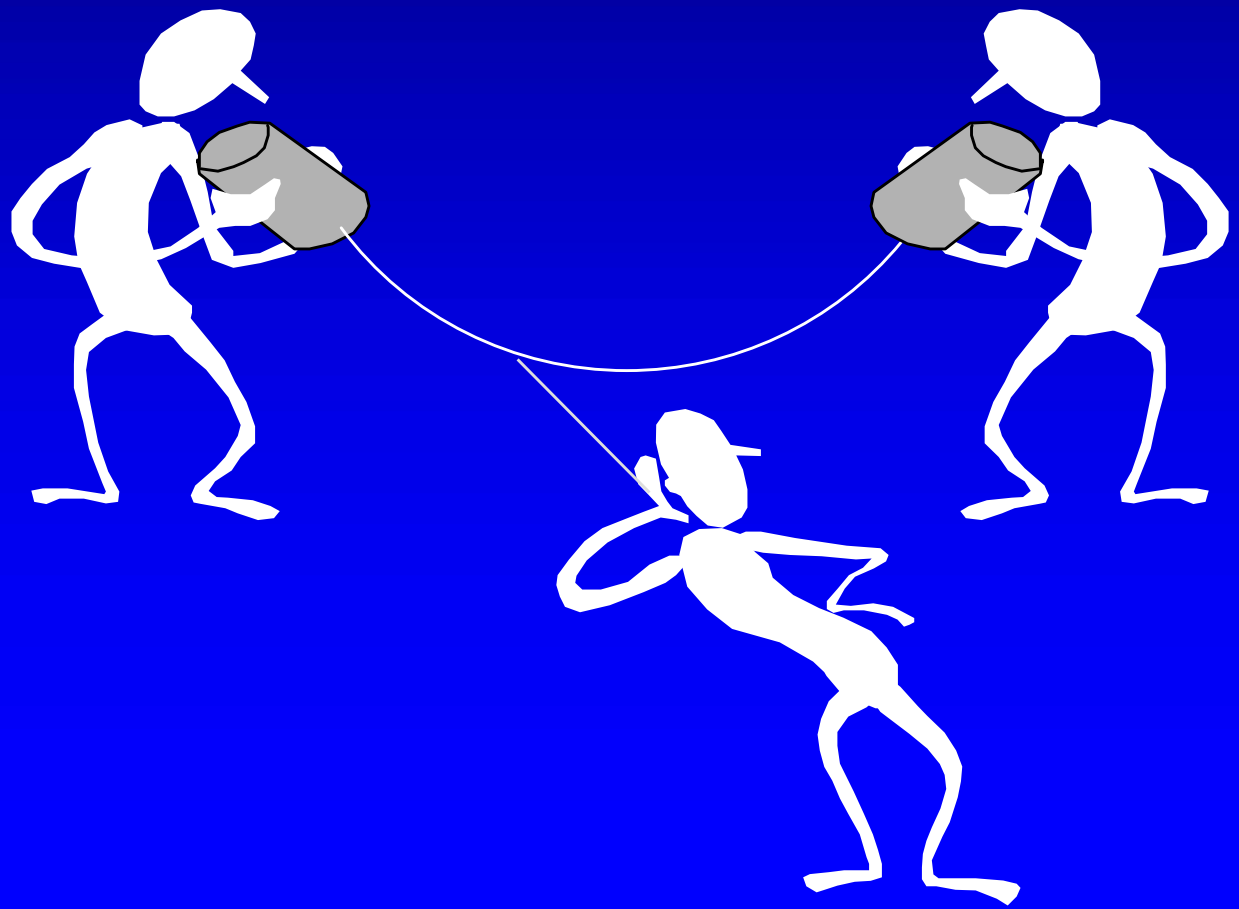
- Funcionalidad
  - Firma digital de documentos
  - Verificación de claves públicas

Necesidad de seguridad

- Continuo desarrollo y expansión de redes
- Uso de equipamiento informático en tratamiento y transmisión de información crítica
- Mayor facilidad para realizar con éxito un ataque

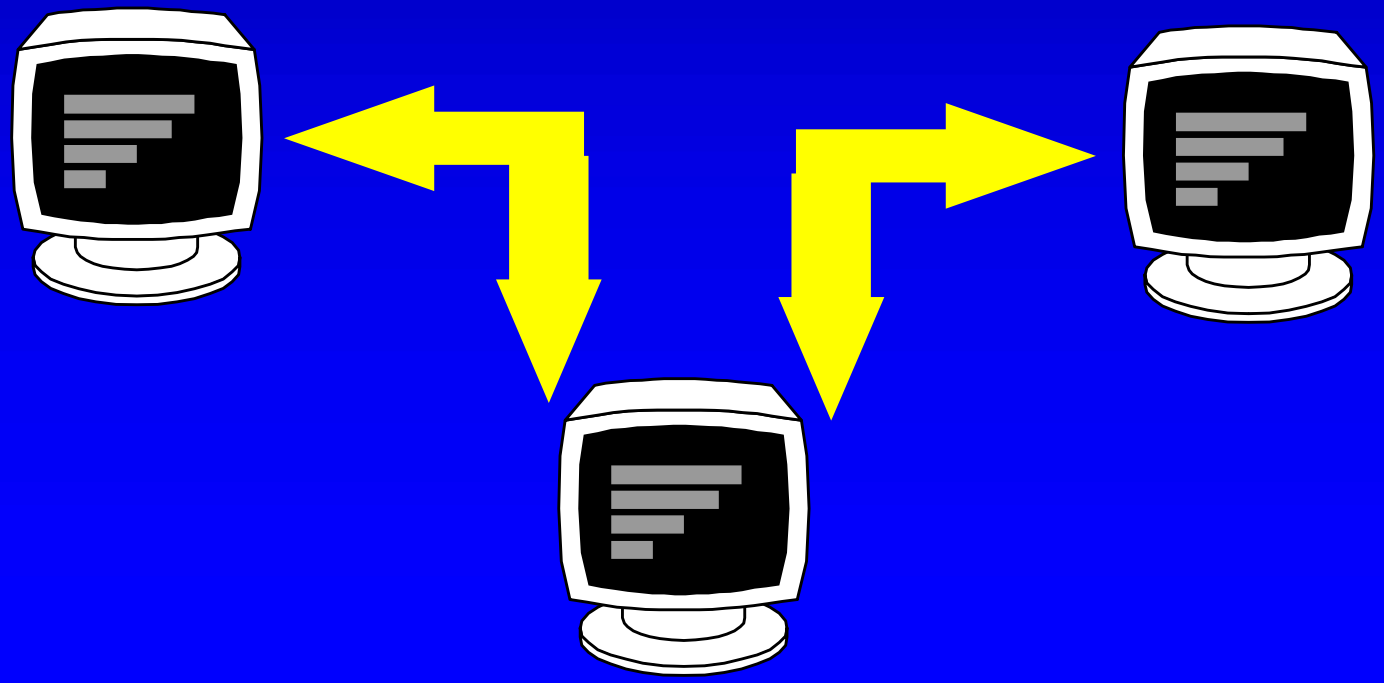
Ataque pasivo

- El enemigo se limita a la escucha, sin modificar el contenido.
- Confidencialidad



Ataque activo

- El enemigo puede emitir suplantando a alguno de los comunicantes, o capturar, modificar, eliminar, repetir, retrasar o reordenar los mensajes.
- Confidencialidad, integridad y autenticidad



Ataque activo. Clasificación

- Suplantación de identidad
- Reactuación
- Modificación
- Degradación fraudulenta del servicio
- Repudio
- Caballos de troya
- Encaminamiento incorrecto

Mecanismos de seguridad

- Intercambio de autentificación
- Cifrado
- Integridad de datos
- Firma digital
- Tráfico de relleno
- Control de encaminamiento

Servicios de Seguridad

SERVICIOS DE SEGURIDAD	EJEMPLOS DE LA VIDA COTIDIANA
AUTENTICACIÓN	DNI
CONTROL DE ACCESO	LLAVES Y CERROJOS
CONFIDENCIALIDAD	TINTA INVISIBLE
INTEGRIDAD	TINTA INDELEBLE
NO REPUDIO	FIRMA NOTORIZADA.

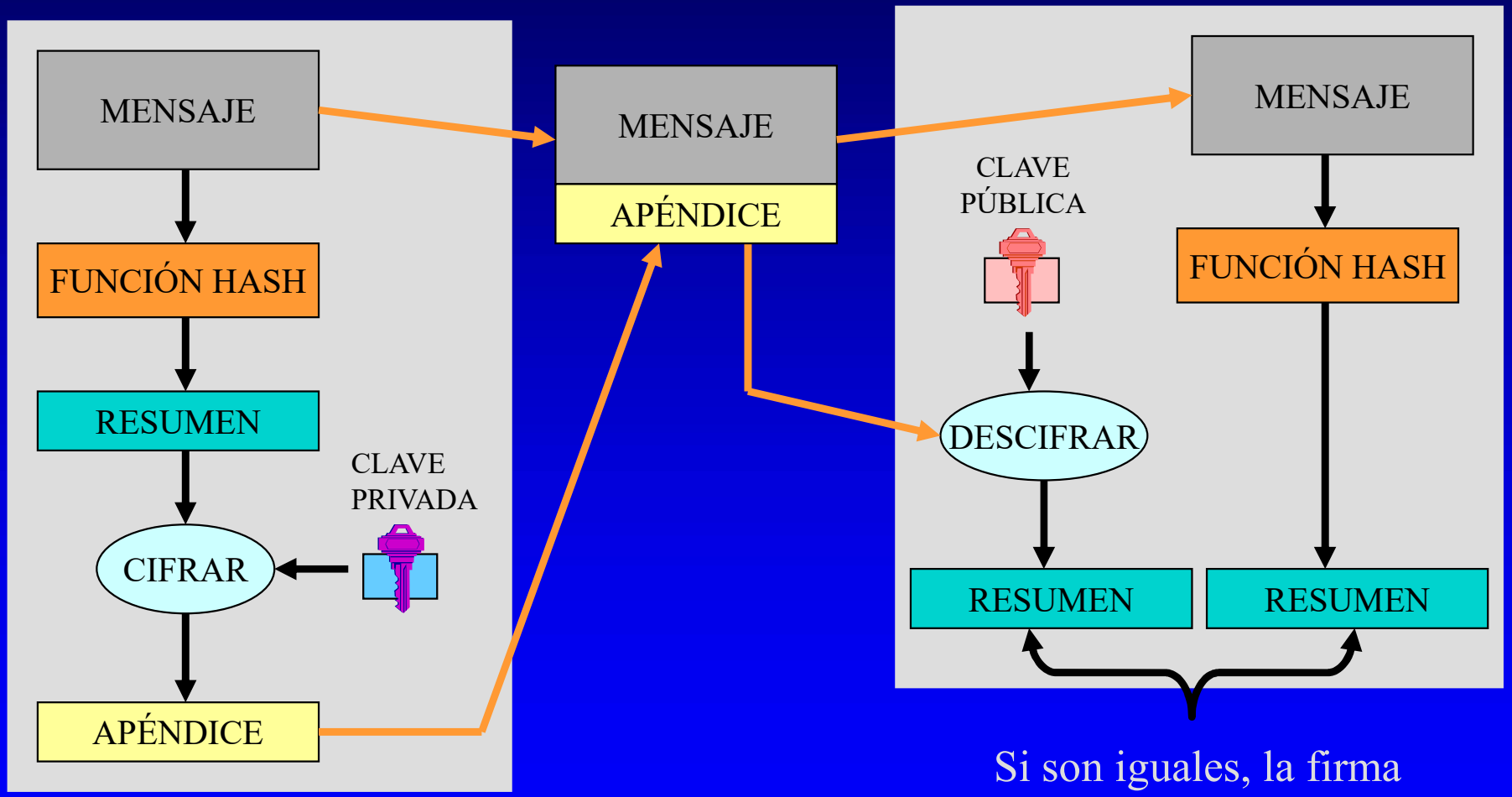
Firma digital (I)

- Características de las firmas digitales:
  - La firma debe ser auténtica
  - La firma no puede ser falsificada
  - La firma no puede ser reutilizada
  - El documento firmado no puede ser alterado
  - La firma no puede ser repudiada por el firmante

Firma digital (II)

- Protocolo de firma digital implementado con funciones de hash
  - A calcula el hash de un documento
  - A cifra el hash con su clave privada
  - A envia el documento y el hash cifrado a B
  - B calcula el hash del documento. B descifra el hash firmado, utilizando la clave pública de A. Si ambos coinciden, la firma es válida.

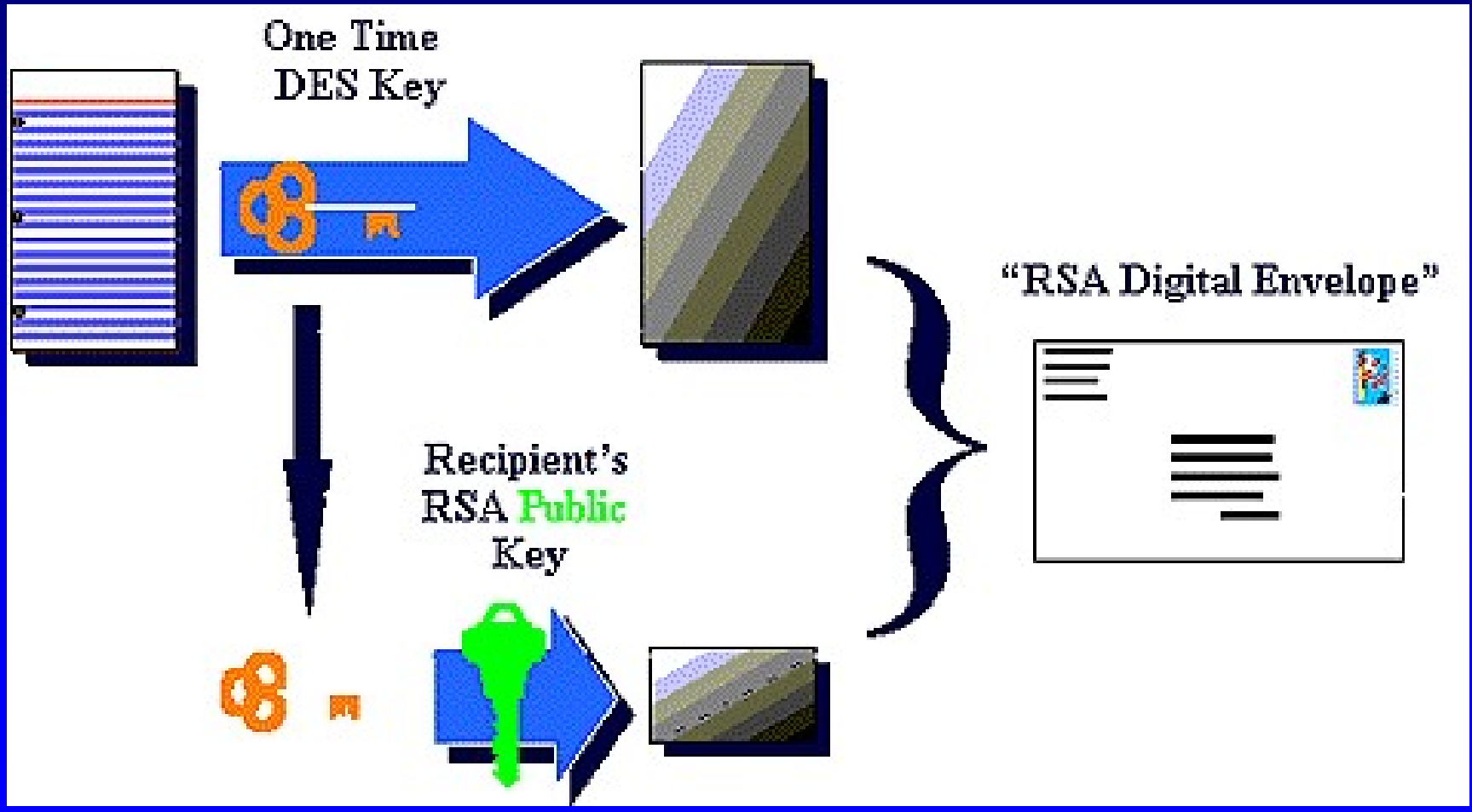
Firma digital (II)



