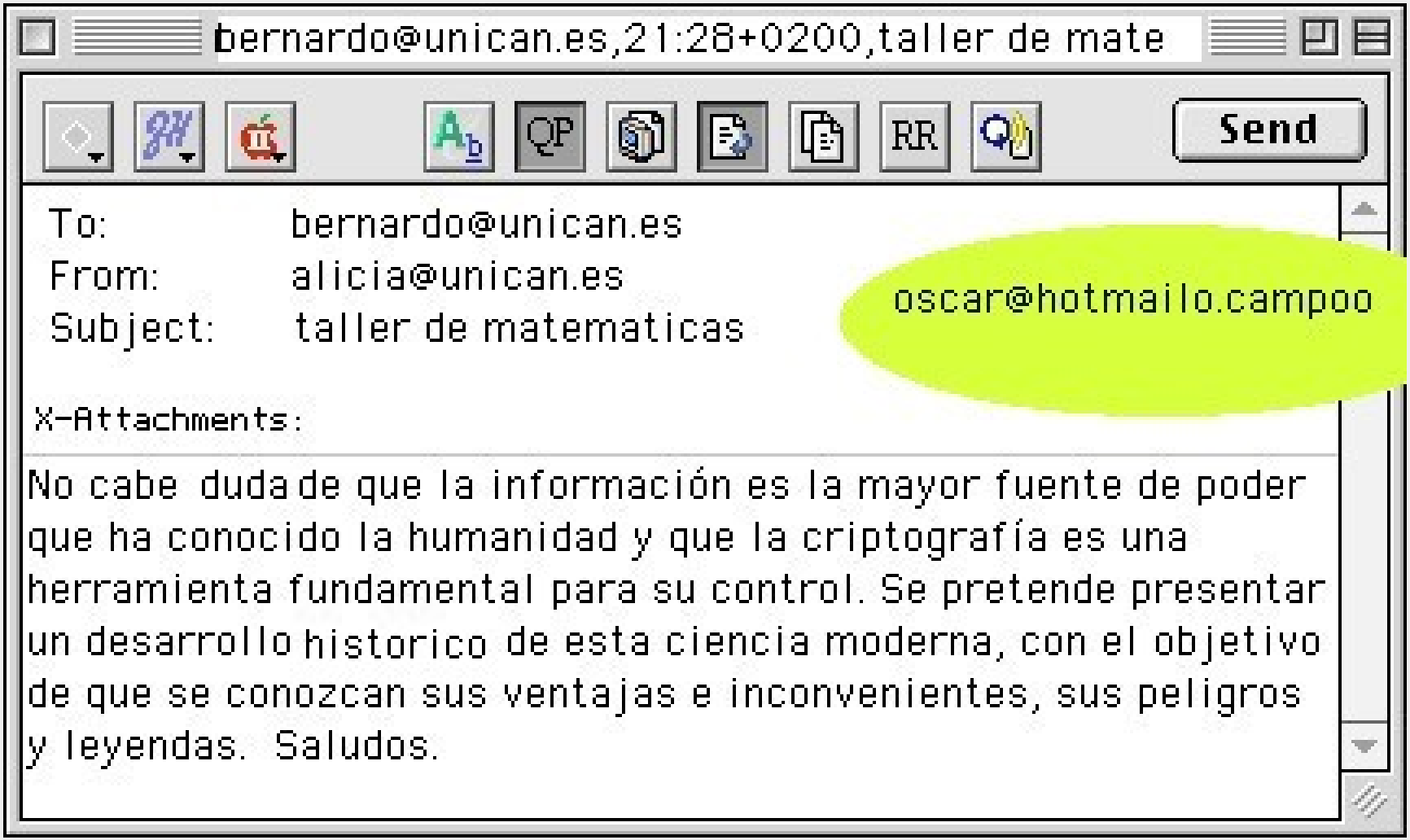
Criptografía (Matemáticas e Internet)

Jaime Gutiérrez

Universidad de Cantabria


bernardo@unican.es,21:28+0200,taller de mate
Send
To: bernardo@unican.es
From: alicia@unican.es
Subject: taller de matematicas
oscar@hotmailo.campoo
X-Attachments:
No cabe duda de que la información es la mayor fuente de poder que ha conocido la humanidad y que la criptografía es una herramienta fundamental para su control. Se pretende presentar un desarrollo historico de esta ciencia moderna, con el objetivo de que se conozcan sus ventajas e inconvenientes, sus peligros y leyendas. Saludos.

CRIPTOLOGÍA

- **Criptografía** es la ciencia dedicada a cifrar y/ó proteger mensajes (información) por medio de un algoritmo usando claves.
- **Criptoanálisis** es la ciencia dedicada a descifrar ó romper los mensajes cifrados.

La base de la Criptología

- C. Shannon (1948, 1949).Teoría de la información.
- W. Diffie y M. Hellman (1976). Criptografía de Clave Pública.

- **Teoría de la Información.** Estadística, el seguro perfecto y la autenticidad perfecta.
- **Teoría de números.** Factorización, cuerpos finitos, LSFR,..
- **Teoría de la complejidad algorítmica** Tiempo de ejecución de programas y/o algoritmos.

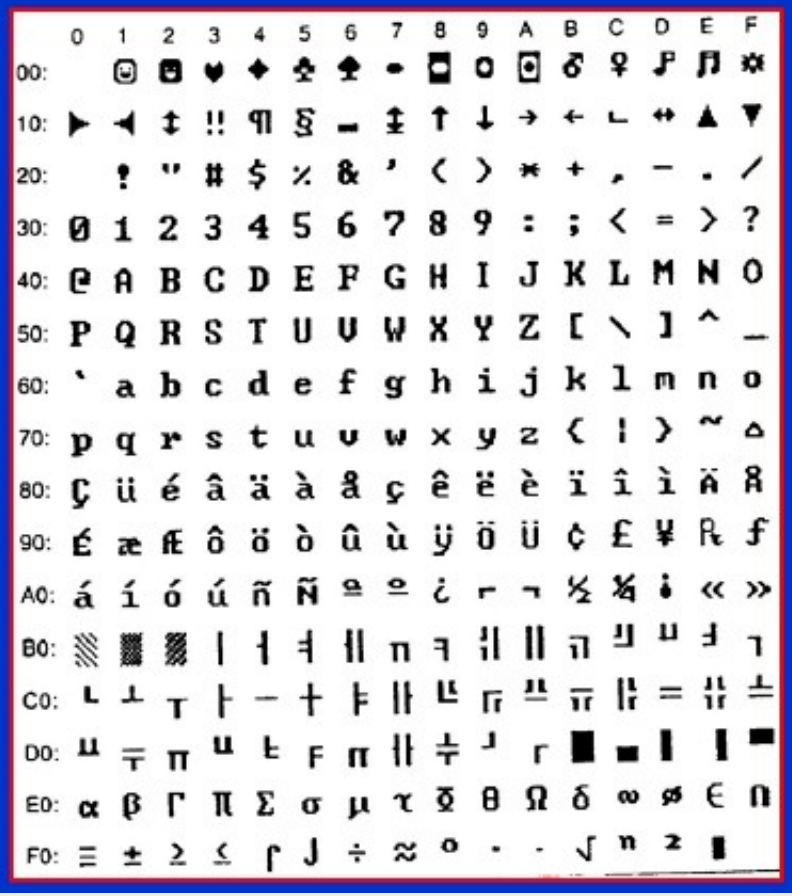
Codificación de la información.

- Entropía de la información.
- **Bit**=binary digit, **Byte**= 8 bits.
- Código Hexadecimal:

  $\mathbf{0} = 0000, \mathbf{1} = 0001, \ldots, \mathbf{9} = 1001, \mathbf{A} = 1010, \ldots, \mathbf{F} = 1111$

- ASCII (American Standard Code for Information Interchange)

Tabla de código ASCII extendido



00-0F: Valor decimal: 000-015

10-1F: Valor decimal: 016-031

20-2F: Valor decimal: 032-047

30-3F: Valor decimal: 048-063

40-4F: Valor decimal: 064-079

50-5F: Valor decimal: 080-095

60-6F: Valor decimal: 096-111

70-7F: Valor decimal: 112-127

80-8F: Valor decimal: 128-143

90-9F: Valor decimal: 144-159

A0-AF: Valor decimal: 160-175

B0-BF: Valor decimal: 176-191

C0-CF: Valor decimal: 192-207

D0-DF: Valor decimal: 208-223

E0-EF: Valor decimal: 224-239

F0-FF: Valor decimal: 240-255

Objetivos de la Codificación

- Comprimir la información: códigos compresores.
- Detectar y corregir fallos: códigos correctores de errores.
- Asegurar y garantizar la privacidad: códigos secretos.

