# SALVAVIDAS-CRIPTOGRAFIA.pdf

**Informatik**

**Criptografía**

**4º Grado en Ingeniería Informática - Ingeniería del Software**

**Escuela Técnica Superior de Ingeniería Informática**
**Universidad de Sevilla**

# SALVAVIDAS CRIPTOGRAFÍA

## INFORMATIK

### Tema 1 CriptoClassic

Cifrado por desplazamiento → La clave es un número entero. Se cifra sumando a cada letra del texto en claro el valor de la clave

Cifrado afín → Para cada x ----> y = a*x +b, donde a es primo con n (el tamaño del alfabeto). El descifrado se hace con la función inversa: Para cada y ----> x = $a^{-1}$* (y-b).

Cifrado por sustitución → Cada clave es una permutación del alfabeto.

Método de análisis de frecuencias → Sabiendo en qué idioma está escrito un texto cifrado, analizar con qué frecuencia aparece cada símbolo para intentar llegar a una conclusión de que letra podría ser realmente.

Cifrado por transposición → En los cifrados por transposición se reordena el texto en claro (nótese que en las sustituciones se reordena el alfabeto).

Cifrado de Vigenere → La clave es una palabra o una frase. Se divide el texto en claro en bloques de la misma longitud que la clave (excepto el último, que puede ser menor). Cada bloque se cifra aplicando a cada carácter un desplazamiento de valor el correspondiente carácter de la clave. Para descifrar, basta con invertir la operación.

También está el método de autoclave, en el que para cifrar el primer bloque se usa la clave, pero para los siguientes se usa el resultado del cifrado del bloque anterior. ES débil ante un ataque de texto cifrado conocido.

CRIPTOANÁLISIS

Test de Kasiski → Es necesario que la razón entre la longitud del texto cifrado y la longitud de la clave sea suficientemente grande.

• Segmentos repetidos de texto claro cifrados con la misma parte de la clave dan lugar a segmentos iguales de texto cifrado.

• Gran número de estas coincidencias ocurrirán entre segmentos iguales separados una distancia múltiplo de la longitud de la clave.

• Kasiski propone buscar coincidencias de grupos de 3 o 4 caracteres para minimizar las colisiones debidas al azar.

**El máximo común divisor de los números resultantes es probablemente la longitud de la clave.**

Índice de coincidencia → Cada idioma tiene un IC característico, aunque dependiente de la naturaleza del texto (científico, literario, etc.) Índice de coincidencia de dos textos: Probabilidad de que al elegir al azar un carácter en cada texto, ambos sean iguales. Si a dos textos se les aplica la misma sustitución, el IC no varía.

Cifrado XOR → Como Vigenere, pero cambiando las sumas y restas por la operación lógica "o exclusivo" (XOR) de bits. El descifrado es igual que el cifrado, ya que XOR es involutiva.

## Criptosistema de Hill

### Cifrado

$$\overline{y} = M \cdot \overline{x}$$

Por ejemplo cifrando con una matriz $3 \times 3$
Texto en claro: $e, n, u \mapsto (2, 13, 21)$

$$\begin{pmatrix} 2 & 3 & 4 \\ 5 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 13 \\ 21 \end{pmatrix} = \begin{pmatrix} 23 \\ 22 \\ 24 \end{pmatrix} \mapsto \begin{pmatrix} w \\ v \\ x \end{pmatrix}$$

### Descifrado

$$\begin{pmatrix} 2 & 3 & 4 \\ 5 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 23 \\ 22 \\ 24 \end{pmatrix} = \begin{pmatrix} 2 \\ 13 \\ 21 \end{pmatrix} \mapsto \begin{pmatrix} e \\ n \\ u \end{pmatrix}$$

Muy débil ante ataques de texto en claro conocido

### Teoría de la información

La entropía depende de la distribución de probabilidad de los caracteres e indica lo plana que es: plana equivale a alta entropía y ocurre cuando todos los valores tienen probabilidades similares, mientras que es poco plana cuando algunos valores son mucho más probables que otros.