Firma digital (III)

- Funciones de la firma digital:
  - Integridad
  - Autentificación
  - No repudiación

Sobre digital



Firma digital y timestamps

- Documentos tienen validez temporal
- Ej. Cheque bancario
  - El receptor puede duplicar el mensaje y cobrarlo n veces
  - Solución: Añadir marcas temporales al mensaje
  - El banco almacena esa marca temporal en su base de datos

No repudio de firmas

- Un usuario puede publicar anónimamente su clave privada, e indicar que no ha firmado un documento => Repudio
- Timestamps limitan este problema
- Solución general: uso de terceras partes

Verificación de claves públicas

- La clave pública será verificada por el receptor mediante la verificación del valor de hash de la clave pública(o fingerprint).
- Ejemplo: (Servidor de claves PGP Red Iris)

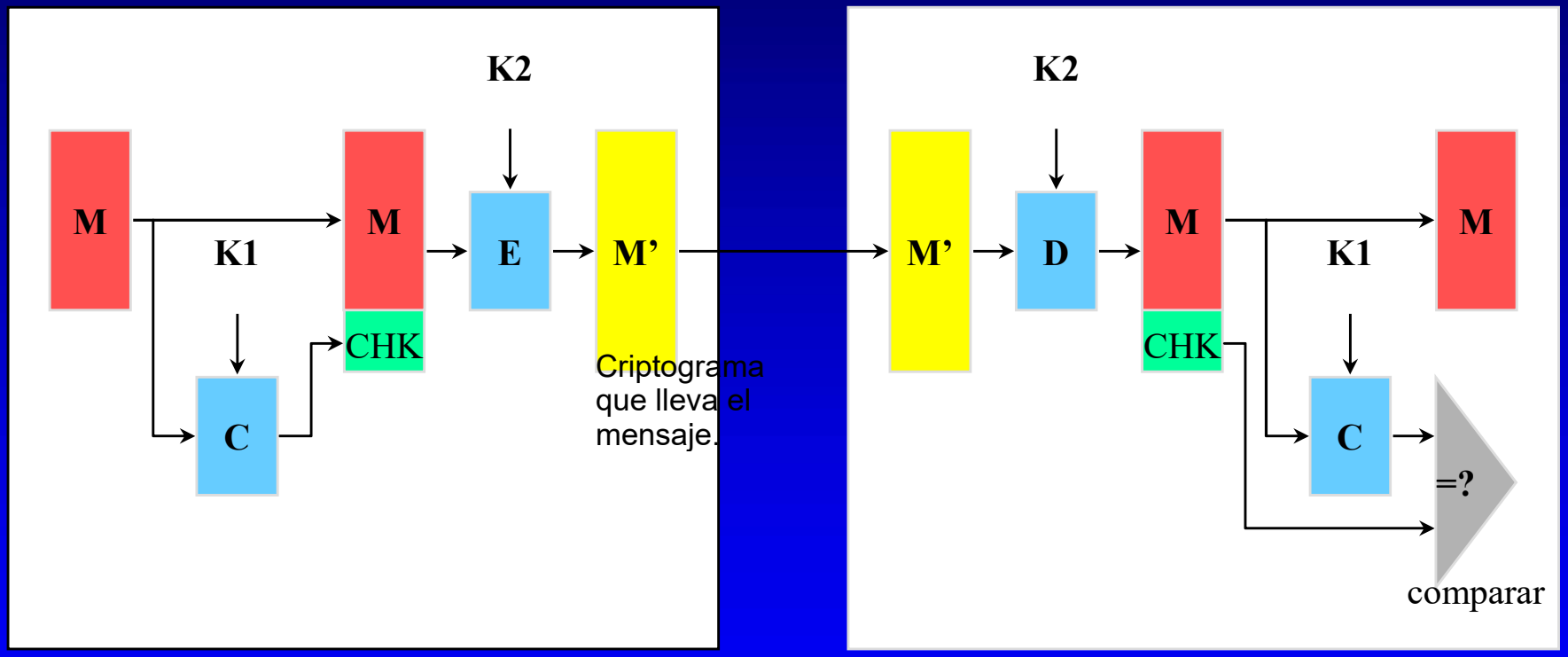
Red iris: red de académicos

Type bits/keyID Date User IDpub

1024/998F00E1 1997/12/12 Miquel Soriano<soriano@mat.upc.es>

Firma digital -> Key fingerprint = 56 70 7B 11 9E 6D A1 EF 75 EA D3 78 59 EA 5C AA 99 8F 00 E1

Autentificación y Confidencialidad



Autentificación

- Passwords.
  - Viajan en claro (en general)
  - ‘Fáciles de adivinar’
  - Autentificación en un sentido
  - Ataques de diccionario
- Información biométrica
  - Muestras de voz
  - Huella dactilar
  - Firmas manuales
  - Hardware costoso
  - Fiabilidad

Autentificación

- Técnicas criptográficas
  - El usuario prueba su identidad demostrando su capacidad de cifrar/descifrar un mensaje
  - Retos deben variar cada vez que se ejecuta el protocolo
  - Técnicas de retos
    - Time-stamps
    - Contadores
    - Nonces

Time-stamps

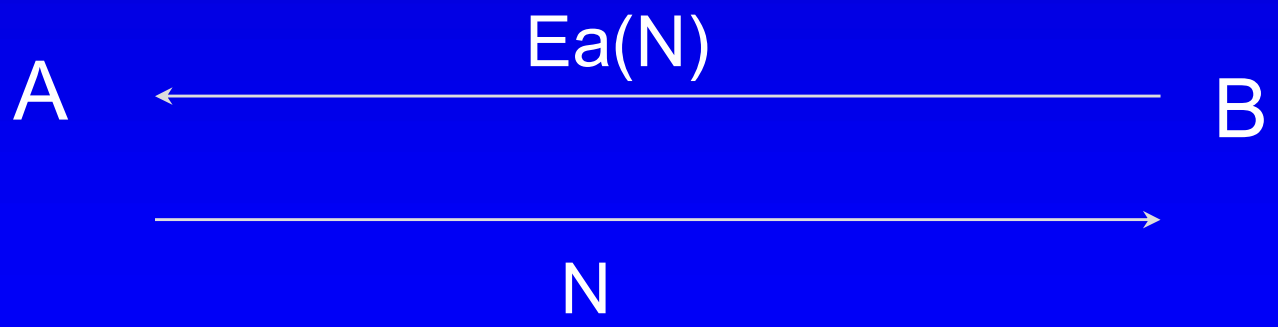
- La entidad a ser autentificada (A) cifra el contenido de su reloj y lo envía (a B)
- B descifra el mensaje y compara con su reloj
- Problemas
  - Sincronismo de relojes
  - Práctica: Tolerancia, ventana temporal
  - Enemigo puede aprovechar dicha tolerancia

Contadores

- A y B deben mantener contadores sincronizados
- Contadores largos (evitar ataques por reactuación)
- Conflicto cuando ambas partes desean iniciar el proceso simultáneamente

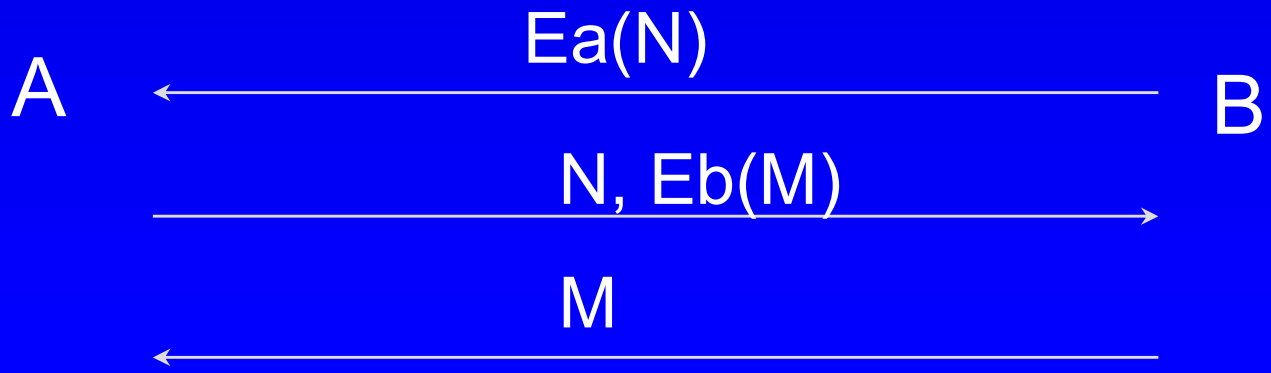
Nonces

- Números generados aleatoriamente
- Coste:incremento de tráfico en la red
- Protocolos de autentificación usando nonces
  - Autentificación unidireccional de A

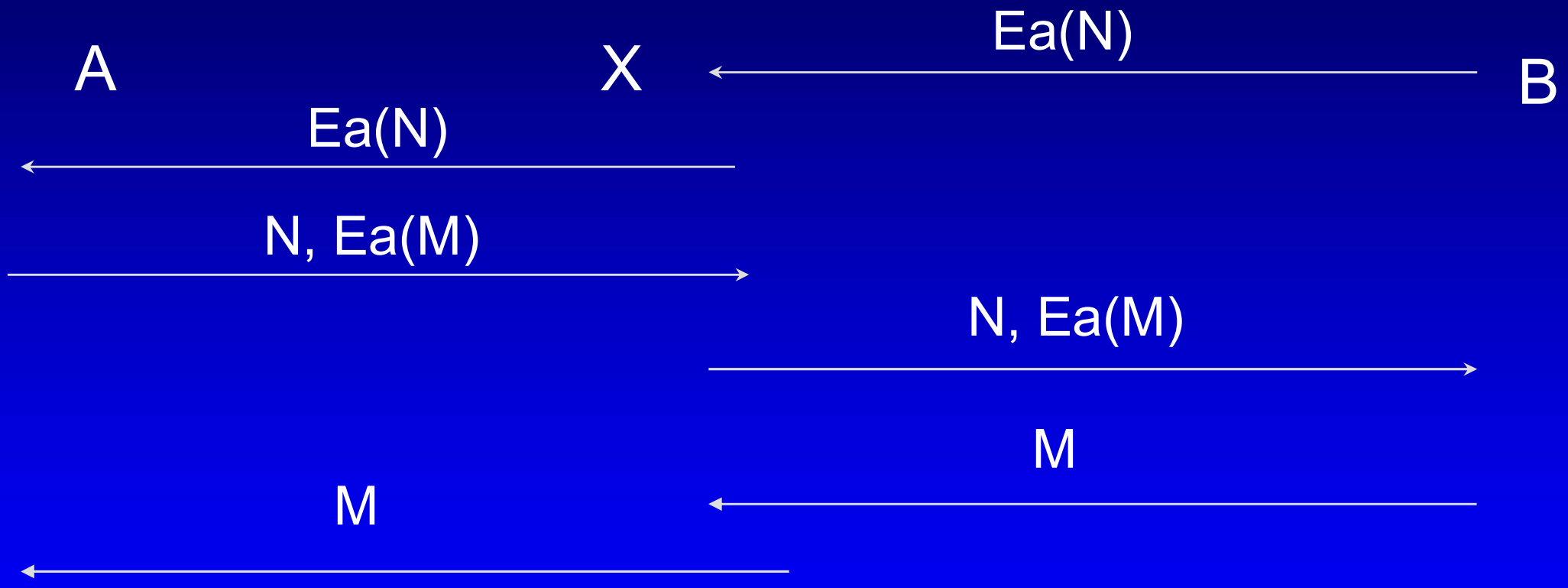


Nonces

- Autentificación bidireccional



Ataque sesiones Oráculo



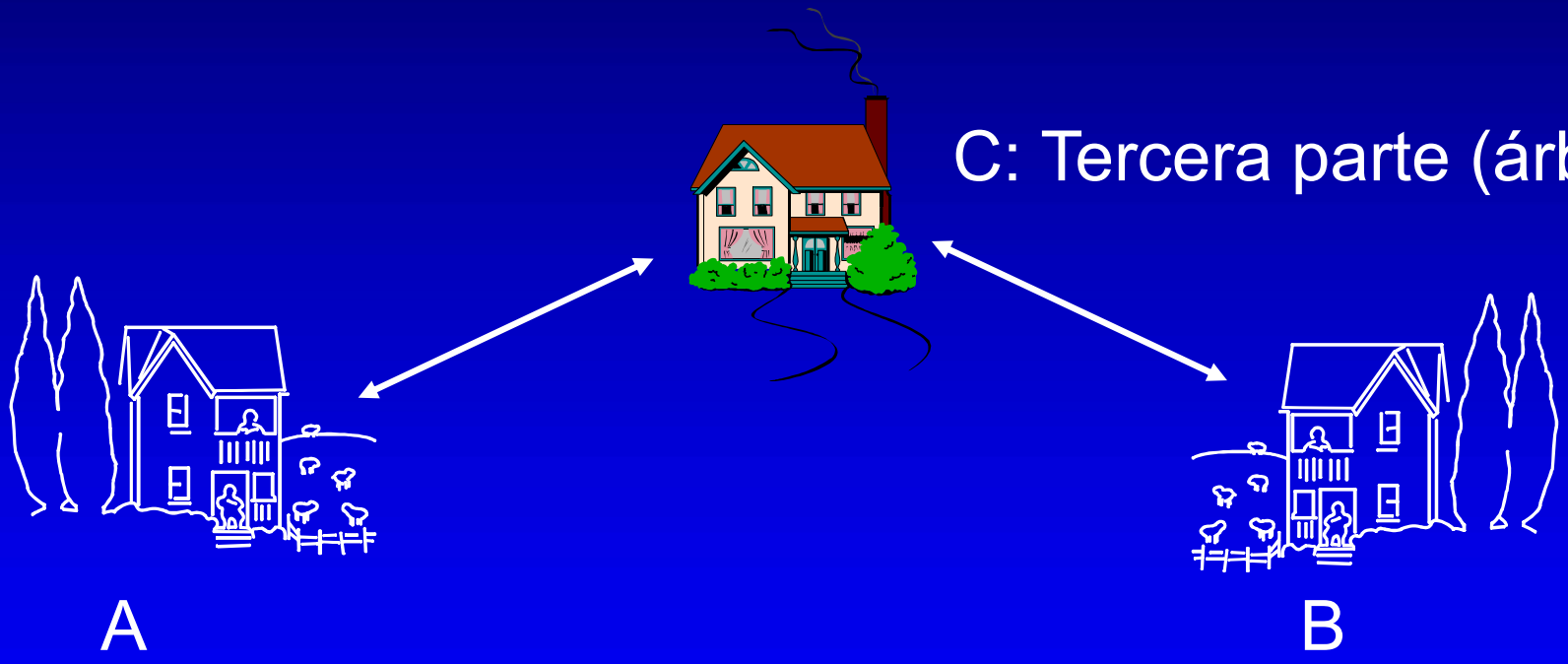
Es posible usar mensajes que aparecen en el segundo paso para reconstruir mensajes del tercer paso.