Cifrador Julius Caesar

Fijar un número k (clave del método) y reemplazar cada letra por la que ocupa el lugar k posiciones a la derecha en el alfabeto. Si k=3:

$\mathbf{x}$ = "atacaremos al amanecer"

a	b	c	d	e	f	g	.	.	.	t	u	v	w	x	y	z
d	e	f	g	h	i	j	.	.	.	w	x	y	z	a	b	c

$\mathbf{y}$ = "dwdfduhov dñ dodphfhu"

Colossus:(antecesor del histórico ENIAC) ordenador electrónico con tubos de vacio. Alang Turing(Segunda Guerra Mundial)

Criptosistema

- $\mathcal{M}$: mensajes en claro o texto claro. Pueden ser cadenas de caracteres de un alfabeto (formado por bits, letras, elementos del conjunto $\mathbb{Z}_n$ —los enteros módulo n— etc.)

- $\mathcal{C}$: mensajes cifrados o texto cifrado.

- $\mathcal{K}$: espacio de claves. En Caesar es $\mathbb{Z}_{26}\backslash\{0\} = \{1, 2, .., 25\}$. En general, son cadenas de bits, vectores, elementos de $\mathbb{Z}_n$, etc.

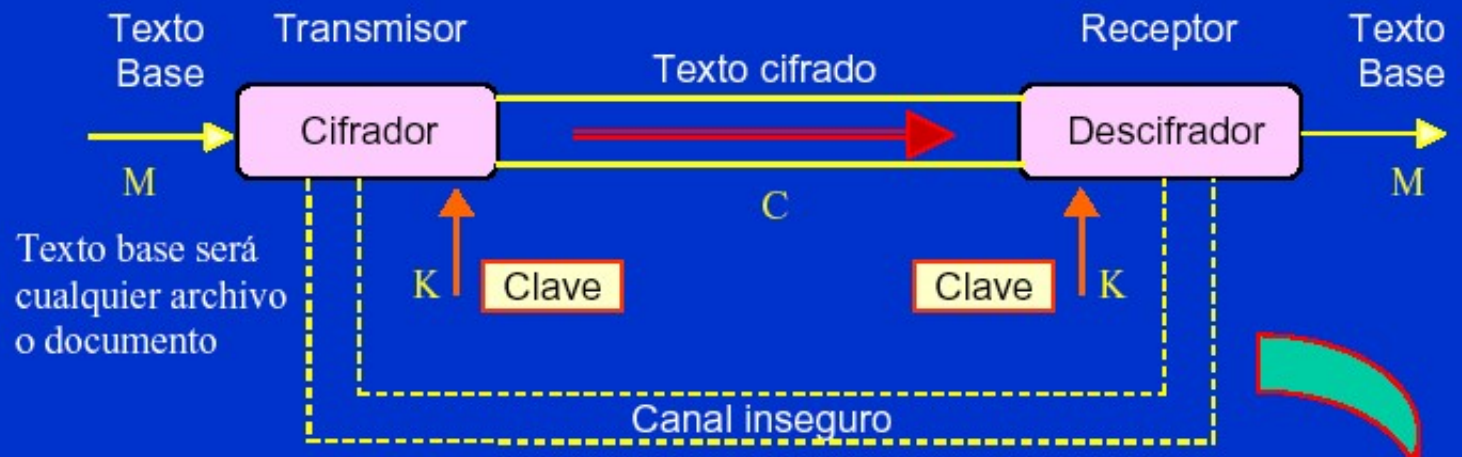
- $\mathcal{E} = \{E_k : \mathcal{M} \to \mathcal{C}, \quad k \in \mathcal{K}\}$ es el : metodos de cifrado . En César E_k es desplazar cada letra de ellas k lugares hacia la derecha.

- $\mathcal{D} = \{D_k : \mathcal{C} \to \mathcal{M}, \quad k \in \mathcal{K}\}$: métodos de descifrado. En Caesar D_k es desplazar cada letra de la cadena k posiciones a la izquierda.

Verificando que para cada $k \in \mathcal{K}$:

$$D_k(E_k(x)) = x, \quad \forall x \in \mathcal{M}.$$

Esquema de un criptosistema



Hablaremos entonces de:

- Un espacio de textos en claro M
- Un espacio de textos cifrados C
- Un espacio de claves K
- Unas transformaciones de cifrado $E_K(M)$
- Unas transformaciones de descifrado $D_K(C)$

Requisitos

- Rapidez, las transformaciones de cifrado y descifrado E_k y D_k tienen que evaluarse rápidamente, complejidad algebraica.

- Seguridad del sistema sólo debe depender del secreto de la clave $k \in \mathcal{K}$; es decir, supondremos que el criptoanalista conoce el método que se utiliza.

- Autenticidad : se debe evitar que un agente extraño pueda generar un mensaje cifrado que el receptor acepte como válido, habiendo suplantado así el papel del emisor.

Clave Privada y Clave Pública

- Clave privada, convencionales o simétricos. La clave K determina tanto el proceso de ciframiento E_K como el de desciframiento D_K.

- Clave pública o asimétricos. Caracterizados por el hecho de que el conocimiento de la función E_K no conlleva el de la función D_K o viceversa. Es decir, una de ellas puede ser revelada públicamente, sin peligro de que la otra pueda ser computada.

Criptoanálisis

Ataques:

- *Sólo a texto cifrado:* el criptoanalista conoce todos los mensajes cifrados (por ejemplo, porque ha "pinchado" el canal). Es el más habitual.

- *A texto claro conocido:* el criptoanalista conoce algunas parejas de mensajes en claro con sus cifrados. Esta situación es habitual cuando se conoce una parte del mensaje original (saludos, despedidas, fechas, "from", "to", etc.)

- *A texto claro escogido:* el criptoanalista puede cifrar una serie de textos en claro por él elegidos. Esta situación se presenta en las tarjetas de teléfonos móviles GSM.

- *A texto cifrado escogido:* el criptoanalista puede ejecutar (temporalmente) el algoritmo de descifrado. Su objetivo es averiguar la clave que lo determina; para, en un futuro (cuando quizá no tenga acceso al método), poder entender cualquier mensaje que intercepte.

Técnicas

- *Fuerza bruta:* se trata de encontrar la clave probando con todas las posibles. Aunque no es considerada como tal técnica,(capacidad de cálculo de los ordenadores y el crecimiento de su velocidad.)

- *Análisis de frecuencia:* se pueden analizar estadísticas de las frecuencias de los caracteres o de bloques de caracteres para romper los criptosistemas.

- *Diferencial:* se parte de pares de mensajes con diferencias mínimas (normalmente un bit) y se analizan las variaciones entre los correspondientes cifrados.

- *Lineal:* usa operaciones or-exclusiva entre algunos bits de texto claro y texto cifrado, con objeto de obtener un único bit. Si lo hacemos con muchos pares podemos obtener información para romper el método.

- *Algoritmos matemáticos:* se trata de diseñar algorimos eficientes computacionalmente para averiguar la clave. Tienen mayor interés en la criptografía pública.

Criptosistemas simétricos

El emisor y el receptor comparten una única clave K. Debemos suponer que hay un canal seguro, por medio del cual se transmite la clave secreta.

Cifrados de Substitución

Se acuerda entre el emisor y el receptor una substitución arbitraria de cada letra por otra (o por otro símbolo). Responde al concepto de la confusión , que trata de esconder la relación entre el texto claro, el texto cifrado y la clave

a	b	c	d	e	f	g	.	.	.	t	u	v	w	x	y	z
v	t	m	g	i	p	q	.	.	.	x	a	c	d	f	h	j

Hay 26! = 10.888.869450.418.352.160.768.000.000 claves distintas. Con los conocimientos estadísticos que se tienen sobre la distribución de las letras en cada idioma (Análisis de frecuencias) se rompe el sistema.

Otro refinamiento son los métodos de substitución polialfabéticos: se fija un entero m, y m tablas de substitución distintas. La primera se aplicará a las letras que ocupen las posiciones $1, m+1, 2m+1, \ldots$; la segunda, a los caracteres número $2, m+2, 2m+2, \ldots$, etc. (Sistema de Vigenère.)

Cifrados de transposición o permutación

El cifrado consiste en dividir el mensaje en bloques de n letras **(cifrado en bloque)** y efectuar la permutación o tranposición elegida (la clave) a cada bloque por separado. Se corresponde con el concepto de la difusión, que trata de propagar la información real en el mensaje cifrado.

x="ATACAREMOS AL AMANECER".