Con alta entropía es difícil poder predecir cuál es el próximo valor que va a presentarse, ya que todos los valores son igualmente probables. Una buena contraseña debe tener entropía alta.

### Cifrado de Vernam

Vernam propuso uno consistente en XOR simple (o Vigenere) en el que la clave:

1. Se genera aleatoriamente.

2. Es tan larga como el texto en claro.

3. Se usa una sola vez (para cada texto en claro se genera una clave nueva).

Las tres condiciones son completamente necesarias. Tiene secreto perfecto.

Problemas → **Generación de claves verdaderamente aleatorias** y distribución de claves.

# Tema 2 Números Pseudoaleatorios y Flujo

Problema de la generación de claves grandes suficientemente aleatorias (por ejemplo, para el cifrado de Vernam)

Las condiciones para que una secuencia binaria se considere criptográficamente segura son:

1. Periodo suficientemente grande.

2. Distribución uniforme: los unos y ceros en cualquier muestra deben distribuirse aproximadamente al 50%.

3. Imprevisible: con una muestra de la secuencia debe ser imposible poder predecir el bit siguiente.
4. Fácil, rápido y barato: eficiencia, coste computacional, consumo de recursos, apto para hardware, etc.

### Generadores por desplazamiento de registros retroalimentados (FSR)

Se utilizan n celdas. En cada iteración se calcula una función f sobre el contenido de las n celdas y el resultado se asigna a la ´ultima, desplazando a las anteriores. Hay dos tipos NLFSR y LFSR. Se suele usar el segundo.

### Generadores de congruencias lineales (LCG)

$x_{i+1} = (a \cdot x_i + b) \bmod m$

El periodo de esta sucesión es m (máximo) si yo sólo si se verifican:

1. b es primo con m.

2. Si p es primo y divide a m, p divide a a − 1.

3. Si m es múltiplo de 4, a − 1 es múltiplo de 4.

### Generador de Fibonacci

No son criptográficamente seguros.

### El generador Blum Blum Shub

Se eligen dos primos grandes p, q ambos congruentes con 3 modulo 4, con mcd(ϕ(p−1), ϕ(q− 1)) pequeño (para que el periodo sea largo) y hacemos m = p · q

Se toma como semilla un entero grande s primo con m y se calcula una sucesión:

$x0 = s^2 \bmod n$

$xi+1 = x_i^2 \bmod n; \forall i \geq 0$

## Flujo

### Modo síncrono

La secuencia pseudoaleatoria (flujo) se calcula independientemente del texto en claro y del texto cifrado; sólo depende de la clave.

### Modo asíncrono

La secuencia pseudoaleatoria generada es función de la semilla y de una cantidad fija de los bits anteriores del cifrado.

# Tema 3 Simetría

Simétrico ⇔ clave secreta compartida. El emisor y el receptor deben conocer la clave.

Una gran parte de los algoritmos de cifrado simétrico operan dividiendo el mensaje que se pretende codificar en bloques de tamaño fijo, y aplican sobre cada uno de ellos una combinación de operaciones de confusión (sustituciones) y de difusión (transposiciones). Estos algoritmos se denominan, en general, cifrados por bloques.

Cuando la longitud de la cadena que queremos cifrar no es un múltiplo exacto del tamaño de bloque tenemos que añadir información al final para que lo sea. El mecanismo más sencillo consiste en rellenar con ceros el último bloque que se codifica.

Los dos modos de operación para algoritmos de cifrado por bloques son:

**ECB (Electronic Codebook)** → Se subdivide la cadena en bloques y se cifran todos con la misma clave. Bueno para codificar cuando no importa el orden (bases de datos o ficheros) y es resistente a errores.

**CBC (Cipher Book Chaining Mode)** → La codificación de bloques anteriores condiciona la codificación del actual. Esto se consigue efectuando una operación XOR entre el bloque del mensaje que queremos codificar y el último criptograma obtenido.