Protocolos de seguridad

- Todos los participantes deben conocer el protocolo y todos sus pasos
- Todos los participantes deben estar de acuerdo en seguir el protocolo
- El protocolo no puede ser ambiguo
- Debe ser completo (especificar una acción para cada posible situación).

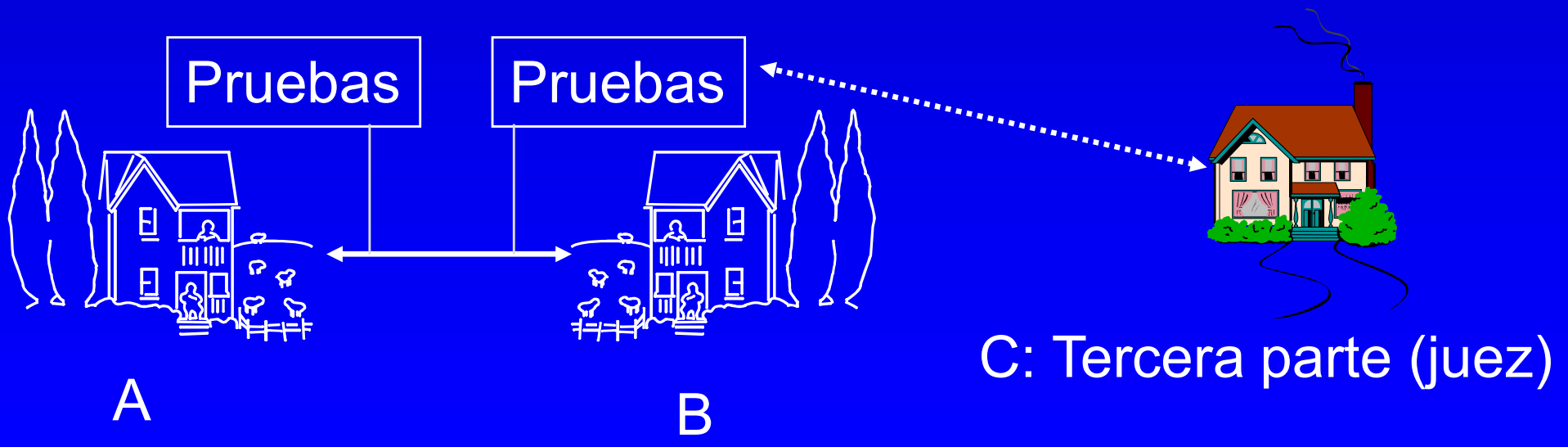
Tipos de protocolos criptográficos

- Protocolo arbitrado (notarios, banqueros,...)



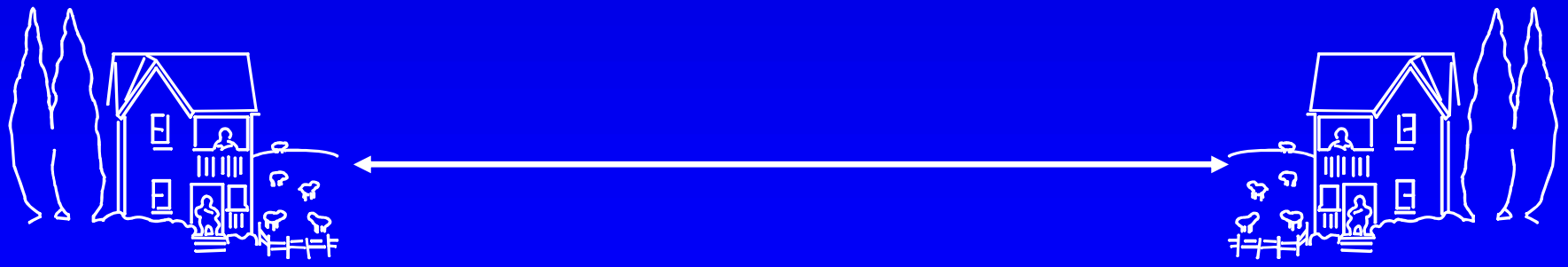
Tipos de protocolos criptográficos

- Protocolo judicial
- Dos subprotocolos: uno no arbitrado ejecutado varias veces y el otro es arbitrado (ejecutado sólo cuando hay disputas).



Tipos de protocolos criptográficos

- Protocolo robusto
  - El propio protocolo garantiza justicia, sin requerir árbitros o jueces.
  - Si un participante intenta mentir, el otro lo detecta inmediatamente y el protocolo detiene su ejecución.



Protocolos de gestión de claves

- Algoritmos de cifrado de dominio público
- Robustez depende confidencialidad clave
- Problema: Generar, distribuir claves
- Problema proporcional al número de usuarios
- Sistema de jerarquías

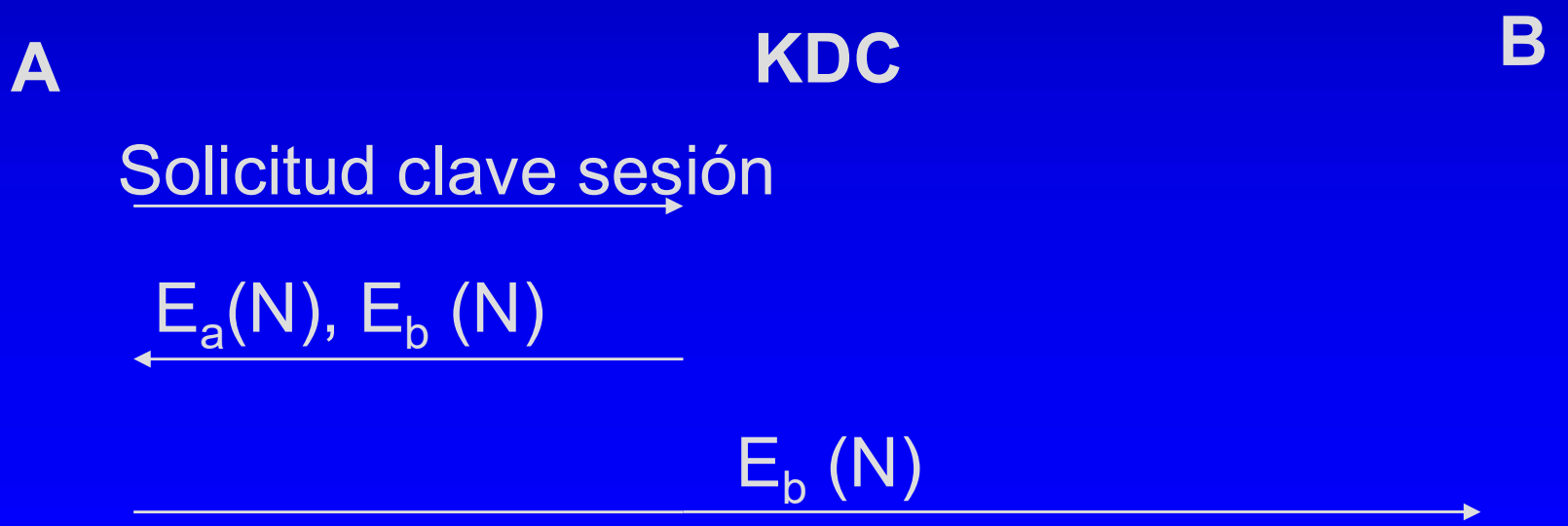
Distribución de claves

- Número de usuarios elevado
- Enlaces temporales
- Generación de claves sólo cuando es preciso
- Técnicas
  - Entre las dos estaciones
  - Basadas en centros de distribución KDC

Protocolos básicos. Clave simétrica

- Intercambio de claves. Se requiere KDC

N: Clave de sesión. Número aleatorio generado por KDC
a: Clave simétrica que comparte A con KDC
b: Clave simétrica que comparte B con KDC



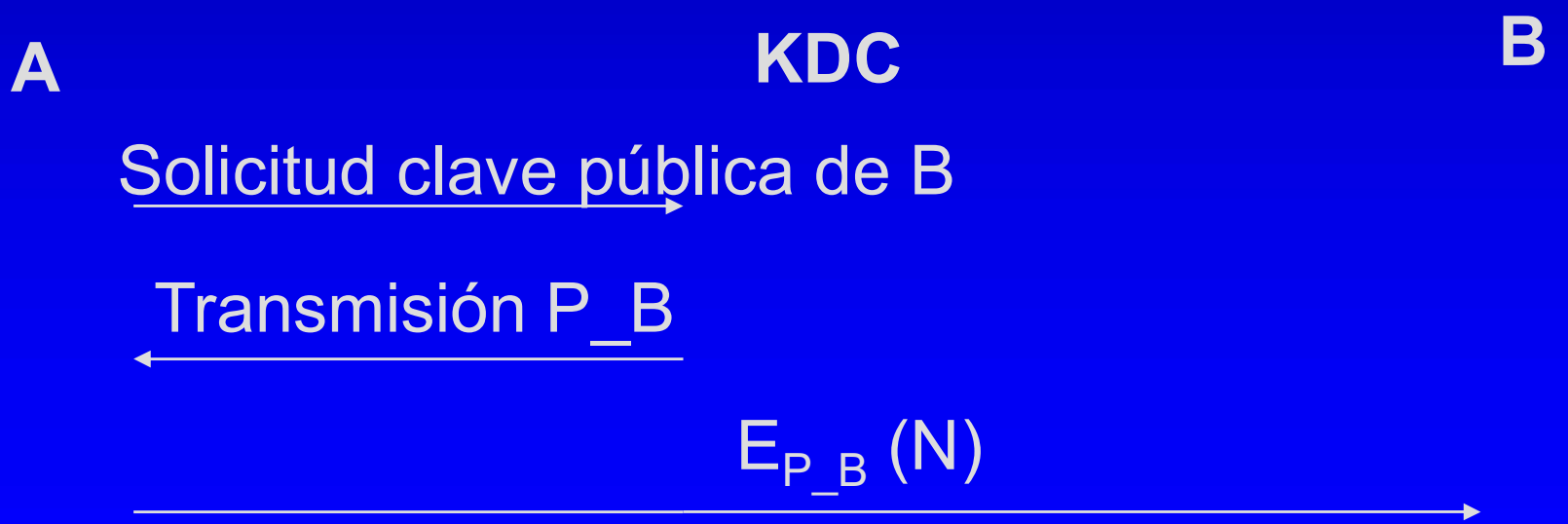
Protocolos básicos. Clave simétrica

- Problemas
- Protocolo basado en confianza absoluta en KDC. Manipulación KDC?
- KDC puede ocasionar cuellos de botella

Protocolos básicos. Clave pública

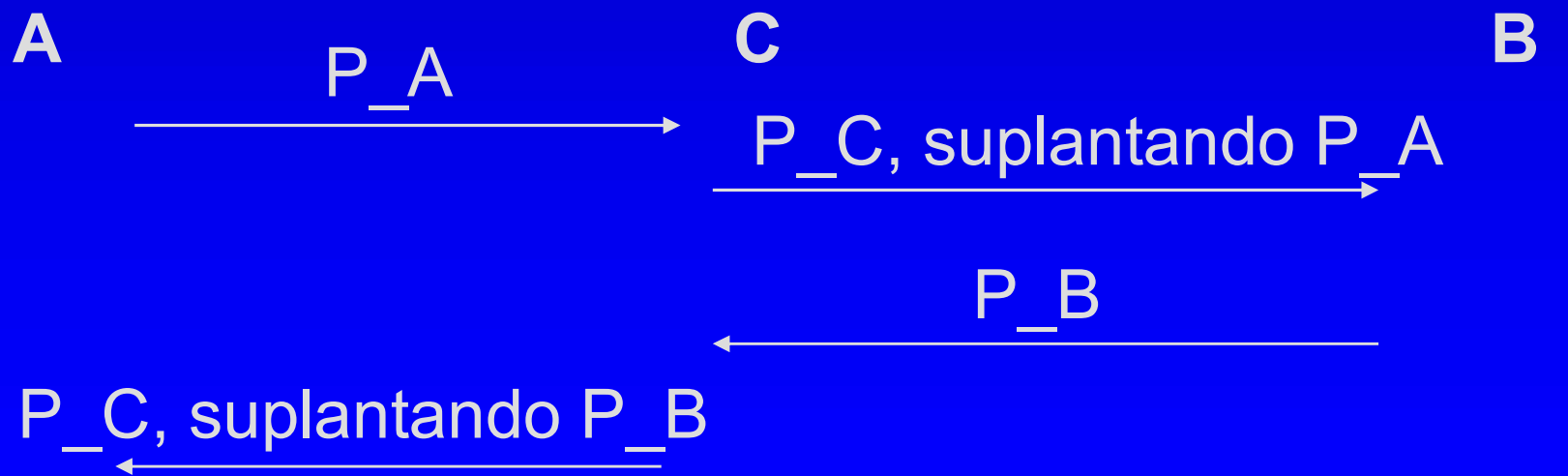
- Intercambio de claves

N: Clave de sesión. Número aleatorio generado por A
KDC contiene las claves públicas de todos los usuarios



Protocolos básicos. Clave pública

- Ataque Man in the Middle
- A envía a B P_A. C intercepta el mensaje y envía a B su propia clave pública, suplantando la identidad de A
- Se repite el proceso cuando B envía a A P_B



Protocolos básicos. Clave pública

- Cuando A envía un mensaje a B, C puede descifrarlo y reenviarlo a B, y viceversa. También puede modificarlo
- Ataque posible aunque P_A y P_B estén en bases de datos, si C puede interceptar las peticiones de A y B a la base de datos.
- Causa: A y B no pueden verificar que están hablando con el otro.

Protocolo interbloqueo

- A envía a B P_A y B transmite P_B a A
- A cifra el mensaje con P_B. Envía la mitad del mensaje. B realiza el mismo proceso
- A envía la otra mitad del mensaje, y a continuación B hace lo mismo
- B y A unen las dos mitades, y las descifran con sus respectivas claves privadas.
- Aunque C haya capturado las claves públicas no puede descifrar hasta tener todo el mensaje. C debería enviar un mensaje nuevo a B. La conversación entre A y B es totalmente distinta.