El tamaño de cada bloque es $n = 6$.

m_1="ATACAR"
m_2="EMOSAL"

$$m_3 = \text{“AMANEC”}$$
$$m_4 = \text{“ER”}$$

Fijamos la permutación $\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 3 & 5 & 1 & 2 & 6 \end{pmatrix}$, y la aplicamos a cada bloque de los anteriores:

$$\sigma(m_1) = \text{“CATAAR”}$$
$$\sigma(m_2) = \text{“SAMEOL”}$$
$$\sigma(m_3) = \text{“NEMAAC”}$$
$$\sigma(m_4) = \text{“TAREEN”}$$

El último bloque, se completo con letras al azar.

$$\mathbf{y} = \text{“CATAARSAMEOLNEMAACTAREEN”} \ .$$

El receptor del mensaje conoce la clave (n,σ) , sólo tiene que ejecutar el mismo algoritmo, pero utilizando la permutación inversa:

$$\sigma^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 5 & 2 & 1 & 3 & 6 \end{pmatrix}$$

Un ataque por "la fuerza bruta", comprobando las transposiciones de tamaño 2,3,. . . llegará a dar con nuestra clave en (a lo sumo) $\sum_{i=2}^{6} i! = 932$ comprobaciones.

Se debe escoger un tamaño para los bloques n mucho mayor.

A pesar del elevado número de claves, hay métodos eficaces: *frecuencias* utilizando propiedades *del grupo* Σ_n.

Códigos matriciales (L.S. Hill 1930)

Sean r y N dos enteros mayores que 2,

$\mathcal{M} = \mathcal{C} = \mathbb{Z}_N^r, \quad \mathcal{K} = GL_r(\mathbb{Z}_N) \times \mathbb{Z}_N^r,$

$GL_r(\mathbb{Z}_N) =$

$\{K \in M_r(\mathbb{Z}_N) \quad / \quad mcd(det(K), N) = 1\}$

Para cada $(K, a) \in \mathcal{K}$, definimos

$$E_K(x) = xK + a$$

y

$$D_K(y) = (y - a)K^{-1}$$

Se trata de un método de substitución donde los elementos del alfabeto son bloques de letras. La multiplicación de matrices rompe la estructural gramatical.

Son muy vulnerables a ataques *a texto claro conocido* , porque dadas $r+1$ parejas (x,y) de texto original con su correspondiente cifrado (bajo condiciones de independencia lineal), se llega a un sistema de conguencias lineales.

Los cifrados en flujo

Usan técnicas de encadenamiento entre bloques, de manera que el cifrado de cada uno dependa del cifrado anterior.

Similarmente, se puede hacer que el cifrado de bloques idénticos no sea el mismo mediante la generación de números pseudoaleatorios de bits, $K = \{k_i, i = 1, \ldots\}$ y su combinación con el criptotexto correspondiente:

$$y_i = x_i \oplus k_i.$$

El secreto perfecto

La interceptación del mensaje cifrado no proporciona ninguna información sobre el original. Con el concepto de entropía, Shannon demostró que el tamaño de la clave debe ser tan grande como el mensaje, además de que sólo se puede utilizar una vez: son los denominados **one-time pads**.

`G.S Vernam, ATT (1926).`

Los one-time pads no son operativos en la práctica.

Redes Substitución-Permutación

Una red SPN es un cifrador *iterado*. Se divide el mensaje en bloques de bits y se aplica un número N_r rondas o vueltas de substituciones y permutaciones a cada bloque.

Cada cifrador requiere la función ronda ; la clave K, que generalmente es una cadena aleatoria de bits y una función de expansión EK de la clave K, proporcionando una lista de claves

$$EK(K) = (K^{(1)}, \ldots, K^{(N_r+1)})$$

y que será construida con un algoritmo público.

Intervienen dos permutaciones: π_S, denominada S-caja y, π_P la cual permuta los bits de cada bloque. Sean l y m enteros positivos.

Dadas las permutaciones

$$\pi_S : \{0,1\}^l \to \{0,1\}^l$$

y

$$\pi_P : \{1,\ldots,lm\} \to \{1,\ldots,lm\}.$$

Dado $\mathbf{x} = (x_1, \ldots, x_{lm})$ de longitud lm. Vemos $\mathbf{x}$ como una concatenación de m cadenas de bits, cada una de ellas con l bits,

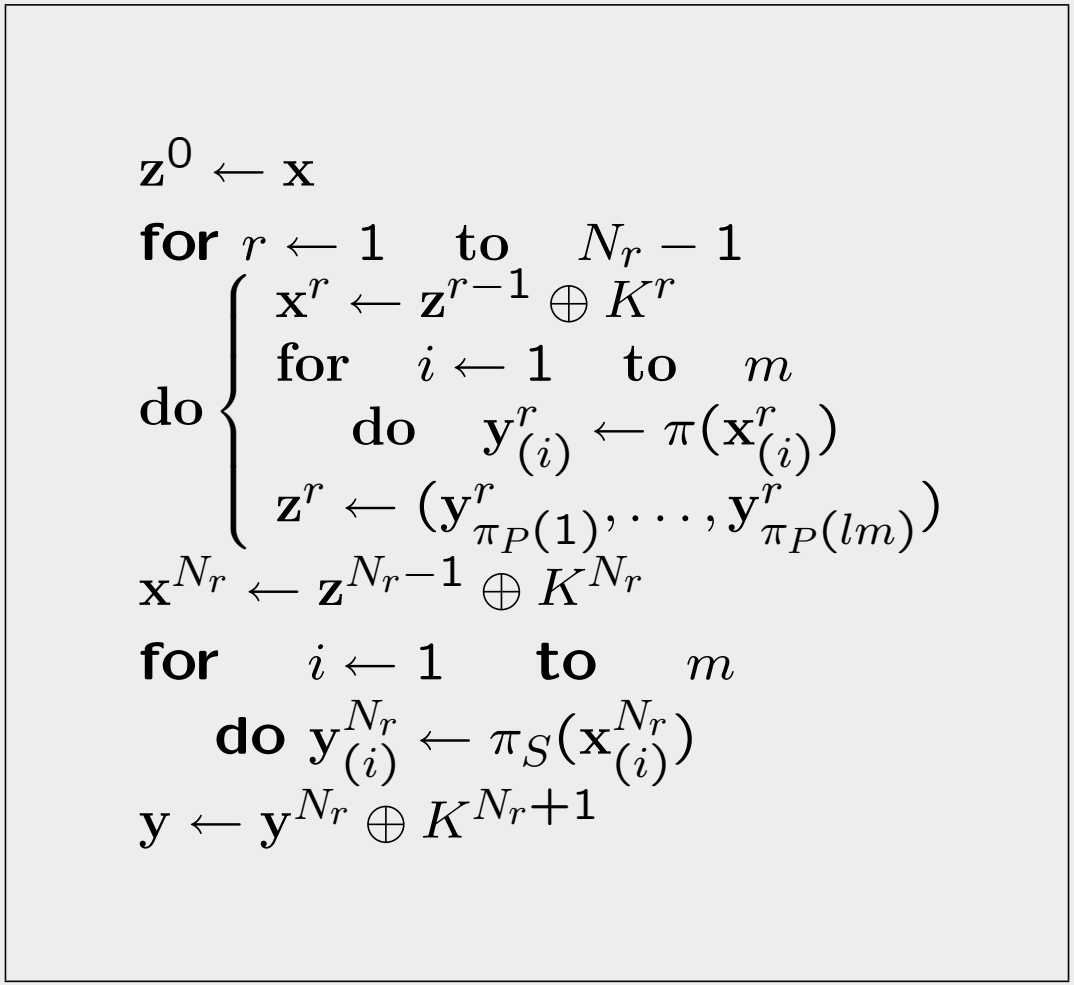
$$\mathbf{x} = (x_{(1)}, \ldots, x_{(m)}), \quad x_{(i)} = (x_{(i-1)l+1}, \ldots, x_{il}).$$

Dada la salida

$$(K^1, \ldots, K^{N_r+1})$$

de la función EK de la clave $K \in \{0,1\}^{lm}$.