**Redes de Feistel** → Método de cifrado por bloques en el que la operación de descifrado es idéntica a la de cifrado, solo hay que invertir el orden. El más conocido es DES (Data Encryption Standard).

DES es bueno y se usa actualmente, pero existen familias de claves llamadas débiles y semidébiles que deben ser evitadas. Se encuentra como opción en PGP.

AES no posee una estructura de red de Feistel, está compuesto de cuatro funciones invertibles diferentes.

# Tema 4 CriptoPublic

Diffie-Hellman

A y B se envían mensajes usando un canal abierto y consiguen compartir un secreto K que sólo es conocido por ellos:

1. Eligen y publican un primo adecuado p y un generador g, $2 \leq g \leq p-2$ del grupo multiplicativo de $Z_p$.
2. A elige un secreto x, $1 \leq x \leq p-2$ y envía a B el mensaje $g^x \bmod p$.
3. B elige un secreto y, $1 \leq y \leq p-2$ y envía a A el mensaje $g^y \bmod p$.
4. B recibe $g^x$ y calcula la clave compartida como: $K = (g^x)^y \bmod p$.
5. A recibe $g^y$ y calcula la clave compartida como: $K = (g^y)^x \bmod p$.

Ejemplo:

1. A y B eligen y publican un primo adecuado p = 71 y un generador g = 21 del grupo multiplicativo de$Z_{71}$.
2. A elige un secreto x = 46 y envía a B el mensaje $g^x \bmod p = 21^{46} \bmod 71 = 9$.
3. B elige un secreto y = 57 y envía a A el mensaje $g^y \bmod p = 21^{57} \bmod 71 = 61$
4. B recibe $g^x = 9$ y calcula la clave compartida como: $K = (g^x)^y \bmod p = 9^{57} \bmod 71 = 16$.
5. A recibe $g^y$ y calcula la clave compartida como $K = (g^y)^x \bmod p = 61^{46} \bmod 71 = 16$

Un elemento $g \in Z * p$ es generador, si cumple que para todo $1 \leq i \leq n$ que $g^{(p-1)/(p_i)} \neq 1 \bmod p$ donde $p - 1 = p^{\alpha 1} \cdots p_n{}^{\alpha n}$.

Ej: Para (g = 2, p = 181) tenemos $p - 1 = 2^2 \cdot 3^2 \cdot 5$

$2^{180/2} = 180 \bmod 181$, $2^{180/3} = 48 \bmod 181$, $2^{180/5} = 29 \bmod 181$.

Si dicen en $Z_{13}$ quiere decir que p=13.

Si un atacante interceptara toda la comunicación, llegaría a conocer los valores de p, g, $g^x$ y $g^y$.

Para encontrar K necesita conocer uno de los valores x, y.

Ataques de intercepción y suplantación

- A elige un secreto $x$, $1 \leq x \leq p-2$ y envía a B el mensaje $g^x \ mod\ p$.
  C intercepta $g^x \ mod\ p$, elige un $z$ con $1 \leq z \leq p-2$ y envía a B el valor $g^z \ mod\ p$.
- B elige un secreto $y$, $1 \leq y \leq p-2$ y envía a A el mensaje $g^y \ mod\ p$.
  C intercepta $g^y \ mod\ p$, y envía a B el valor $g^z \ mod\ p$.
- B recibe $g^z$ y calcula la clave compartida como
  $$K = (g^z)^y \ mod\ p$$
- A recibe $g^z$ y calcula la clave compartida como
  $$K = (g^z)^x \ mod\ p$$
  C calcula también la clave compartida haciendo
  $$K = (g^x)^z \ mod\ p$$

El protocolo es sensible a ataques activos del tipo Men-in-the-middle. Si la comunicación es interceptada por un tercero, este se puede hacer pasar por el emisor cara al destinatario y viceversa.