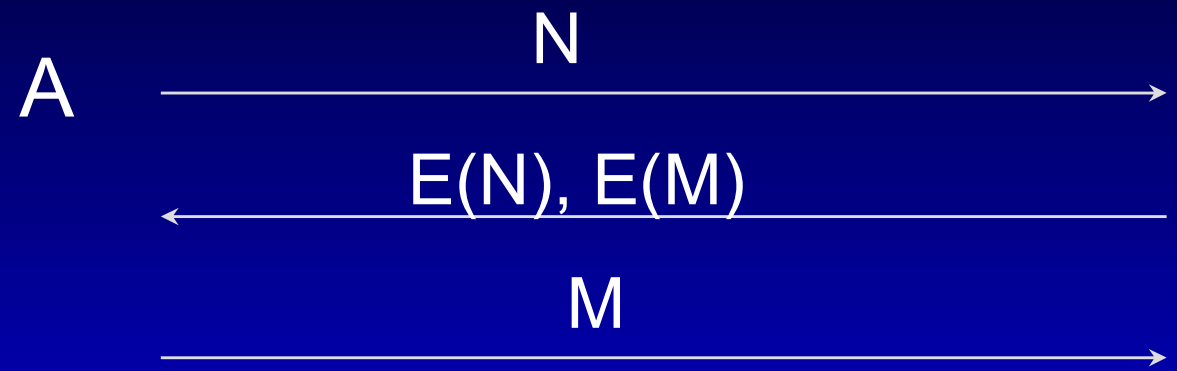
Intercambio de claves con firmas digitales

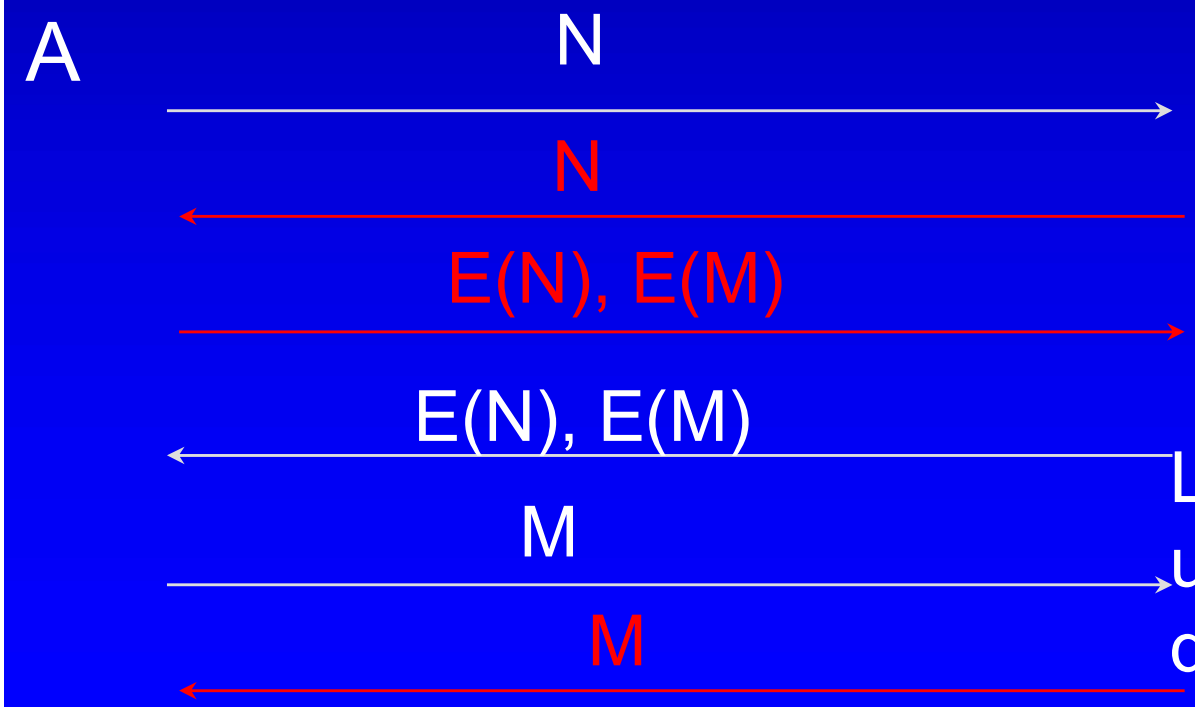
- Evita el ataque man-in-the-middle
- KDC firma P_A y P_B. Dicha firma va unida a un certificado
- A genera una clave de sesión aleatoria, K y cifra M usando K
- A obtiene P_B de la base de datos
- A envía $E_K(M)$ y $E_{P\_B}(K)$ y firma la transmisión

Gestión de claves (autentificación)

- Verificación de la identidad de una entidad y de la integridad de los datos generados
- Fundamental en la distribución de claves
- Fundamental en entornos abiertos

Protocolo autentificación ISO SC27

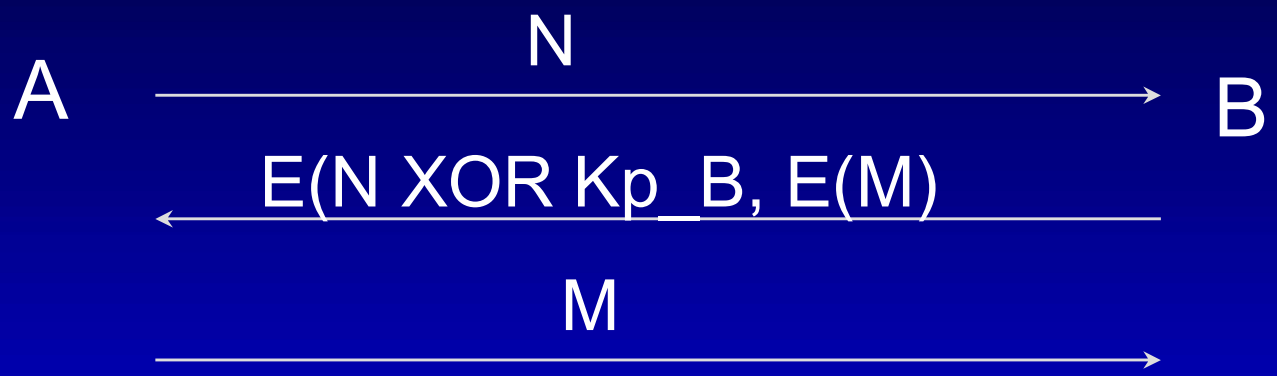


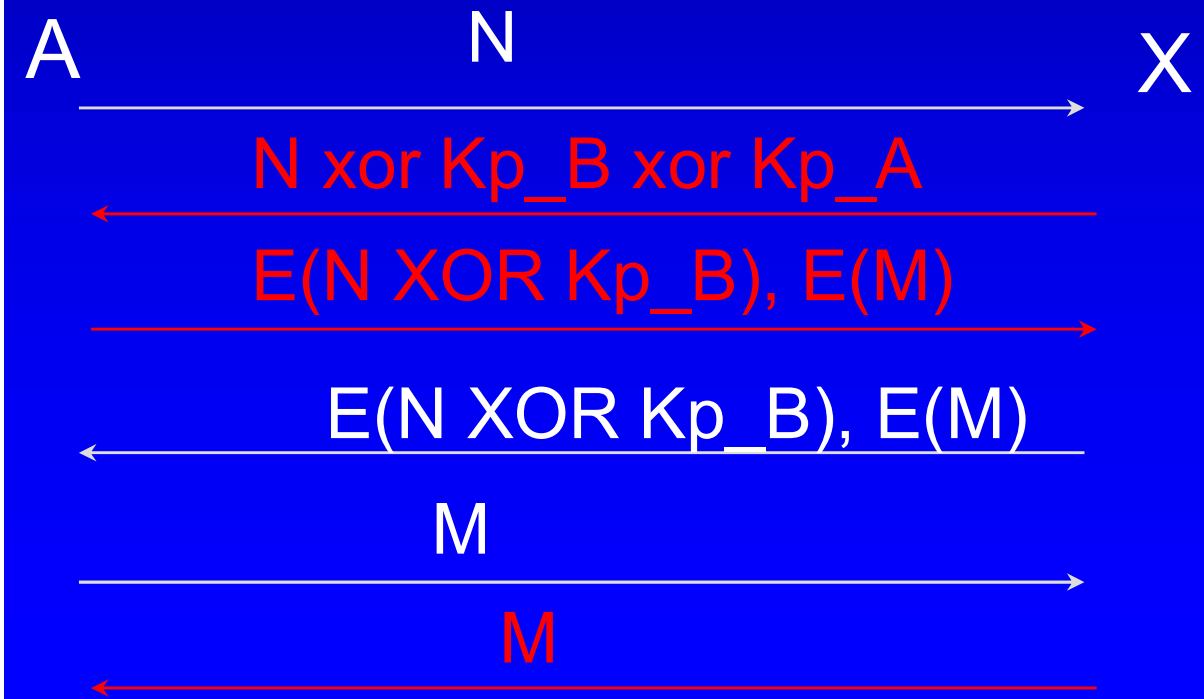


Ataque mediante sesiones paralelas

La expresión criptográfica usada en el segundo mensaje debe depender de la dirección

Protocolo autentificación ISO SC27 modificado





Ataque mediante suma con offset

Conclusiones dis. protocolos autentificación

- Protocolos muy simples, aparentemente seguros son susceptibles a ataques que incluyan reactuación y reutilización de respuestas a retos observadas en otras ejecuciones del protocolo
- Son necesarios mecanismos de verificación de la seguridad de los protocolos

Protocolo NS usando claves asimétricas

- Cada usuario conoce la clave pública del servidor de autentificación AS, y confían en él

PROTOCOLO DE DISTRIBUCIÓN DE CLAVES

- A-->AS: A, B (pide clave pública de B)
- AS->A: S_AS(B, P_B)

Protocolo NS usando claves asimétricas

PROTOCOLO DE CONEXIÓN

- A-->B: P_B (N_A,A) (envía un nonce)
- B-->A: P_A (N_A,N_B) (responde y envía otro nonce)
- A-->B: P_B (N_B) (responde el nonce)

SKEY

- Programa para autentificación basado en funciones de hash
- A introduce un número aleatorio N
- El terminal calcula f(N), f(f(N)), ..., y así hasta unas cien veces (x_1, x_2, x_3, ..., x_{100}) Se imprimen esos números y A los guarda en su cartera. El terminal calcula también x_{101}, y lo almacena en claro en la base de datos.
- A introduce x_{100}, y se calcula f(x_{100}). Cada número se usa sólo una vez

Compartición de secretos

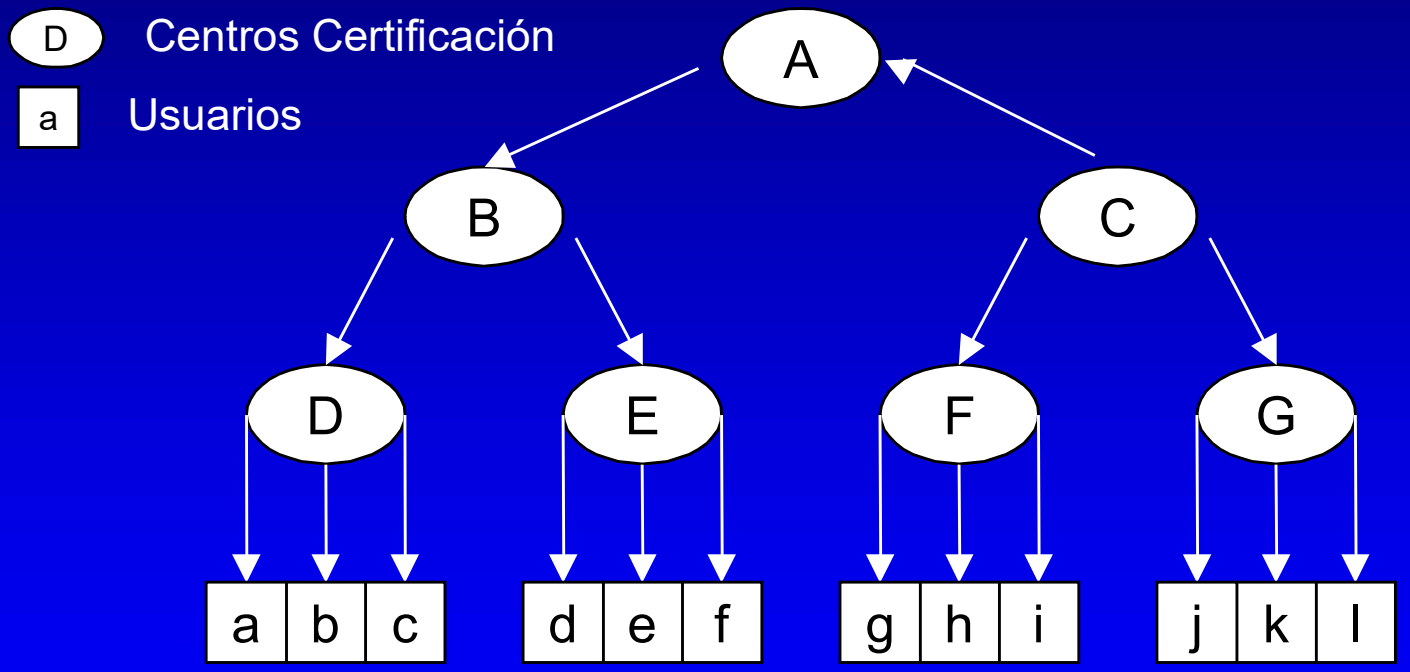
- Esquemas umbral (m,n)
  - El mensaje o secreto se divide en n partes
  - El mensaje se puede reconstruir con m partes
  - Problemas
    - Existencia de tramposos.
      - Se sabe que hay uno, pero no quien es
      - El tramposo consigue una parte válida de los otros, sin ser descubierto
      - Pruebas de existencia de mentirosos