$SPN(\mathbf{x}, \pi_S, \pi_P, (K^1, \ldots, K^{N_r+1}) = \mathbf{y}$:


$\mathbf{z}^0 \leftarrow \mathbf{x}$
for $r \leftarrow 1$ to $N_r - 1$
do
$\mathbf{x}^r \leftarrow \mathbf{z}^{r-1} \oplus K^r$
for $i \leftarrow 1$ to m
do $\mathbf{y}^r_{(i)} \leftarrow \pi(\mathbf{x}^r_{(i)})$
$\mathbf{z}^r \leftarrow (\mathbf{y}^r_{\pi_P(1)}, \ldots, \mathbf{y}^r_{\pi_P(lm)})$
$\mathbf{x}^{N_r} \leftarrow \mathbf{z}^{N_r-1} \oplus K^{N_r}$
for $i \leftarrow 1$ to m
do $\mathbf{y}^{N_r}_{(i)} \leftarrow \pi_S(\mathbf{x}^{N_r}_{(i)})$
$\mathbf{y} \leftarrow \mathbf{y}^{N_r} \oplus K^{N_r+1}$

Redes Feistel

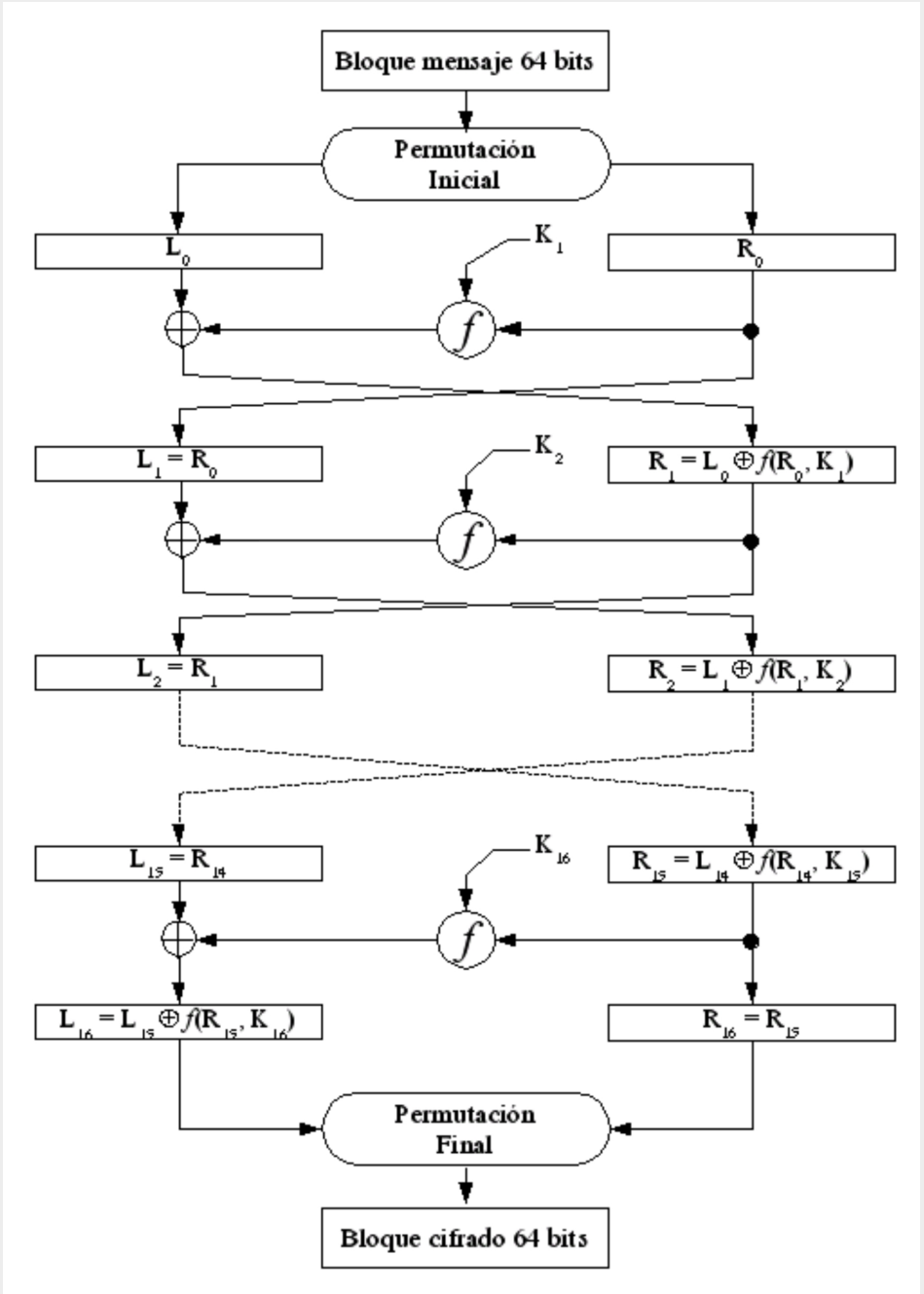
Muchos de los cifradores producto tienen en común que dividen cada bloque S_i en la ronda i-ésima en dos mitades: $S_i = L_i R_i$ y aplican una red tipo SPN, en el que la salida de cada ronda es utilizada como entrada para la siguiente:

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i).$$

Esta clase de estructuras fue introducida por `H. Feistel, 1973` y es utilizada por varios algoritmos de cifrado en bloque, como DES, CAST, FEAL, Lucifer, etcetera.

DES(Data Encryption Standard)

El DES (IBM 1975) cifra y descrifra bloques de 64 bits y lo somete a 16 rondas, con una clave 64 bits (56 bits reales y 8 bits de control de paridad).


Bloque mensaje 64 bits
Permutación Inicial
L0
R0
K1
f
L1 = R0
R1 = L0 ⊕ f(R0, K1)
K2
L2 = R1
R2 = L1 ⊕ f(R1, K2)
L15 = R14
R15 = L14 ⊕ f(R14, K15)
K16
L16 = L15 ⊕ f(R15, K16)
R16 = R15
Permutación Final
Bloque cifrado 64 bits

DES(Data Encryption Standard)

- La función f es una permutación de expansión, convirtiendo un bloque de 32 bits en uno de 48 bits.

- A continuación realiza un or-exclusiva con el valor de la función clave K^i y aplica una substitución (las famosas ocho S-cajas)

- Validez como uso estandard entre 10 y 15 años.

- Critica a la construcción de las S-cajas.

LAS S-BOX

14 04 13 01 02 15 11 08 03 10 06 12 05 09 00 07
00 15 07 04 14 02 13 01 10 06 12 11 09 05 03 08
04 01 14 08 13 06 02 11 15 12 09 07 03 10 05 00
15 12 08 02 04 09 01 07 05 11 03 14 10 00 06 13
S-BOX 1

15 01 08 14 06 11 03 04 09 07 02 13 12 00 05 10
03 13 04 07 15 02 08 14 12 00 01 10 06 09 11 05
00 14 07 11 10 04 13 01 04 08 12 06 09 03 02 15
13 08 10 01 03 15 04 02 11 06 07 12 00 05 14 09
S-BOX 2

10 00 09 14 06 03 15 05 01 13 12 07 11 04 02 08
13 07 00 09 03 04 06 10 02 08 05 14 12 11 15 01
13 06 04 09 08 15 03 00 11 01 02 12 05 10 14 07
01 10 13 00 06 09 08 07 04 15 14 03 11 05 02 12
- S-Box 3

07 13 14 03 00 06 09 10 01 02 08 05 11 12 04 15
13 08 11 05 06 15 00 03 04 07 02 12 01 10 14 09
10 06 09 00 12 11 07 13 15 01 03 14 05 02 08 04
03 15 00 06 10 01 13 08 09 04 05 11 12 07 02 14
S-BOX 4

02 12 04 01 07 10 11 06 08 05 03 15 13 00 14 09
14 11 02 12 04 07 13 01 05 00 15 10 03 09 08 06
04 02 01 11 10 13 07 08 15 09 12 05 06 03 00 14
11 08 12 07 01 14 02 13 06 15 00 09 10 04 05 03
S-BOX 5

12 01 10 15 09 02 06 08 00 13 03 04 14 07 05 11
10 15 04 02 07 12 09 05 06 01 13 14 00 11 03 08
09 14 15 05 02 08 12 03 07 00 04 10 01 13 11 06
04 03 02 12 09 05 15 10 11 14 01 07 06 00 08 13
S-BOX 6

04 11 02 14 15 00 08 13 03 12 09 07 05 10 06 01
13 00 11 07 04 09 01 10 14 03 05 12 02 15 08 06
01 04 11 13 12 03 07 14 10 15 06 08 00 05 09 02

Análisis diferencial (E.Biham y A. Shamir, 1990).

- Critica al tamaño de la clave. DES Challenge III y DES-Cracker junto con otros 100.000 ordenadores a traves de internet, rompieron DES en 22 horas (1999).

- Ataques: diferencial y lineal (`Matsui 1994`). Este permite romper el DES con ataques a texto claro conocido, usando 2^{43} pares, en 40 días de trabajo.

- Triple DES consiste en actuar tres veces el DES, con tres claves distintas.

AES(Advanced Encryption Standard)

Un poco de historia

En 1997 Instituto Nacional de Estándares y Tecnología (NIST) comenzó el proceso de elegir un estándar de cifrado avanzado AES, con una convocatoria pública a la comunidad científica.