## Clave Pública

Asimetría ⇔ Separación de claves: una pública que cifra y otra privada que descifra.

Cifrado y descifrado asimétrico: la clave publica la tienen todos, la clave privada sólo la tiene el destinatario.


## Mochila de Merkle-Hellman

Dada una lista de enteros positivos ($a_1, a_2, \ldots, a_n$) y un entero positivo s, encontrar, si existe, una lista de enteros ($x_1, x_2, \ldots, x_n$) con xi ∈ {0, 1}, 1 ≤ i ≤ n tales que Pn i=1 xi · ai = s

Si la mochila es supercreciente (va de menor a mayor), es fácil decidir si el problema tiene solución y, en su caso, hallarla: Para i = n hasta 1

xi = 0

Si s ≥ bi

xi = 1, s = s – bi

Ej: s = 500, con A = {40, 50, 100, 300, 400} fíjate que A es supercreciente.

500 >400 lo cojo, s = 100 → 100 < 300 no lo cojo → 100 =100 lo cojo, s = 0 paro. Tiene solución.

Para la lista A, X quedaría 00101, siendo 1 los números que he cogido de la lista y 0 los que no.

## Cifrado y Descifrado

Cifrado → Representa el mensaje m como una cadena binaria $m_1\, m_2 \ldots m_n$. Calcula el entero c = $m_1a_1 + m_2a_2 + \ldots + m_na_n$.

Descifrado → $d = W^{-1} \cdot c \bmod M$. Resuelve el problema de la mochila supercreciente y encuentra enteros $r_1, r_2, \ldots, r_n \in \{0, 1\}$ tales que $d = r_1 b_1 + r_2 b_2 + \ldots r_n b_n$.

## RSA

Dado x ∈ Zn el inverso de x módulo n existe si y sólo si son coprimos (mcd es 1).

Ecuación diofántica. Hallar $e^{-1}$.

Para el cálculo de dicho inverso (en caso de que exista) se utiliza el algoritmo extendido de Euclides.

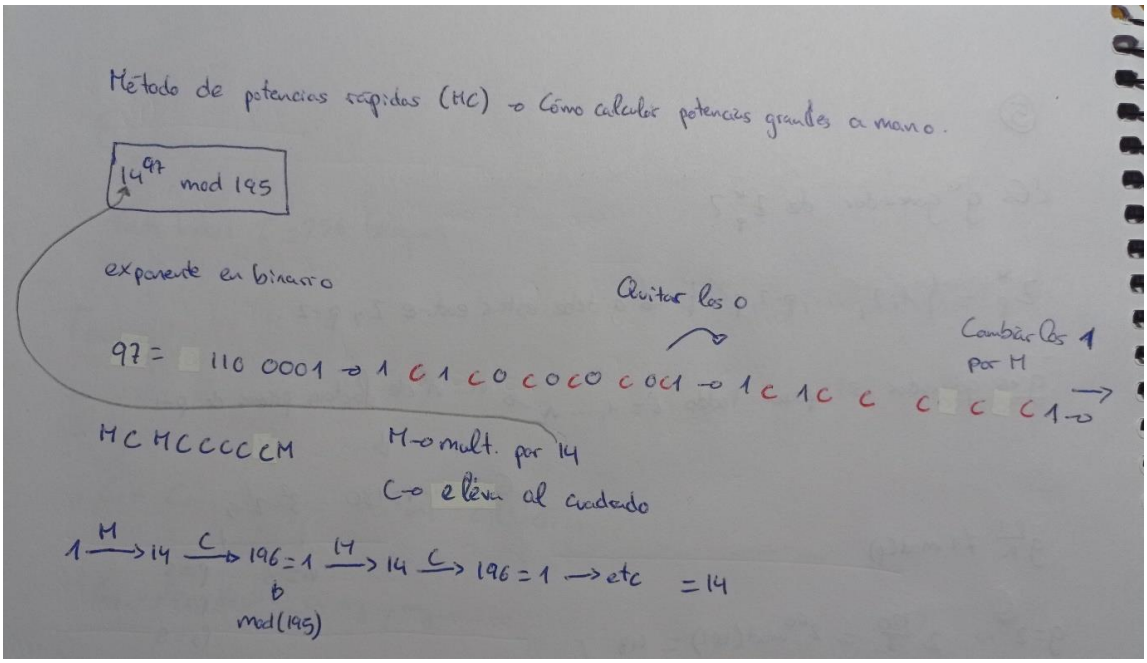
Cálculemos manualmente $17^{-1}$ *mod* 65:

| | | |
|---|---|---|
| (1) | $65 = 3 \cdot 17 + 14$ | $1 = 3 - 1 \cdot 2 =$ |
| (2) | $17 = 1 \cdot 14 + 3$ | $= 3 - 1 \cdot (14 - 4 \cdot 3) = -14 + 5 \cdot 3 =$ |
| (3) | $14 = 4 \cdot 3 + 2$ | $= -14 + 5 \cdot (17 - 1 \cdot 14) = 5 \cdot 17 - 6 \cdot 14 =$ |
| (4) | $3 = 1 \cdot 2 + 1$ | $= 5 \cdot 17 - 6\,(65 - 3 \cdot 17) = 23 \cdot 17 - 6 \cdot 65$ |

Así, la identidad de Bezout $1 = 23 \cdot 17 + (-6) \cdot 65$ leida módulo 65 nos da que $1 = 23 \cdot 17 \; mod \; 65$

En este caso $e^{-1}$ =23.

## Método de potencias rápidas




El problema matemático de difícil resolución en que radica la fortaleza del criptosistema RSA es el siguiente, dados:

• Un entero n, que se sabe que es producto de dos primos p y q (desconocidos)

• Un entero e, que sabemos que es primo con p − 1 y con q − 1.

• Un entero c, que sabemos que es el resultado de elevar un número desconocido 'm' a 'e' módulo n.

Hay que encontrar un entero m tal que $m^e = c \pmod n$, es decir, descifrar el mensaje cifrado c, que corresponde al texto en claro m.

• Factorizamos n = p·q (probando con los primos < √n)

• Calculamos $\phi = (p - 1) \cdot (q - 1)$

• Hallamos $d = e^{-1}$ en $Z_\phi$

• La solución es m = c d mod n

De hecho conociendo ϕ sería fácil encontrar p y q con sólo resolver una ecuación de segundo grado, ya que $\phi(n) = (p - 1)\cdot(q - 1)$ y, por ejemplo q = n/p, con lo que quedaría $\phi(n) = (p - 1) \cdot (n/p - 1)$, y resolveríamos la ecuación: $p^2 + (\phi(n) - n - 1) \cdot p + n = 0$

## Generación de claves

Cada usuario cera una clave pública y la correspondiente clave privada:

1. Genera dos primos grandes p y q, de tamaño similar.

2. Calcula $n = p \cdot q$ y $\phi = (p - 1) \cdot (q - 1)$.

3. Elige un entero e, $1 < e < \phi$ con mcd $(e, \phi) = 1$.

4. Calcula el único entero d, $1 < d < \phi$ tal que: $e \cdot d = 1 \bmod \phi$.

5. La clave publica es (n, e), la privada es d.

El entero 'e' se llama el exponente de cifrado, 'd' el exponente de descifrado y 'n' el módulo.

## Cifrado y descifrado

**Cifrado**: B cifra un mensaje para A.

(a) Obtiene la clave pública de A (n, e).

(b) Representa el mensaje como un entero m en el intervalo [0, n − 1].

(c) Calcula $c = m^e \bmod n$.

(d) Envía el texto cifrado c, a A.

**Descifrado**: A descifra el mensaje de B

Usa su clave privada para recuperar $m = c^d \bmod n$.

## Debilidades

Algunos valores del mensaje pueden dar cifrados inseguros:

• Los valores m=0 y m=1 siempre se cifran en sí mismos.