Estructura Certificados

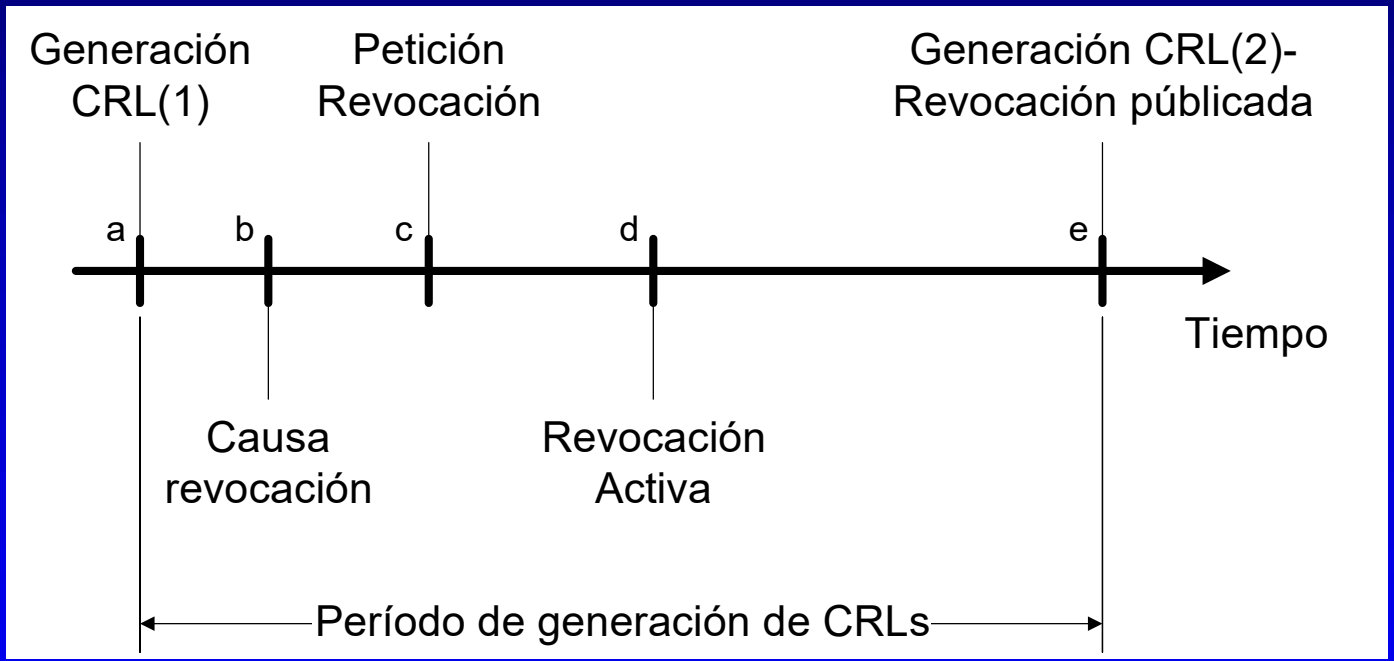
Funciones Centros Certificación

- Emisión de certificados para nuevos usuarios
- Rutinas para modificar o dar de baja un certificado
- Generar listas de revocación
- Comunicarse con otros centros de certificación (estructuras jerárquicas)

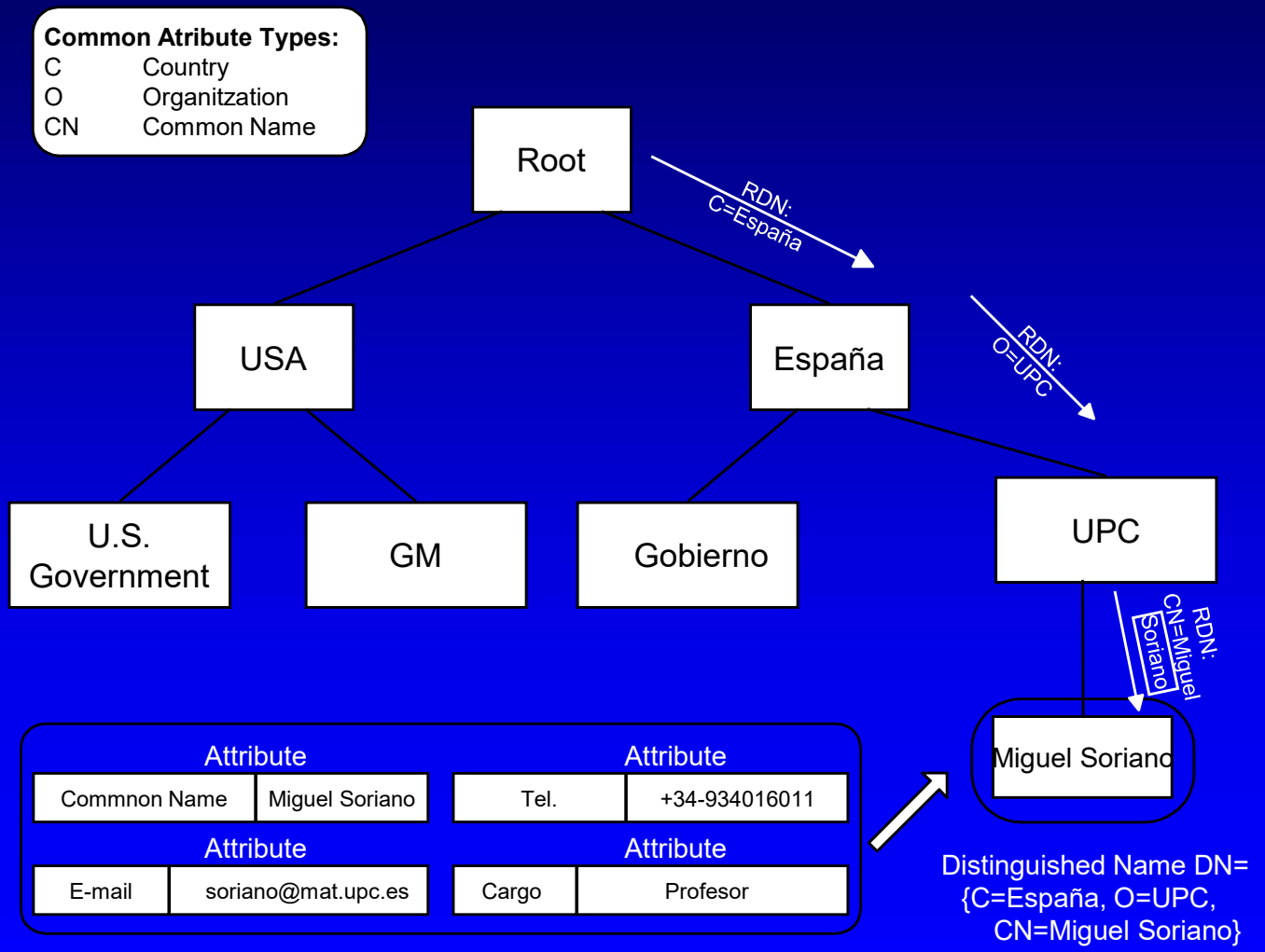
Estructura Jerárquica



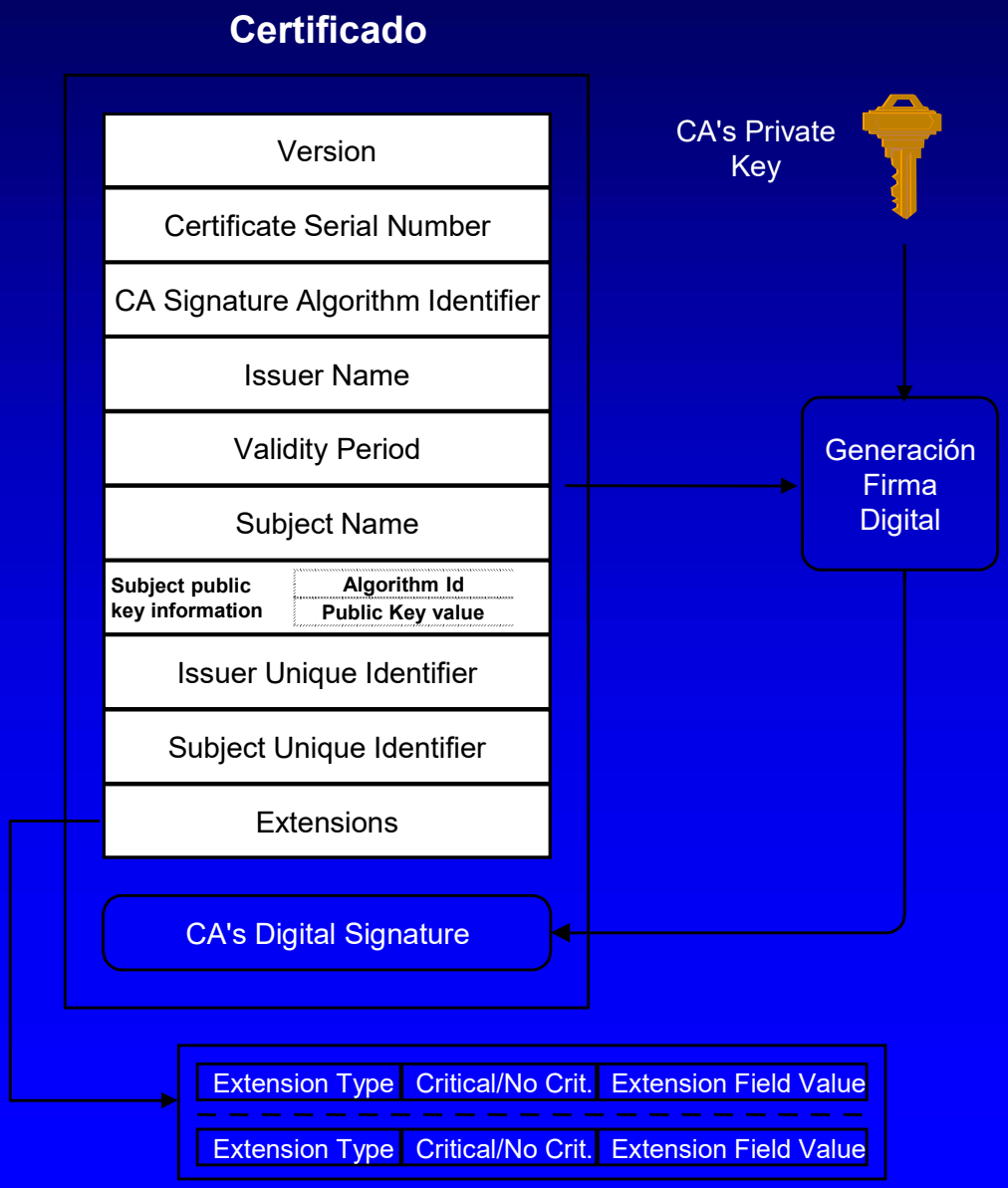
Revocación


Generación
CRL(1)
Petición
Revocación
Generación CRL(2)-
Revocación públicada
a
b
c
d
e
Tiempo
Causa
revocación
Revocación
Activa
Período de generación de CRLs