- El algoritmo debe ser público. Gratuito .
- De cifrado en bloque simétrico y diseñado de forma que se permita aumentar la longitud.

- Debe poderse implementar tanto en hardware como en software.

- Serán evaluados de acuerdo con: seguridad, eficiencia computacional y requisitos de memoria, adecuación hardware y software, simplicidad de diseño y flexibilidad, y requisitos de licencia.

- Deben soportar cifrados con una longitud de bloque de 128 bits y una longitud de clave de 128, 192 y 256 bits.

Los cinco finalistas

De las 21 propuestas, sólo 15 cumplían las exigencias de la convocatoria.

En abril de 2000 se celebró el AES2 en Nueva York, donde se presentaron nuevos estudios de evaluación y criptoanálisis de los últimos cinco candidatos.

El 2 de octubre de 2000 se anunció el algoritmo ganador: RIJNDAEL, propuesto por los belgas `Vincent Rijmen y Joan Daemen.`

Los motivos para seleccionarle fue su buena combinación de seguridad-velocidad-eficiencia, sencillez y flexibilidad.

- 1. RIJNDAEL, 86 votos.
- 2. SERPERNT, 59 votos.
- 3. TWOFISH, 31 votos.
- 4. RC6, 23 votos.
- 5. MARS, 13 votos.

Las matemáticas

Varias operaciones en Rijndael estan definidas al nivel de bytes, entidades de 8 bits, tratados como elementos del cuerpo finito $\mathbb{F}_{2^8}$, es decir un byte $(b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0)$ es considerado como un polinomio $f(x)$ con coeficientes en $\mathbb{F}_2 = Z_2 = \{0, 1\}$:

$$b_7 x^7 + b_6 x^6 + b_5 x^5 + b_4 x^4 b_3 x^3 + b_2 x^2 + b_1 x + b_0.$$

$$\Updownarrow$$

$$(b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0).$$

En notación hexadecimal:

$$(01011110) = x^6 + x^4 + x^3 + x^2 + x = \{5E\}.$$

Sumar dos polinomios es sumar sus coeficientes módulo 2, (or-exclusiva $\oplus$) de dos bytes:

$$(x^6+x^4+x^3+x^2+x)+(x^7+x^4+x^3+1)=x^7+x^6+x^2+x+1$$
$$\Updownarrow$$
$$(01011110)\oplus(10011001)=(11000111).$$

La multiplicación en el cuerpo $\mathbb{F}_{2^8}$ se corresponde con la multiplicación de polinomios módulo un polinomio irreducible de grado 8. El polinomio propuesto por Rijndael es :

$$m(x)=x^8+x^4+x^3+x+1.$$

Multiplicar dos polinomios $f(x)$ y $g(x)$ módulo el polinomio $m(x)$, consiste en tomar el resto de la división de $f(x)g(x)$ entre $m(x)$.

$$(x^6 + x^4 + x^3 + x^2 + x) \times (x^7 + x^4 + x^3 + 1) \equiv$$

$$x^{13} + x^{11} + x^6 + x^3 + x^2 + x \equiv$$

$$x^7 + x^6 + x^4 + x^3 + x + 1 \bmod m(x).$$

Cada elemento $f(x)$ tiene inverso $f(x)^{-1}$ en el cuerpo $\mathbb{F}_{2^8} = \mathbb{F}_2[x]/m(x)$. Para determinar este el inverso, se utiliza el famoso algoritmo extendido de Euclides a los polinomios $f(x)$ y $m(x)$ obteniendo

$$f(x)g(x) + m(x)n(x) = 1,$$

$$f(x)g(x) = 1 \bmod m(x), \quad g(x) = f(x)^{-1}.$$

La multiplicación por x es muy simple y útil: todas las multiplicaciones entre polinomios pueden ser expresadas por éstas y por or-exclusivas. Por ejemplo,

$$f(x)(x^2+1) = x(x(f(x))) \oplus f(x).$$

Si $b_7 = 0$, la multiplicación es un desplazamiento a la izquierda: $x(01101110) = (11011100)$.

En cambio si $b_7 = 1$, es un desplazamiento a la izquierda seguido de una or-exclusiva con $(00011011) = \{1B\}$,

$$x(10101100) = (01011001) \oplus (00011011) = (01000001).$$

Una palabra es un fila de 4 bytes, que es tratada como un polinomio de grado 3 en $\mathbb{F}_{2^8}$:

$$[a_3 a_2 a_1 a_0] \mapsto a_3 y^3 + a_2 y^2 + a_1 y + a_0,$$

donde los a_i son bytes.

Sumar dos palabras se suman los coeficientes (or-exclusivo).

La multiplicación de dos palabras, lo reducimos módulo un polinomio de grado 4:

$$y^4 + 1.$$

Este polinomio tiene propiedad interesante:

$$y^i \bmod (y^4 + 1) = y^{i \bmod 4},$$

permitiendo realizar la multiplicación, sólo con or-exclusiva y desplazamientos.

$$a(y) \otimes b(y) = a(y)(y)b \bmod y^4 + 1.$$

$y^4 + 1 = (y^2 + 1)^2$ no es irreducible, se elige un polinomio fijo $c(y)$, que es primo con $y^4 + 1$ y por lo tanto es inversible módulo $y^4 + 1$:

$$c(y) = \{03\}y^3 + \{01\}y^2 + \{01\}y + \{02\}.$$

El inverso $c(y)$ es $d(y)$:

$$d(y) = \{0E\} + \{09\}y + \{0D\}y^2 + \{0B\}y^3.$$

El Algoritmo

Rijndael cifra bloques de 128, 192 ó 256 bits (4,6 ó 8 palabras) con claves, también, de 128, 192 ó 256 bits. El estándar AES solo permite cifrar y descifrar bloques de 128 bits.

Al número de palabras del bloque es N_b y N_k el número de palabras de la clave. Para cifrar un bloque se realizan N_r iteraciones o vueltas que depende de N_b y N_k:

N_r	$N_b = 4$	$N_b = 6$	$N_b = 8$
$N_k = 4$	10	12	14
$N_k = 6$	12	12	14
$N_k = 8$	14	14	14

Internamente, Rijndael utiliza estados S, que son matrices $4 \times N_b$ (de cuatro filas por N_b columnas), donde cada elemento de la matriz es un byte. La *función input* al comienzo que transforma un bloque en un estado y la *función output* al final que transforma un estado en un bloque.

Cada byte de un bloque esta numerado desde 0 a $4(N_b - 1)$ la función input coloca al elemento de índice n en la fila $i = n \bmod 4, i = 0, 1, 2, 3$. Y en la columna $j = \lfloor n/4j \rfloor, j = 0, \ldots, N_b - 1$ de un estado. A su vez la transformación output inserta al elemento de la fila i y la columna j de un estado en la posición $n = 4i + j$ de un bloque.

Un bloque al pasar a un estado, cada palabra es una columna, y el índice i denota un byte dentro de una palabra y j indica una palabra dentro de un bloque.

Para cifrar un bloque con una clave se hacen N_r rondas, en cada una de ellas intervienen 4 transformaciones o funciones invertibles:

- **SB** SubBytes (substitución de bytes),
- **SR** ShiftRows (desplazamiento de filas),
- **MC** MixColumns (mezcla de columnas) y

- **ARK** AddRoundKey (añadir elementos a la clave).

El Cifrado

Las transformaciones lineales MC y SR proporcionan la difusión, la transformación SB es no lineal, (lo parejo a las S-cajas del DES) que junto a la ARK (un or-exclusiva entre el estado intermedio y la subclave intermedia $EK(i)$ en cada ronda i) proporcionan dosis de confusión al criptosistema.

- $S := input(\mathbf{x}); EK(K);$
- $S_0 := ARK(S, EK(0));$
- $S_i := ARK(MC(SR(SB(S_{i-1}))), EK(i)),$
  $i = 1, \ldots, N_r - 1;$
- $S_{N_r} := ARK(SR(SB(S_{N_r-1})));$
- $\mathbf{y} := output(S_{N_r}).$

Donde **x** es el bloque en texto claro e **y**, es el bloque en texto cifrado.