• Con exponentes y valores pequeños de m, el cifrado podría ser estrictamente menor que el módulo y el texto en claro podría obtenerse haciendo la raíz e-´ésima, sin tener en cuenta el módulo.

• Siempre hay mensajes que se cifran en sí mismos (se quedan sin cifrar), al menos 9. De hecho, su número exacto es:

$(1 + \text{mcd}(e - 1, p - 1)) \cdot (1 + \text{mcd}(e - 1, q - 1))$

• RSA es determinista, por lo que es viable un ataque de texto elegido: el atacante construye un diccionario de textos probables y sus cifrados. Interceptando un texto cifrado, el atacante puede usar este diccionario para descifrar el mensaje

## Ataques

**Fuerza bruta** → Si p y q son primos cercanos, p está cerca de √(n); así sería viable un ataque de fuerza bruta dividiendo n entre impares menores que p (n) y comprobando que, de un número exacto, que será el otro primo.

**Factorización de Fermat** → Si llamamos x = (p+q)/2, y = (p−q)/2, entonces $x^2 - n = y^2$.

Como $x^2 > n$, la idea es probar todos los x > √(n) hasta dar con uno que $x^2 - n$ sea cuadrado perfecto. Cuanto más cercanos sean p y q, más pequeño será y, y harán falta menos iteraciones. Con x e y, calculamos p = x + y, q = x − y.

## Múltiples claves que descifran

Además de la clave privada $d$ pueden encontrarse otras claves que también sirvan para descifrar, aunque no sean $d = e^{-1} \ mod \ n$:

Llamando

$$\gamma = mcm(p-1, q-1)$$

$$d_\gamma = e^{-1} \ mod \ \gamma$$

$$\lambda = \lfloor \frac{n - d_\gamma}{\gamma} \rfloor$$

Habrá $\lambda$ claves $d_k$ que descifran (además de $d$):

$$d_k = d_\gamma + k \cdot \gamma, \qquad k = 0, 1, \ldots, \lambda$$

Ejemplo:
Clave pública $(n, e) = (3053 = 71\dot{4}3, 157)$.
Clave privada $d = 157^{-1} \ mod \ (70 \cdot 42) = 1573$.

$$\gamma = mcm(70, 42) = 210.$$

$$d_\gamma = 157^{-1} \ mod \ 210 = 103.$$

Claves que descifran:

$$103 + k \cdot 210, \qquad k = 0, \ldots \frac{3053 - 103}{210} = 14$$

Esto es: $103, 313, 523, 733, 943, 1153, 1363, \mathbf{1573}, 1783, 1993, 2203, 2413, 2623, 2833, 3042.$

## Elección de claves buenas

• p y q deben ser suficientemente grandes, del mismo tamaño, pero no muy cercanos.

• Debe maximizarse mcm(p − 1, q − 1)

• Deben minimizarse mcd(e − 1, p − 1) y mcd(e − 1, q − 1).

• p y q deben ser primos fuertes.

Error: elegir un mismo módulo n y distribuir distintos pares (e, d) a los usuarios de la organización.

El problema es que el conocimiento de un simple par (e, d) puede revelar la factorización de n y romper todo el sistema.

# Tema 5 Firma y Resumen Digital

La firma digital cumpla una serie de requisitos:

1. Única: solo puede ser generada por el firmante.

2. No falsificable: para falsificarla hay que resolver problemas computacionalmente intratables.

3. Dependiente del documento.

4. Verificable por terceros.

5. Innegable: el firmante no debe poder negar su firma.

6. Viable: deben existir algoritmos eficientes de generación y verificación

La firma RSA cumple todos los requisitos excepto el sexto.

## Función resumen

El resumen de un documento D debe servir como una representación compacta de D y debe poder usarse como si el documento estuviera identificado de forma única por su resumen.