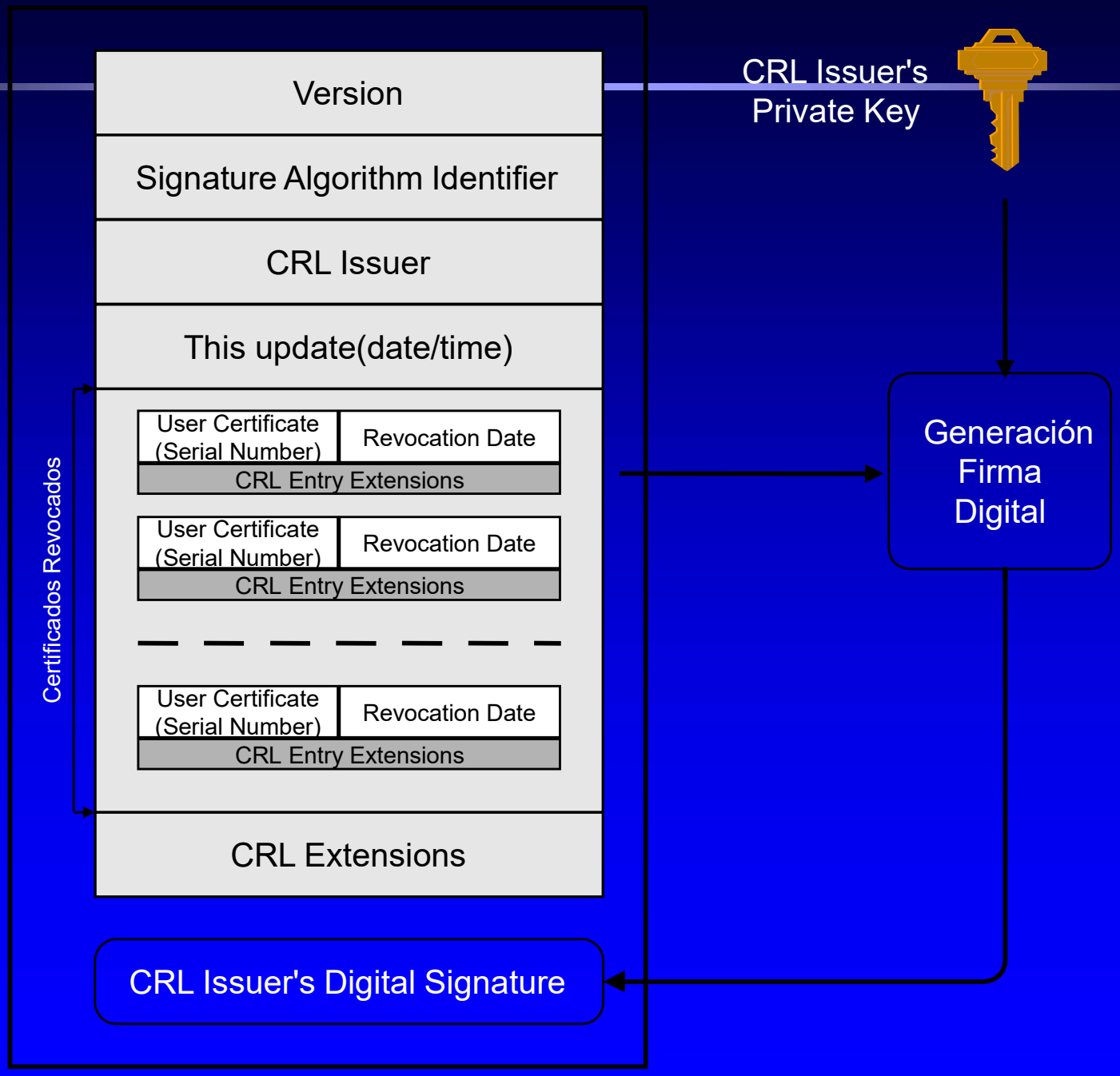
El Directorio X.500



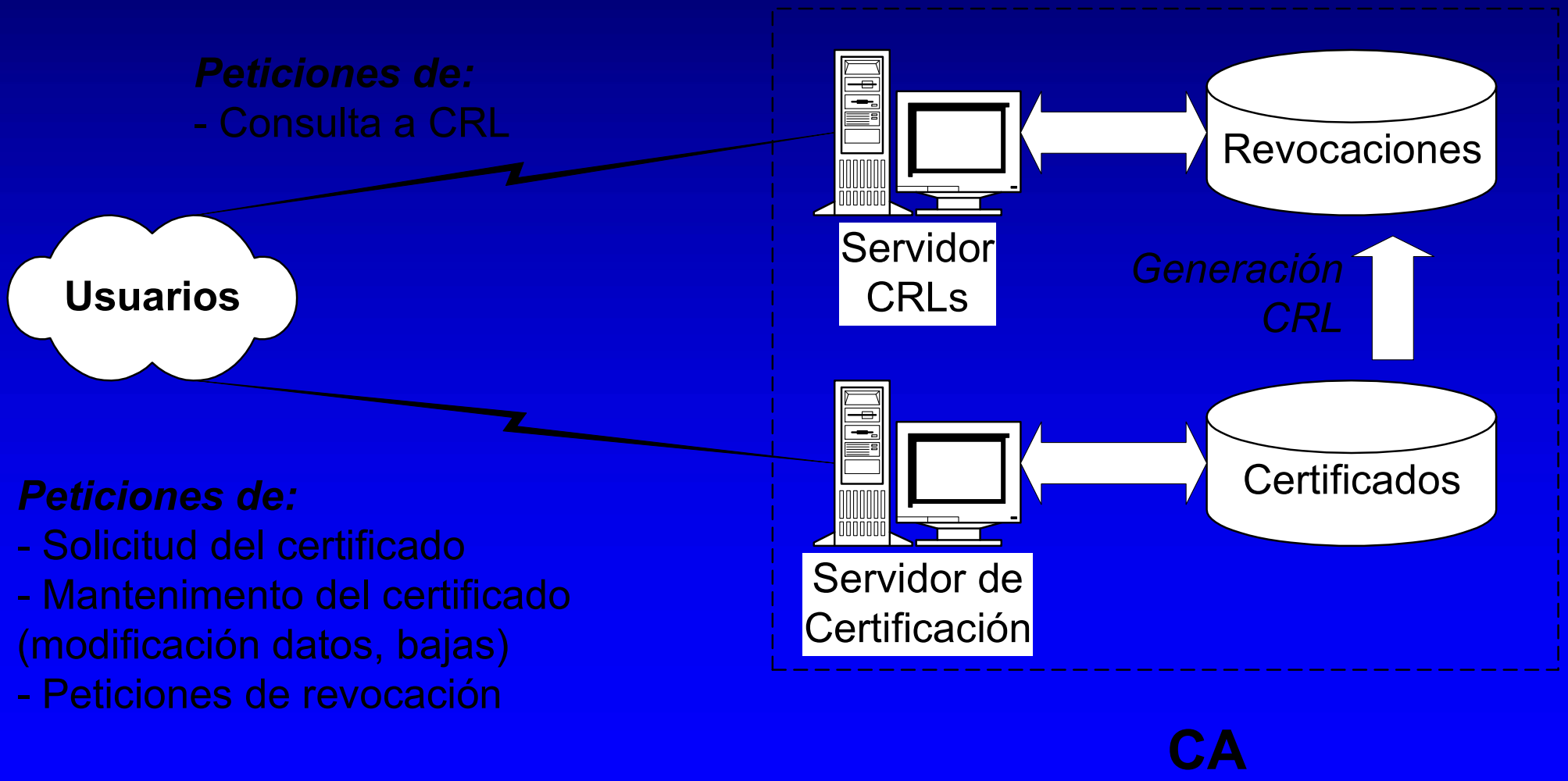
Certificado X.509



CRL


Version
Signature Algorithm Identifier
CRL Issuer
This update(date/time)
User Certificate (Serial Number)
Revocation Date
CRL Entry Extensions
User Certificate (Serial Number)
Revocation Date
CRL Entry Extensions
User Certificate (Serial Number)
Revocation Date
CRL Entry Extensions
Certificados Revocados
CRL Extensions
CRL Issuer's Digital Signature
CRL Issuer's Private Key
Generación Firma Digital

Certificados con Revocaciones



Certificados Netscape

This Certificate belongs to:
Miguel Soriano
soriano@mat.upc.es
Certificados Internet
IPS Seguridad
Barcelona, Barcelona, ES

This Certificate was issued by:
IPS SEGURIDAD
CERTIFICADOS
IPS SEGURIDAD
MADRID, MADRID, ES

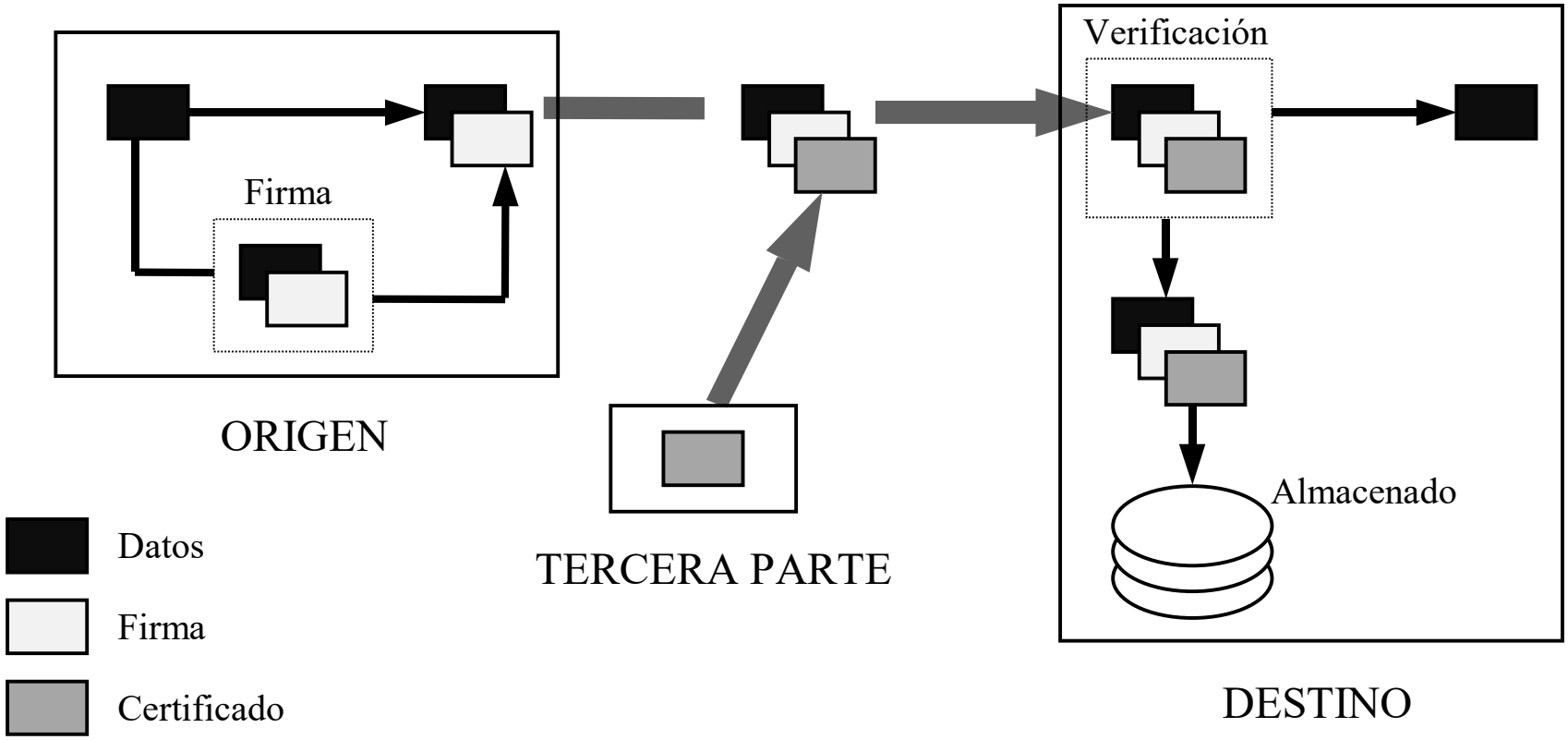
Serial Number: 61:EC:C8:3A:00:00:00:8A
This Certificate is valid from Tue Jun 09, 1998 to Wed Jun 09, 1999
Certificate Fingerprint:
01:F1:CB:F2:CB:F4:16:88:4F:6F:EE:39:0E:61:59:0E

No Repudio

- Repudio de Origen
  - Hay desacuerdo sobre si una determinada parte originó un determinado dato, o hay desacuerdo en el instante en que se originó.
- Repudio de Entrega
  - Hay desacuerdo sobre si una determinada parte recibió un determinado dato, o hay desacuerdo sobre el instante en que se recibió.
  - El servicio se ha de implementar en la capa de Aplicación