Substitución de Bytes SB

SB tiene como entrada y como salida un estado. Actúa independientemente sobre cada byte del estado mediante otra transformación F de modo que

$$F(\alpha) = \alpha', \quad \alpha' = (b'_7 b'_6 b'_5 b'_4 b'_3 b'_2 b'_1 b'_0).$$

Si $\alpha \neq \{00\}$, pongamos $\alpha^{-1} = (x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0)$, el inverso de α en el cuerpo $\mathbb{F}_{2^8}$; si $\alpha = \{00\}$, pongamos $\alpha^{-1} = \{00\}$. Sea $\mathbf{A}$ la matriz circulante definida por el vector-byte $(10001111) = \{8F\}$,

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$F(\alpha) = \mathbf{A}(x_0x_1x_2x_3x_4x_5x_6x_7)^t + \{C6\})^t =$$

$$(b'_0b'_1b'_2b'_3b'_4b'_5b'_6b'_7)^t.$$

La matriz $\mathbf{A}$ es una matriz inversible en $\mathbb{F}_2$:

$$F(\alpha') = (\mathbf{A}^{-1}[(b'_0b'_1b'_2b'_3b'_4b'_5b'_6b'_7)^t + \{C6\}^t])^{-1} = \alpha.$$

Desplazamiento de Filas SR

Esta operación actúa sobre las filas mediante desplazamientos cíclicos a la izquierda, la primera fila no es desplazada, a la segunda fila se le aplican c_1 desplazamientos, c_2 a la tercera, y c_3 la cuarta. Donde c_1, c_2 y c_3 dependen de N_b:

N_b	c_1	c_2	c_3
4	1	2	3
6	1	2	3
8	1	3	4

La inversa de SR aplica a la fila i un total de $N_b - c_i$ desplazamientos cíclicos a la izquierda con $c_0 = 0$.

Mezcla de Columnas MC

MC actua sobre cada columna $(a_0^j, a_1^j, a_2^j, a_3^j)$ de un estado.

$$MC(a_0^j, a_1^j, a_2^j, a_3^j) = (b_0^j, b_1^j, b_2^j, b_3^j) \text{ donde,}$$

$$(a_0^j + a_1^j y + a_2^j y^2 + a_3^j y^3) \otimes c(y) =$$

$$b_0^j + b_1^j y + b_2^j y^2 + b_3^j y^3.$$

La transformación inversa de MC, es análoga a esta, cambiando $c(y)$ por su inverso $d(y)$, es decir,

$$(b_0^j + b_1^j y + b_2^j y^2 + b_3^j y^3) \otimes d(y) =$$

$$a_0^j + a_1^j y + a_2^j y^2 + a_3^j y^3.$$

Añadir elementos de la clave ARK

La función ARK tiene como entrada un estado S y una fila de N_b palabras y como salida otro estado S'. La fila de palabras de entrada lo denotamos por $EK(i) = (w(4i), w(4i+1), \ldots, w(4i+N_b-1))$ donde i representa la iteración en que nos encontramos y cada w es una palabra. La transformación ARK es una or-exclusiva de la palabra correspondiente a la columna j del estado S la palabra $w(4i+j), j = 0, \ldots, N_b$ transformándolo en el nuevo estado S'.

La función inversa de ARK es ella misma, puesto que es una or-exclusivo.

Las $EK(i)$ son palabras que se obtienen mediante la función expansión de clave EK de la siguiente forma:

Expansión de clave EK

La función EK tiene como entrada la clave K y como salida $N_b(N_r + 1)$ palabras:

$$EK(K) = (w(0), w(1), \ldots, w(N_b(N_r + 1)).$$

Las primeras N_k palabras se corresponden con las de la clave K y las siguientes:

- Si $i \bmod N_k = 0$,

  $$w(i) = SW(RW(w(i-1))) \oplus Rcon(i/N_k) \oplus w(i - N_k).$$

- Si i mod $N_k = 4$ y $N_k > 6$, $w(i) = SW(w(i-1)) \oplus w(i-N_k)$.

- El resto, $w(i) = w(i-1) \oplus w(i-N_k)$

donde SW actúa igual que la función F y

$$RW(a_0a_1a_2a_3) = (a_1a_2a_3a_0)).$$

La función $Rcon$ tiene la entrada un entero n y salida una palabra: $Rcon(n) = (a_0a_1a_2a_3)$

$$a_1 = a_2 = a_3 = \{00\}, a_0 = \{02\}^{-1}.$$

El Descifrado

Es muy similar al del cifrado, sólo es necesario realizar todas las operaciones en orden inverso y usar la generación de la clave, también, en orden inverso. Además, se deben aplicar las transformaciones inversas, ya descritas en el los pasos anteriores.

- $S := input(\mathbf{y}); EK(K);$
- $S_{N_r} := SB^{-1}(SR^{-1}(ARK(S, EK(N_r))));$
- $S_{N_r-i} := SB^{-1}(SR^{-1}(MC^{-1}(ARK$
  $(S_{N_r+1-i}, EK(N_{r+1-i})), i = 1, \ldots, N_r - 1;$
- $S_0 := ARK(S_1, EK(0));$
- $\mathbf{x} := output(S_0).$

Donde SB^{-1}, SR^{-1} y MC^{-1} son las transformaciones inversas de SB, SR y MC, respectivamente.

Por cuestiones de eficiencia en la implementación, puede ser adaptado a un criptosistema cifrado-descifrado.

Análisis de seguridad

La S-caja SB, fue diseñada frente ataques de tipo criptoanálisis diferencial y lineal.

- Invertibles.
- Minimizar la correlación entre las combinaciones lineales de bits a la entrada con las combinaciones de bits a la salida.
- Dificultar manipulaciones algebraicas, para prevenir a ataques de interpolación.

El número a desplazar en la transformación SR, fueron elegidos para ofrecer resistencia contra ataques "truncated differentials".

El número de rondas fue determinado buscando el mínimo número de vueltas necesarias para ofrecer un margen de seguridad alto. Para un bloque y clave de 128 bits se utilizan 10 vueltas, puesto que no se han encontrado atajos en ataques para versiones reducidas con mas de 6 vueltas.

Modos de operación: DES, AES

- ECB (Electronic Codebook Mode), para mensajes cortos.
- CBC (Cipher Block Chaining Mode) para mensajes largos.
- CFB (Cipher Feedback Mode) para cifrar bit por bit ó byte por byte.
- OFB (Output Feedback Mode) el mismo uso pero evitando propagación.

CBC(Cipher Block Chaining)

El modo CBC hace depender el bloque i-ésimo del texto cifrado/descifrad
del (i-1)-ésimo:

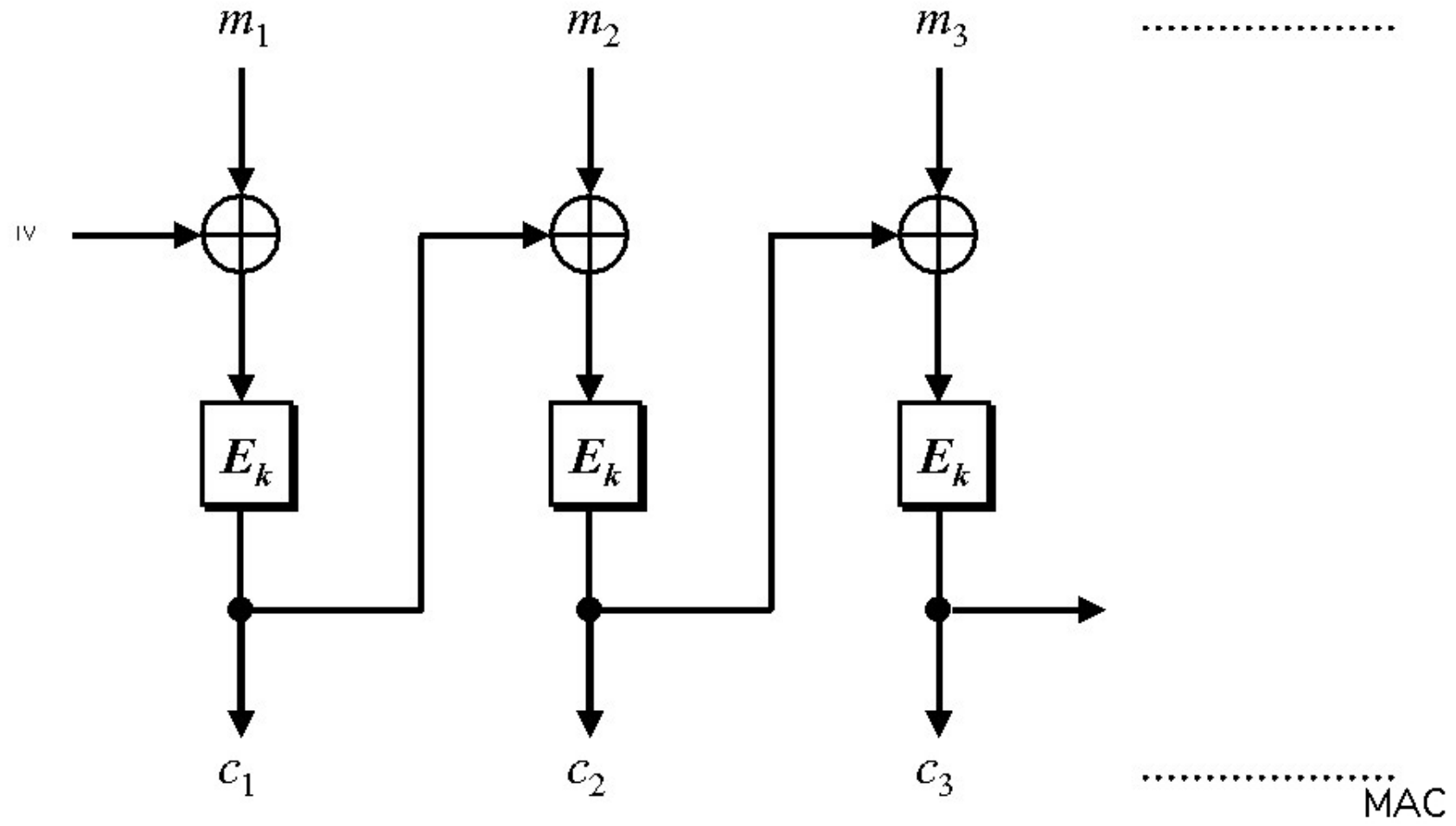
$$c_i = E_K(m_i \oplus c_{i-1}), \quad m_i = D_K(c_i) \oplus c_{i-1}.$$