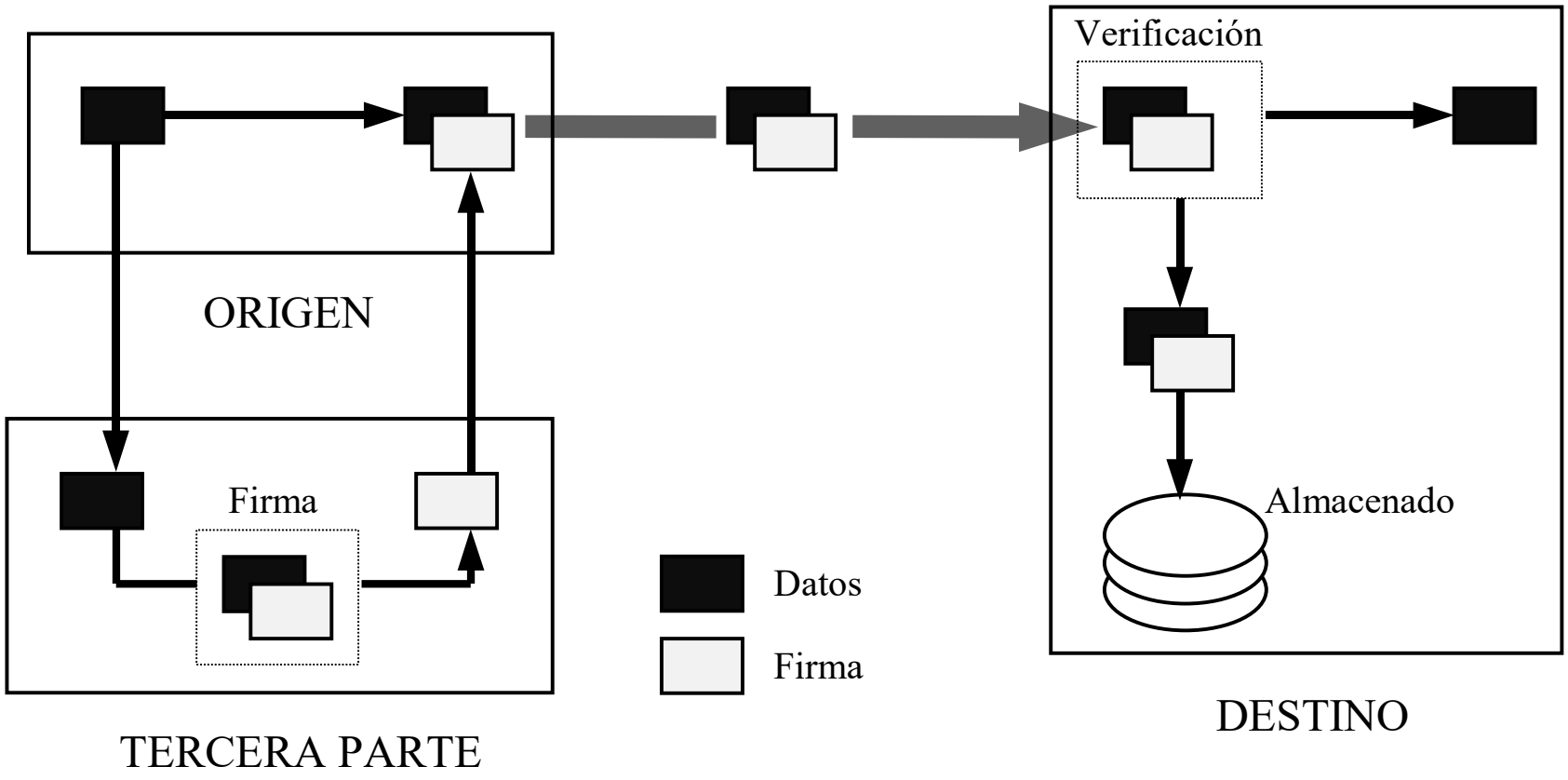
No Repudio Origen

Firma Digital del Origen



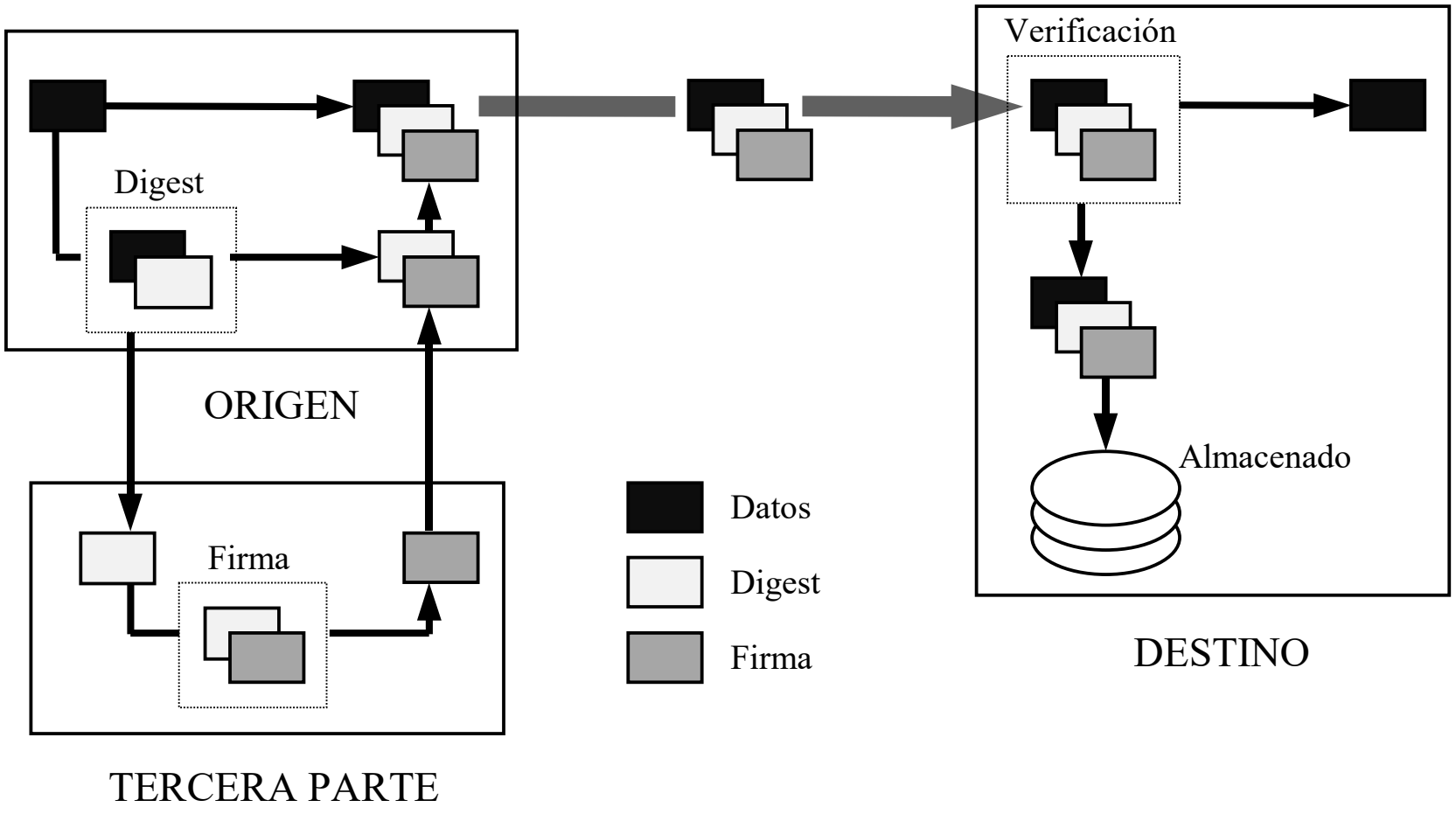
No Repudio Origen

Firma Digital de una Tercera Parte sobre los Datos



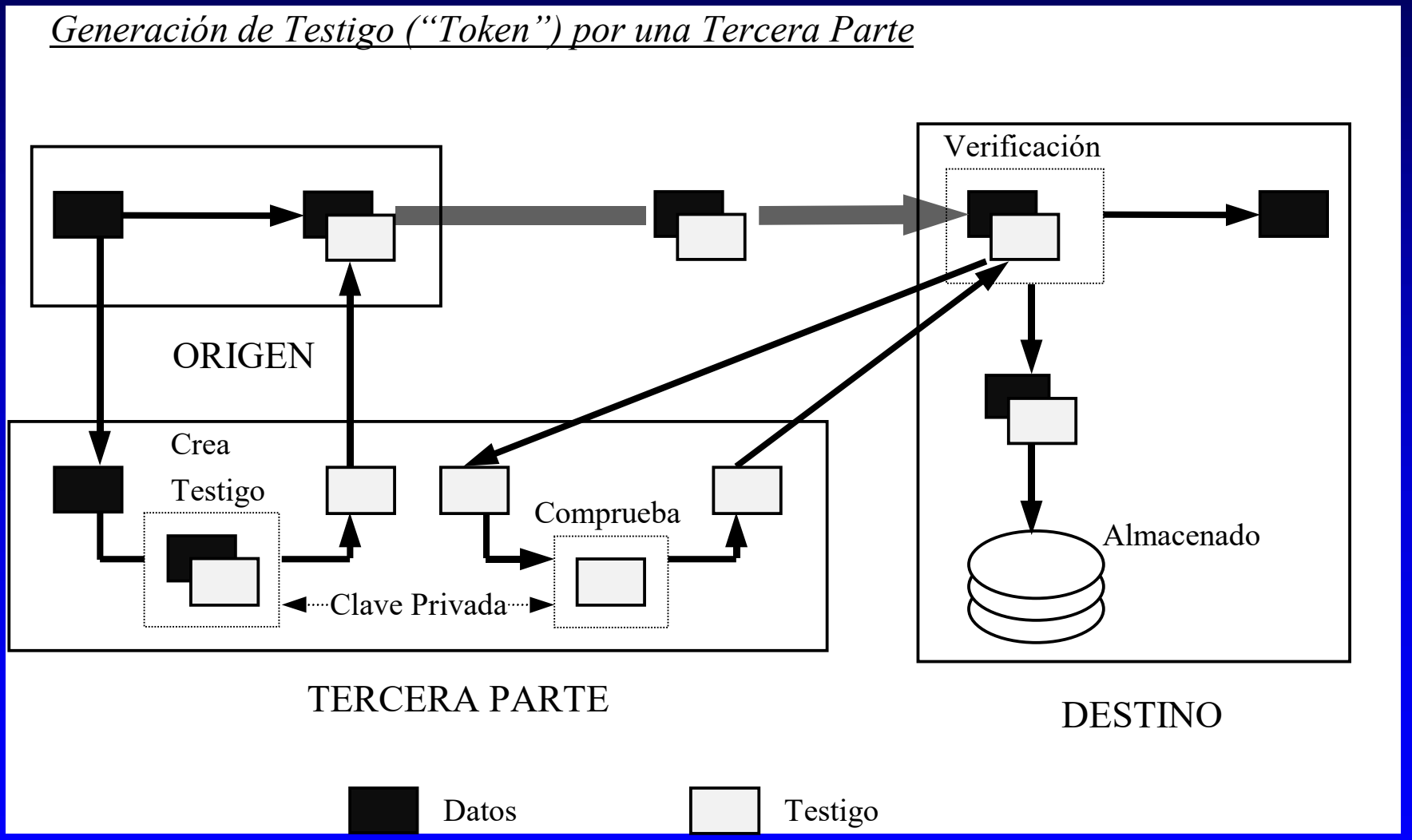
No Repudio Origen

Firma Digital de una Tercera Parte sobre un "Digest"