Partimos de un vector inicial fijo IV.

De este modo, se produce un

encadenamiento(chaining)

entre los distintos bloques, y el resultado de cifrar cada uno de ellos depende de todos los anteriores.


m_1
m_2
m_3
IV
E_k
E_k
E_k
c_1
c_2
c_3
MAC

El último bloque firma digital o checksum del resto:

MAC (Message Authentication Code)

permitiendo certificar que no ha sido alterado:

autentificación.

Criptosistemas

asimétricos

Los sistemas convencionale han ido mostrando puntos de debilidad y no responden a las necesidades actuales: el problema de la distribución de claves a través de una canal seguro, la autentificación, o el diseño de redes que comuniquen a muchos usuarios, y que permitan a cada dos de ellos compartir información sin que pueda acceder a ella el resto.

Whitfield Diffie y Martin Hellman (1976)

Consideramos un conjunto amplio de usuarios, cada uno X dispondrá ahora de dos claves: una secreta D_{K_Y};, que deberá conservar y otra pública E_{K_Y}, que debe difundir en la red.

Autoridad Certificadora (CERES, VERISIGN,...)

$$\{(I_A, E_{K_A}), \quad (I_B, E_{K_B}), \quad (I_C, E_{K_C}), \quad \ldots.\}$$

Cuando el usuario A quiera enviar un mensaje x a otro miembro B de la red, basta con que localice en el directorio su clave pública, con la que será capaz de ejecutar el algoritmo de cifrado $E_{K_B}(x)$, y enviar el mensaje a B.

Condiciones Diffie-Hellman

- La obtención de D_K debe ser imposible a partir de E_K y de un texto cifrado.

- El cálculo de E_K y D_K debe ser sencillo.

Función unidireccional:

Es una transformación $f : x \to f(x)$ que sea fácil de calcular, pero que haga impracticable (computacionalmente) la determinación a partir de $f(x)$ de la antiimagen x.

El logaritmo discreto:

Dado un primo p, y un elemento primitivo $a \in \mathbb{Z}_p$, *se tiene que:* $\{a^x \bmod p / 0 < x < p\} = \mathbb{Z}_p^*$. *Se puede calcular con un algoritmo eficaz la exponencial discreta* $a^x \bmod p$, *pero el cálculo inverso, esto es, encontrar* x *tal que* $a^x \bmod p = b$, *conocidos* b *y la base* a*) es computacionalmente difícil.*

Criptografía de clave pública:

La transformación inversa $f(x) \rightarrow x$ puede ser factible si se conoce un dato adicional, al que llamaremos *trampa*.

El Gamal

Sea p un primo y $a \in \mathbb{Z}_p$ un elemento primitivo

Ponemos $\mathcal{M} = \mathcal{C} = \mathbb{Z}_p \backslash \{0, 1\} = \{2, \ldots, p-1\}$

$\mathcal{K} = \{(k, a^k)/k \in \mathbb{Z}_p \backslash \{0, 1\}\}$

Cada usuario escoge su clave privada

$Priv_U \in \mathbb{Z}_p \backslash \{0, 1\}$

Su clave pública es $Pub_U = a^{Priv_U}$

- Si A quiere mandar un mensaje x a B; escoge un entero f y calcula a^f. Después, transmite a B el par: $(a^f, x(Pub_U)^f)$
- Una vez que B recibe el cifrado, como conoce $Priv_B$, puede calcular $(a^f)^{Priv_B} = (Pub_U)^f$. Entonces,
  $x = ((a^f)^{Priv_B})^{-1} \times x(Pub_U)^f$
  (Todos los cálculos se hacen en $\mathbb{Z}_p$)

RSA

Rivest, Shamir, Adleman (1978)

Cada usuario elige dos números primos p y q, calcula su producto $n = pq$ y el indicador de Euler $\phi(n) = (p-1)(q-1)$. Después, toma un número e tal que $0 < e < \phi(n)$ y $\text{mcd}(e, \phi(n)) = 1$, por lo que existe un inverso $d \in \mathbb{Z}_{\phi(n)}$, esto es: $ed \equiv 1 \bmod \phi(n)$. La clave que públicará es (n, e), y como clave privada d, p y q.

Sea $n = pq$, donde p y q numeros primos.
Sea $\mathcal{M} = \mathcal{C} = \mathbb{Z}_n \backslash \{0, 1\}$, definimos:

$$\mathcal{K} = \{(n, p, q, e, d), \quad ed \equiv 1 \bmod \phi(n)\}.$$

Para $K = (n, p, q, e, d)$, se define

$$E_K(x) = x^e \bmod n$$

y

$$D_K(y) = y^d \bmod n$$

(n, e) son públicos, y p, q y d privados

Un Ejemplo

Alicia quiere enviar un mensaje a Bernardo

Bernardo elige los primos $p = 127$ y $q = 223$, entonces $n = 28321$ y $\phi(n) = 126 \times 222 = 27972$.

Bernardo elige $e = 3025$ con $mcd(e, n) = 1$ y computa, utilizando el eficiente algoritmo extendido de Euclides, $d = 15877$, verificando que

$$e \times d \equiv 1 \bmod 27972.$$

En el directorio público Bernardo pone su clave pública $n = 28321$ y $e = 3025$.

Alicia quiere enviar el texto claro $x = 4599$ a Bernardo, entonces ella computa

$$4599^{3025} \bmod 28321 = 12346$$

y envia por el canal 12346.

Cuando Bernardo recibe el texto cifrado 12346 usa su clave privada $d = 15877$ y computa

$$12346^{15877} \bmod 28321 = 4599.$$

Los primos p y q.

- Los primos constituyen la base de la aritmética.
- Existen infinitos números primos.
- Los primos gemelos y los primos consecutivos muy separados.
- Teorema de los números primos: $prob(a = primo)$ es como $1/ln(a)$.

Los primos se deben elegir con unos 100 cifras decimales.

Test de Primalidad.

Es un algoritmo que permite decidir si un número natural es primo o compuesto. Los podemos clasificar:

Deterministicos

Probabilisticos

Se parte de un número pseudoaleatorio e impar q de 100 cifras, y utilizando tests de primalidad se comprueba si es primo o no; en caso negativo se intenta con $q+2, q+4, \ldots$, y así sucesivamente, hasta que se llega a uno que lo sea.

Algunos Test de Primalidad

- Criba de Eratóstenes.
- Solvay-Strassen 1977 (Ley de reciprocidad cuadrática).
- Miller-Rabin 1976-1980.
- Agrawal,Nayal, Saxena 2003.
- Otros,... (para ciertos enteros,...)

Seguridad del RSA. Algoritmos de factorización

El cálculo de $\phi(n)$ es equivalente a factorizar n

- Metodo de Fermat.
- Algoritmo de Pollard
- Dixon, Métodos de Sieve.
- Curvas elípticas.

Otros ataques : shortest vector problem L^3, iteración,....

AUTENTICIDAD. FIRMA

El remitente A, que desea autentificar un mensaje M, enviado a B, $E_A(D_A(M) = M$

La autenticidad del remitente es evidente, porque solo A ha podido cifrar tal mensaje. La integridad del mensaje queda asimismo garantizada, la probabilidad de que un falso mensaje se descodifique correctamente con la clave pública de A es prácticamente nula.