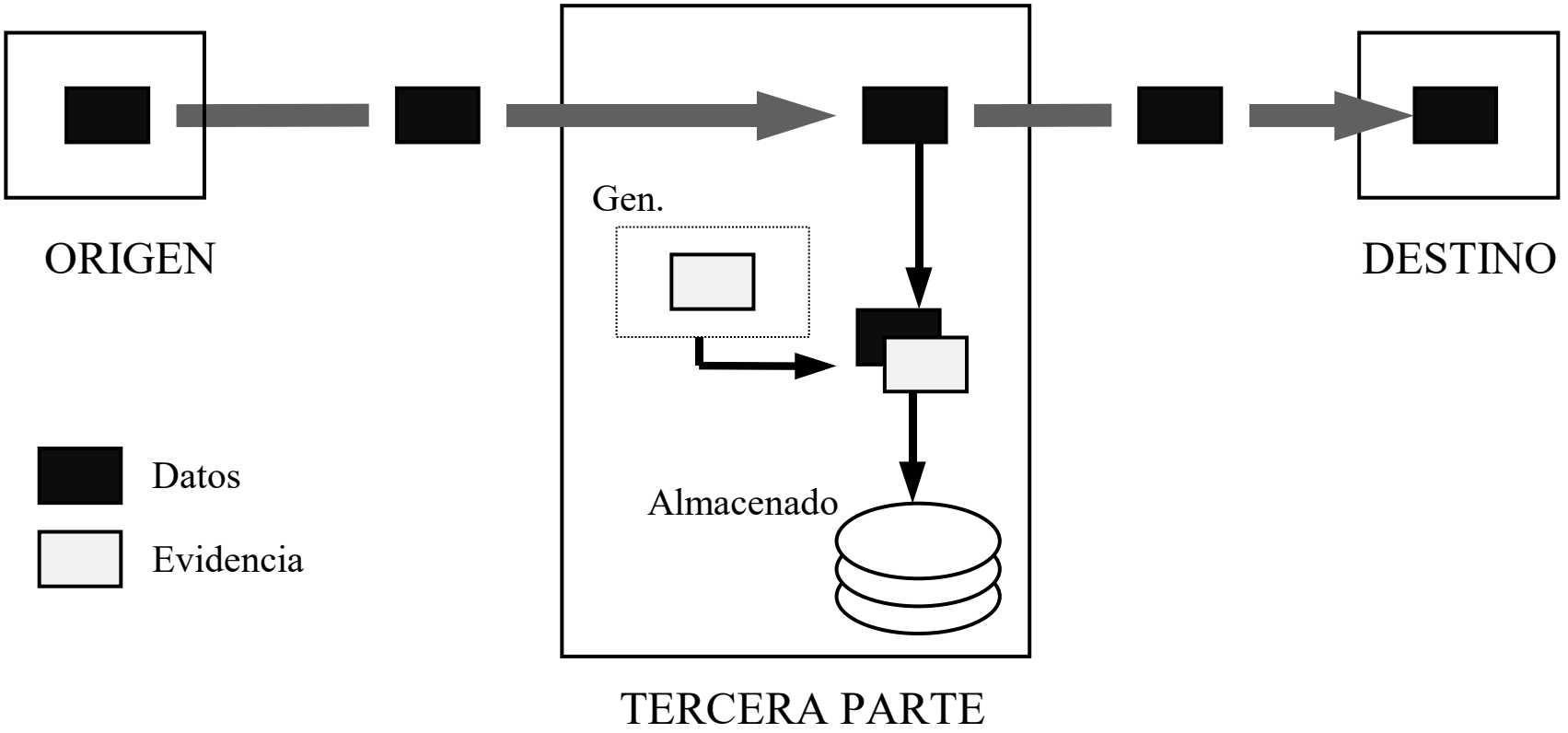
No Repudio Origen

Generación de Testigo ("Token") por una Tercera Parte

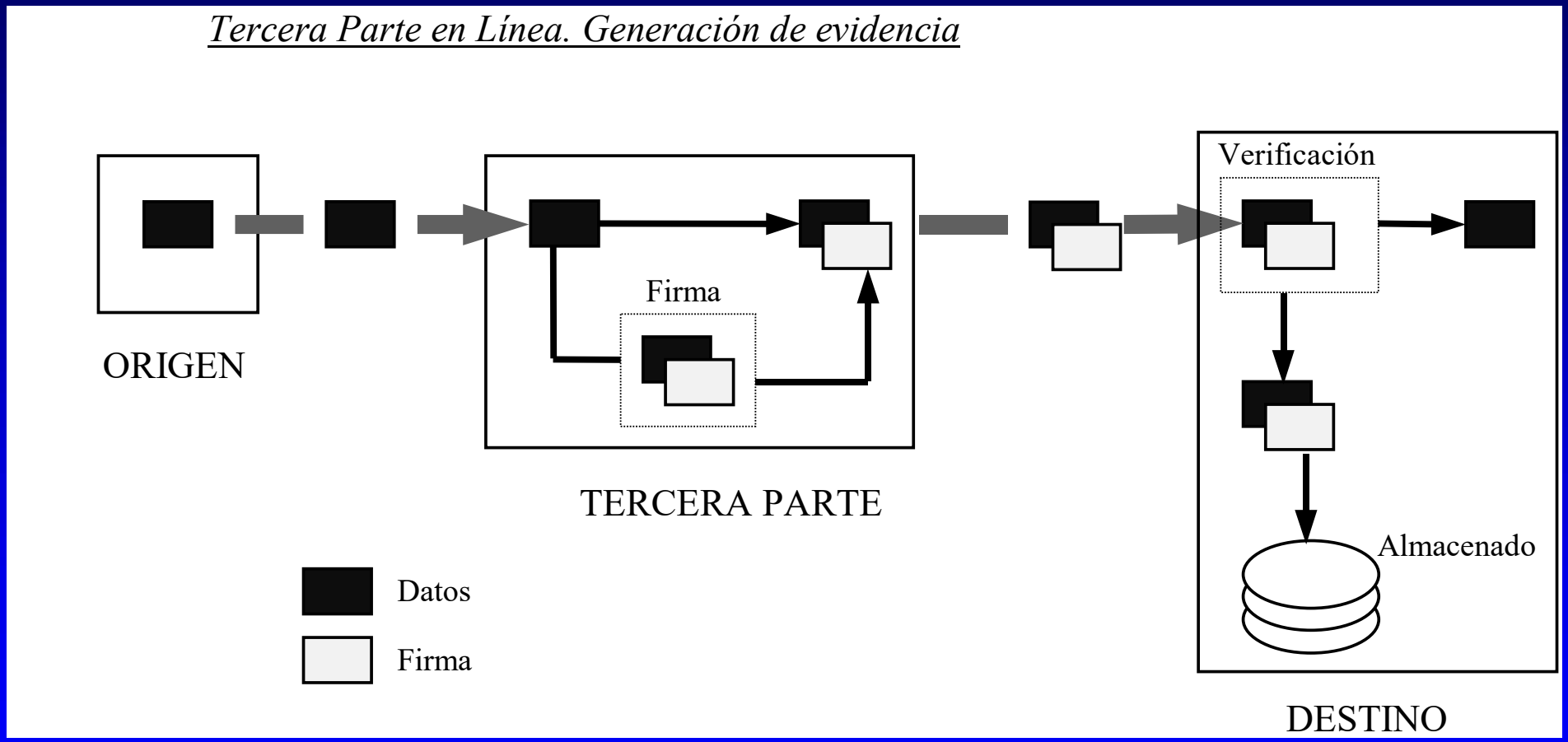


No Repudio Origen

Tercera Parte en Línea. Almacenamiento de la evidencia

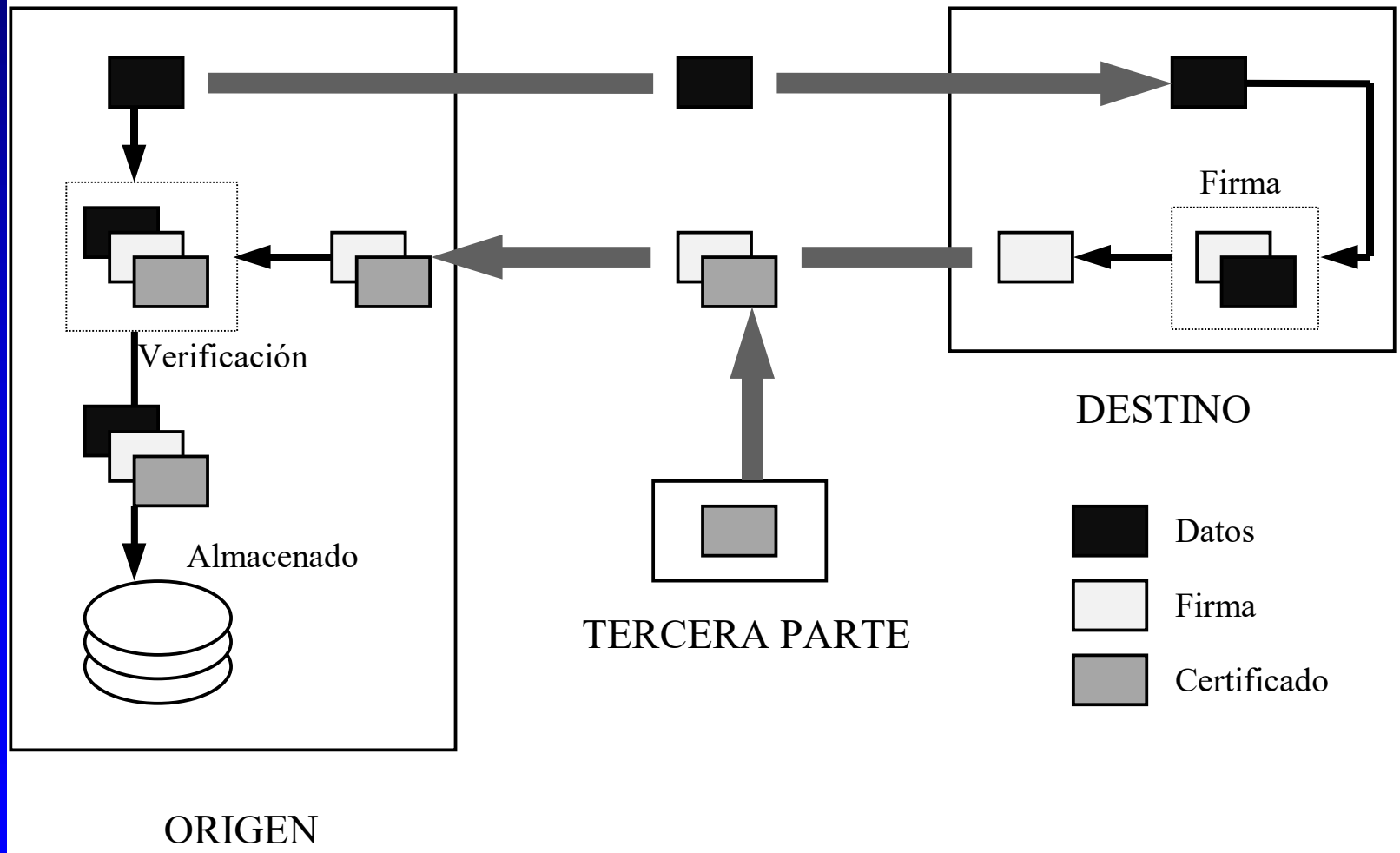


No Repudio Origen

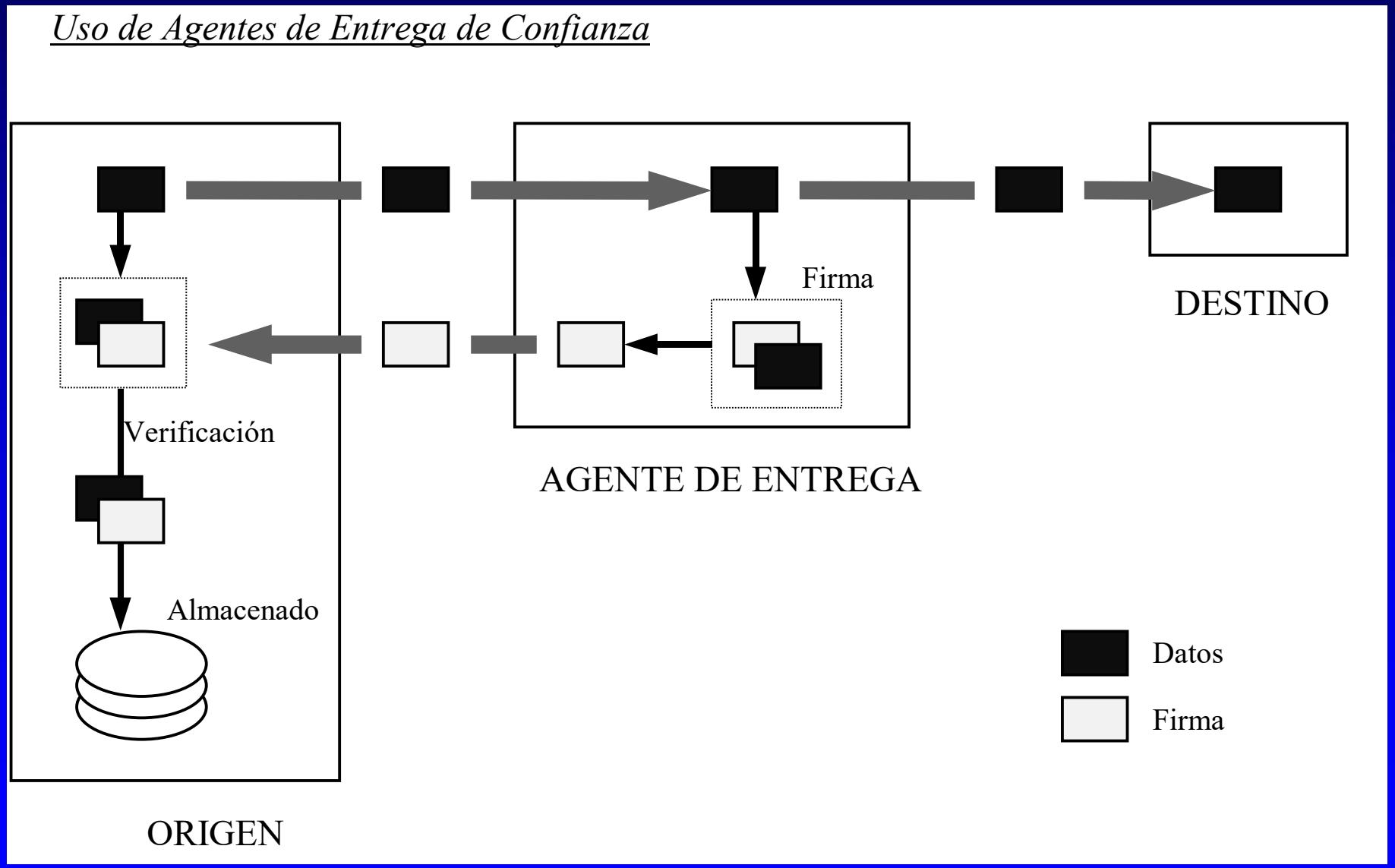


No Repudio de Entrega

Reconocimiento de Recepción mediante firma del Destinatario



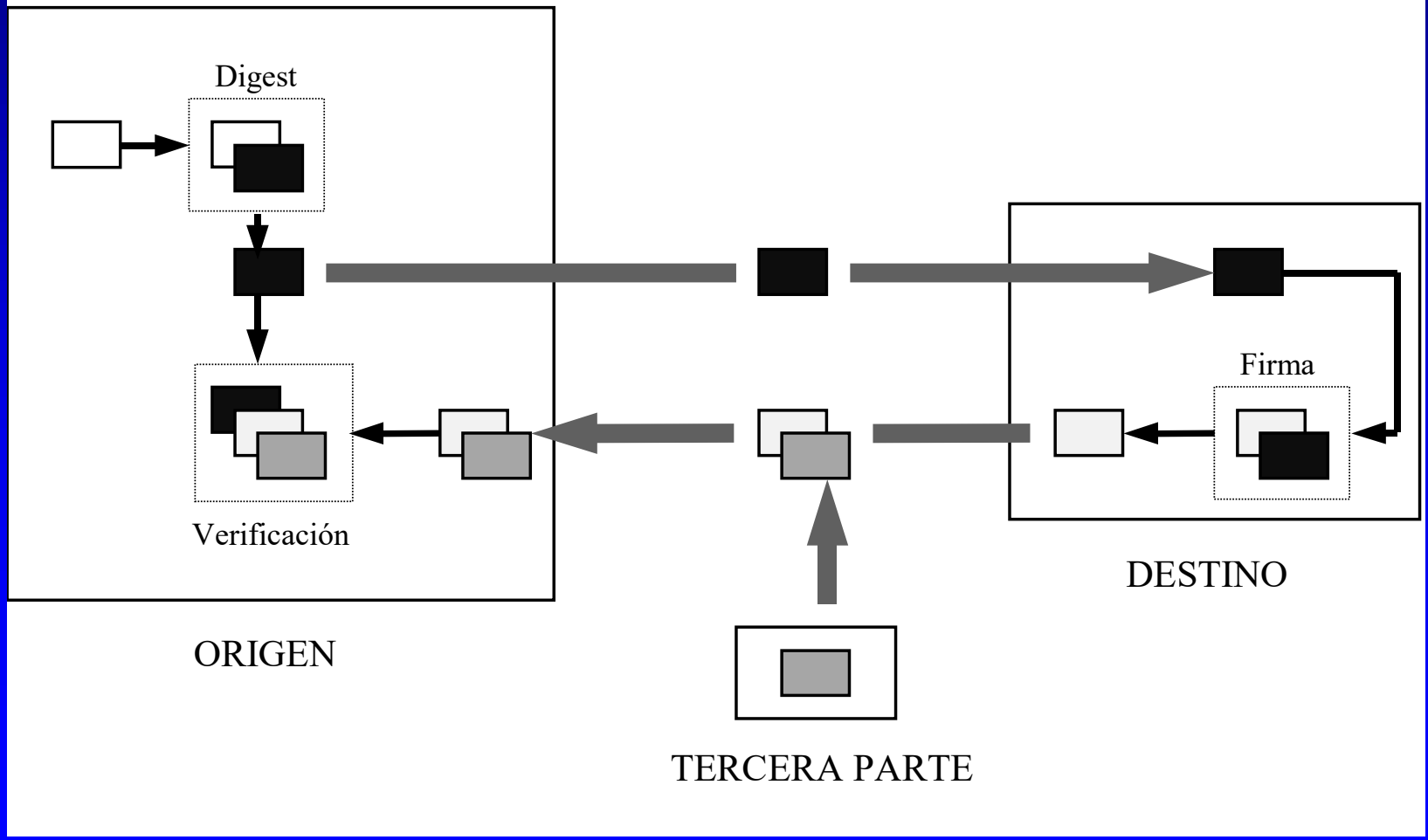
No Repudio de Entrega



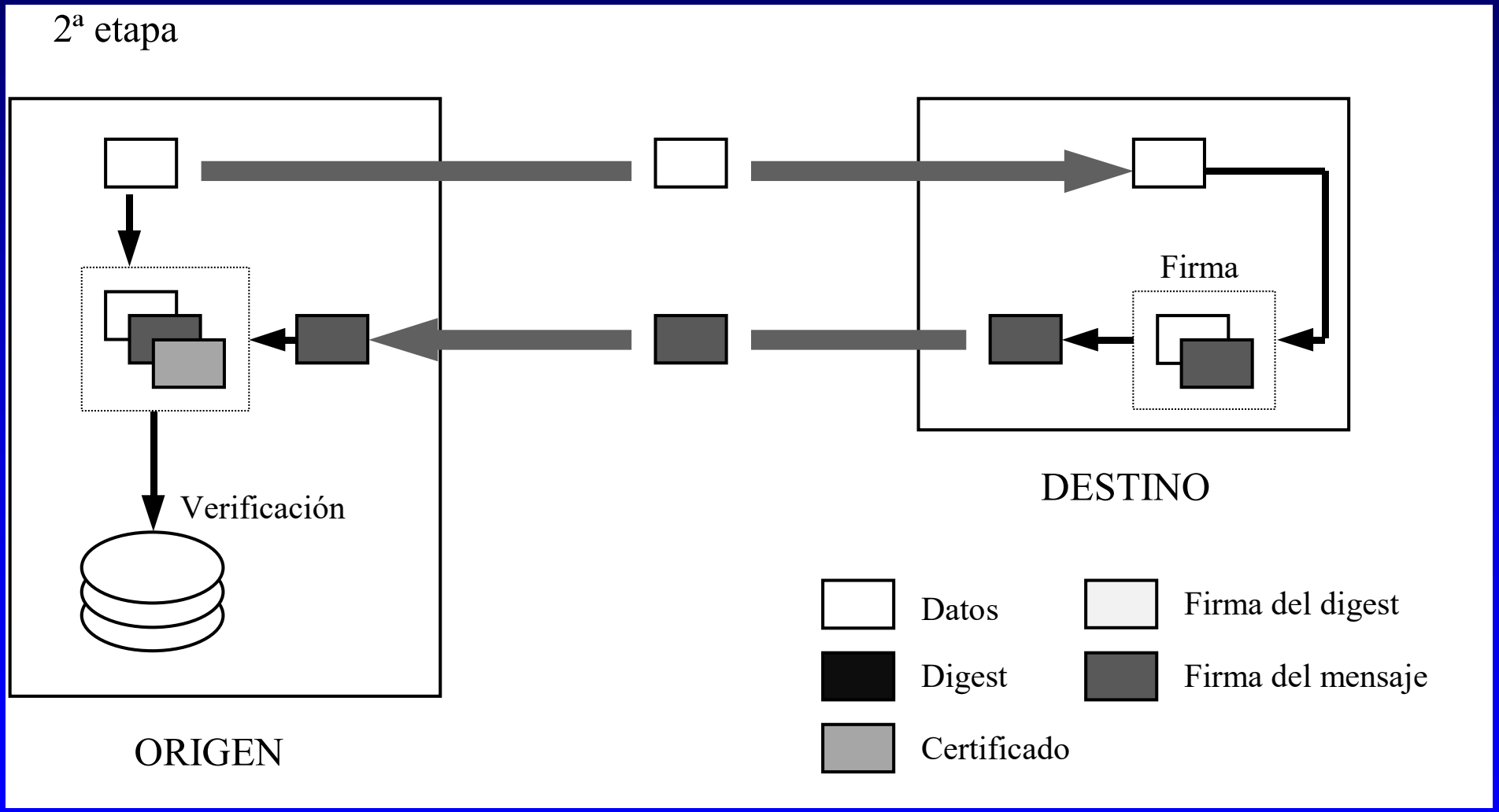
No Repudio de Entrega

Entrega en Dos Etapas

1ª etapa



No Repudio de Destino



Esteganografía

- Ocultar mensajes dentro de otros mensajes
- Equivalencia a tinta invisible
- Actualmente, ocultar mensajes en imágenes o audio
  - Ej. Modificar el bit menos significativo de todos los pixels de una imagen. La imagen no varía mucho, y el conjunto de bits, constituye otra imagen o texto oculto.

Esteganografía

El valor (1 1 1 1 1 1 1 1) es un número binario de 8 bits.
Al bit ubicado más a la derecha se le llama "bit menos significativo" (LSB) porque es el de menor peso, alterándolo cambia en la menor medida posible el valor total del número representado.

Ejemplo:
Un ejemplo de esteganografía: Ocultamiento de la letra "A". Si se tiene parte de una imagen con píxeles con formato RGB (3 bytes):

Esteganografía

Su representación original podría ser la siguiente (3 píxeles, 9 bytes):
(1 1 0 1 1 0 1 0) (0 1 0 0 1 0 0 1) (0 1 0 0 0 0 1 1)
(0 0 0 1 1 1 1 0) (0 1 0 1 1 0 1 1) (1 1 0 1 1 1 1 1)
(0 0 0 0 1 1 1 0) (0 1 0 0 0 1 1 1) (0 0 0 0 0 1 1 1)
El mensaje a cifrar es 'A' cuya representación ASCII es (1 0 0 1 0 1 1 1), entonces los nuevos píxeles alterados serían:
(1 1 0 1 1 0 1 **1**) (0 1 0 0 1 0 0 **0**) (0 1 0 0 0 0 1 **0**)
(0 0 0 1 1 1 1 **1**) (0 1 0 1 1 0 1 **0**) (1 1 0 1 1 1 1 **1**)
(0 0 0 0 1 1 1 **1**) (0 1 0 0 0 1 1 **1**) (0 0 0 0 0 1 1 1)
Observar que se ha sustituido el bit del mensaje (letra A, marcados en negritas) en cada uno de los bits menos significativos de color de los 3 píxeles. Fueron necesarios 8 bytes para el cambio, uno por cada bit de la letra A, el noveno byte de color no se utilizó, pero es parte del tercer pixel (su tercera componente de color).

Esteganografía

El método del LSB funciona mejor en los archivos de imágenes que tienen una alta resolución y usan gran cantidad de colores. En caso de archivos de audio, favorecen aquellos que tienen muchos y diferentes sonidos que poseen una alta tasa de bits.

Además este método no altera en absoluto el **tamaño** del archivo portador o cubierta (por eso es "una técnica de sustitución"). Posee la desventaja de que el tamaño del archivo portador debe ser mayor cuanto más grande sea el mensaje a embeber; se necesitan 8 bytes de imagen por cada byte de mensaje a ocultar; es decir, la capacidad máxima de una imagen para almacenar un mensaje oculto es de su 12,5%.

Si se pretende emplear una mayor porción de bits de la imagen (por ejemplo, no sólo el último, sino los dos últimos), puede comenzar a ser perceptible al ojo humano la alteración general provocada.

Esteganografía


+

=


GRACIAS POR SU ATENCIÓN