Una **función hash** es una función de una vía, públicamente conocida (a diferencia del MAC no utiliza ninguna llave) que produce, a partir de un mensaje M de longitud variable, un bloque $H(M)$ de longitud fija. **MD5, SHA-1**.

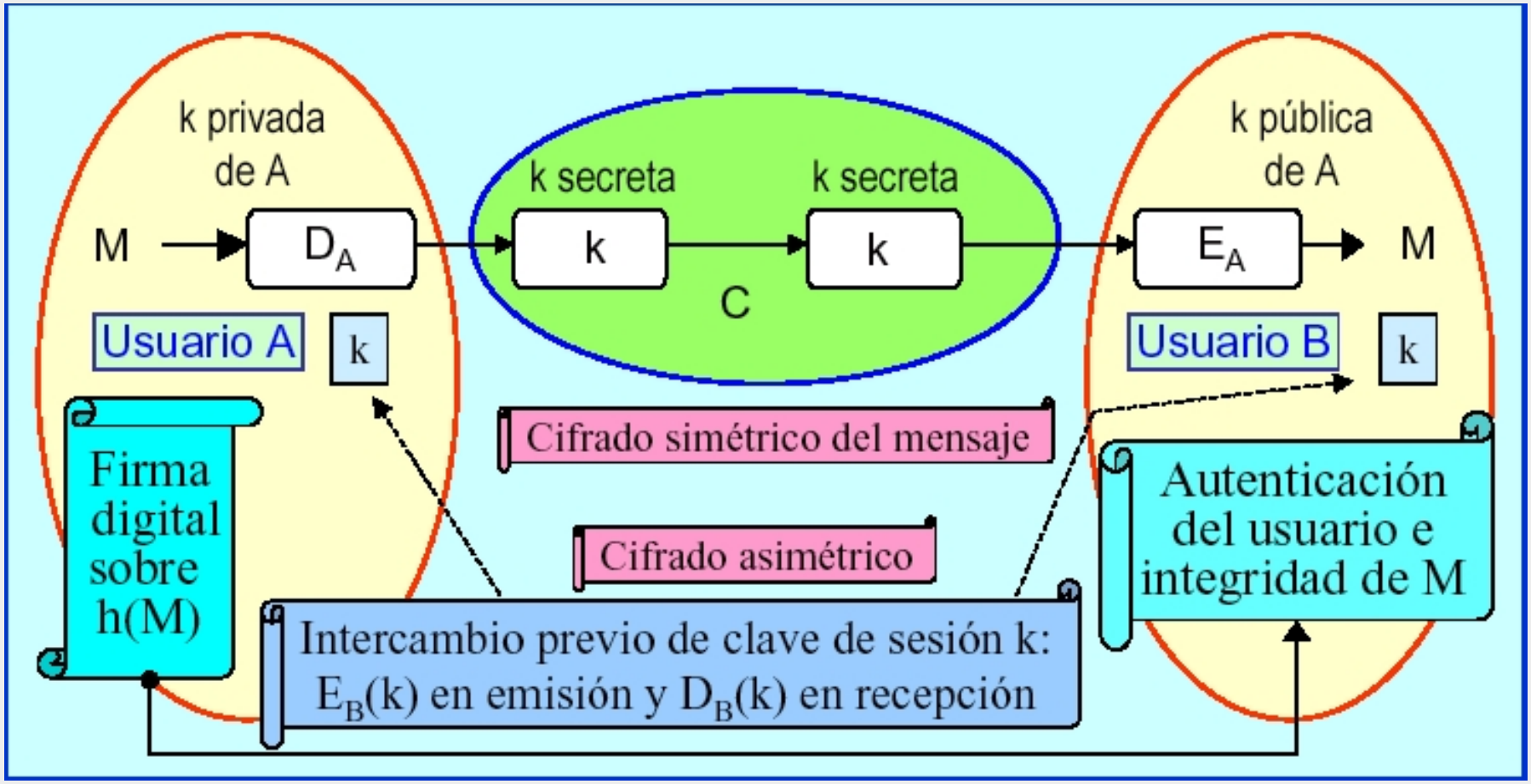
Implementaciones y Software

La implementación del sistema RSA resulta más complicada que la de los métodos simétricos como DES y AES. Si bien estos últimos tienen una descripción muy elaborada, carecen del problema que tiene el RSA para la obtención de claves.

- RSA: 900 Kbits por segundo.

- DES: 2 Gbit por segundo. AES: 3,1 Gbit por segundo.

Philip Zimmerman elaboró en 1991 el software PGP (Pretty Good Privacy). GRATIS GPG


k privada
de A
k secreta
k secreta
k pública
de A
M
D_A
k
k
C
E_A
M
Usuario A
k
Usuario B
k
Cifrado simétrico del mensaje
Firma
digital
sobre
h(M)
Cifrado asimétrico
Autenticación
del usuario e
integridad de M
Intercambio previo de clave de sesión k:
$E_B(k)$ en emisión y $D_B(k)$ en recepción

Protocolos Criptográficos

1. *Protocolos de Autentificación de Usuario*: Garantizar que el remitente es realmente quién pretende ser.

2. *Protocolos de Autentificación del Mensaje*: Garantizan la integridad del mensaje.

3. *Protocolos para Compartir Secretos:* Su objetivo es distribuir un cierto *secreto* entre un conjunto $\mathcal{P}$ de participantes, de forma que ciertos subconjuntos prefijados de $\mathcal{P}$ puedan, uniendo sus participaciones, recuperar dicho secreto.

4. *Pruebas de Conocimiento Cero:* Permiten a un individuo convencer a otro de que posee una cierta información, sin revelarle nada sobre el contenido de la misma.

5. *Transacciones Electrónicas Seguras:* Permiten realizar electrónicamente las operaciones bancarias habituales, firma electrónica de contratos, etc.

6. *Compromiso de bit*: Permiten a una parte A comprometerse con una elección (un bit o más generalmente una serie de bits) sin revelar tal elección hasta un momento posterior. El protocolo garantiza a otra parte B que A no cambia su elección.

7. *Elecciones Electrónicas:* Permiten realizar un proceso electoral electrónicamente, garantizando la deseable privacidad de cada votante y la imposibilidad de fraude.

8. *Jugar al Poker por Internet*: Posibilita a dos personas, físicamente separadas, mantener una partida de poker (o similar: cara o cruz, chinos, etc), comunicándose por correo electrónico, teléfono, etc, garantizando la imposibilidad de hacer trampa.

Esquemas para Compartir Secretos

PROBLEMA: Dado un secreto, repartir unos fragmentos de información entre varias personas, de modo que ciertas agrupaciones de estas personas puedan recuperar el secreto, pero las restantes agrupaciones no sean capaces de obtenerlo.

EJEMPLO: Sea un banco con una cámara acorazada que debe abrirse cada mañana con una cierta clave (el secreto). El banco tiene 5 cajeros encargados de tal apertura, pero desea que al menos sean necesarios 3 de ellos para abrirla.

La formalización matemática

$\mathcal{P} = \{P_1, \ldots P_l\}$, el conjunto de l *participantes* que quieren compartir un secreto,

$D \notin \mathcal{P}$ el *gestor* del esquema,

$\mathcal{K}$ un conjunto de *secretos* a repartir,

$\Gamma \subseteq 2^{\mathcal{P}}$, subconjunto de partes de P: agrupaciones autorizadas

$\mathcal{S}$ el conjunto de *Participaciones* que se reparten, en particular, $\mathcal{S}_i \subset \mathcal{S}$ es el conjunto de participaciones que puede recibir P_i.

Esquemas Umbral

Un caso particular de estructura de acceso es el de los denominados (l, t)-*Esquemas Umbral*: los conjuntos autorizados son los que tienen al menos t participantes, es decir, cualquier subconjunto de t o más participantes puede, uniendo sus participaciones, recuperar el secreto, mientras que cualquier subconjunto con menos de t no puede hacerlo.

Así, el ejemplo anterior de la cámara bancaria sería un $(5, 3)$-esquema umbral.

Esquema de Shamir

Dados l y $t \leq l$, enteros no negativos y un secreto $K \in \{0, ..., s-1\}$.

- Se toma un primo $p \geq \max\{s, l+1\}$.
- Se escogen aleatoria e independientemente $a_1, \ldots, a_{t-1} \in \mathbf{Z}_p$
- Se construye el polinomio de grado $t-1$, $q(x) = K + \sum_{i=1}^{t-1} a_i x^i$
- Se distribuye el secreto en las participaciones

  $$s_i = q(i) \in \mathbf{Z}_p,\ i = 1, \ldots, l$$

  que se reparten entre los miembros del colectivo.