Condiciones básicas que debe cumplir una función resumen:

1) Facilidad y velocidad de cálculo. h(D), el resumen de un documento D, debe ser muy fácil y rápido de calcular.

2) Resumen de longitud fija. El tamaño del resumen h(D) debe ser fijo (definido por la función), sin depender del tamaño de D.

3) Difusión. h(D) debe ser una función en la que intervengan todos los bits de D. Si en D cambiase un bit, en h(D) deberían cambiar aproximadamente la mitad de los bits.

4) Unidireccionalidad. Conocido un resumen h(D), debe ser computacionalmente imposible encontrar D.

5) Resistencia a colisiones. Es evidente que h nunca puede ser una función inyectiva, con lo que siempre existen colisiones: pares (D, D') tales que h(D) = h(D'). Distinguiremos entre resistencia débil y fuerte a las colisiones.

> 5a) Resistencia débil. Conocido D, debe ser computacionalmente imposible encontrar D0 tal que h(D) = h(D0 ).
>
> 5b) Resistencia fuerte. Debe ser computacionalmente imposible encontrar D y D0 tales que h(D) = h(D0 ).

Ejemplos de funciones resumen → CRC32, MD5, SHA-1, SHA-2.

## Proceso de Firma

Holmes tiene (n, e, d) como clave RSA y quiere firmar un documento m (cifrado o no), para ello:

1. Elige una función resumen h y calcula el resumen r = h(m).
2. Calcula y publica cifrado del resumen con la clave privada: $f = RSA_{(n,d)}(r) = r^d \bmod n$
3. Se adjunta la firma al documento.
4. Sólo puede firmar el propietario de la clave privada.

## Proceso de verificación de firma

1. Se calcula el resumen r = h(m)
2. La firma es válida si $RSA_{(n,e)}(f) = f^e \bmod n = r$
3. Si los resultados no coinciden: el documento ha sido alterado o el firmante no es el legal.
4. Cualquiera puede verificar la firma ya que para ello se usa la clave pública.

Ej de firma y verificación:

Firmamos el mensaje $m =$ "esto es muy importante"

- Como función resumen usamos $h = CRC32$:

$$h(m) = F0EBDC78 = 4041989240$$

- Parámetros RSA:

$$(n = 10009202107, e = 123456789)$$

$$d = 4295292925$$

- Firma:

$$4041989240^{4295292925} \ mod \ 10009202107 = 14275210$$

- Verificación de la firma:

$$14275210^{123456789} \ mod \ 10009202107 = 4041989240 = h(m)$$

**Ataques a firmas digitales**

Un ataque factible cuando la función resumen no es muy robusta es el llamado "ataque de cumpleaños": Moriarty quiere engañar a Watson para que firme un contrato fraudulento, y para ello:

1. Prepara un contrato bueno C y otro malo C'.

2. Modifica C y C' sin cambiar los significados (usando sinónimos, comas, espacios, líneas en blanco, etc) hasta que encuentra una pareja (C, C') con h(C) = h(C').

3. Envía C a Watson para que lo firme. Watson está de acuerdo con C, lo firma (usando h) y devuelve (C, $f_W$ (C)).

4. Moriarty sustituye (C, $f_W$ (C)) por (C', $f_W$ (C)).

La protección contra este tipo de ataques proviene de una buena elección de h. La probabilidad de que en el paso Moriarty encuentre una tal pareja (C, C') con h(C) = h(C') es muy pequeña si la función resumen h genera resúmenes suficientemente grandes.

## Cifrado de ElGamal

Se basa en Diffie-Hellman. Puede ser utilizado tanto para generar cifras digitales como para cifrar o descifrar.

### Generación de claves

Cada usuario crea su clave pública y la correspondiente clave privada:

1. Construye un primo grande p y un generador del grupo multiplicativo de $Z_p$.

2. Elige aleatoriamente un entero a, $1 < a \leq p - 2$ y calcula $\beta = g^a \bmod p$.

3. La clave publica es (p, g, β). La clave privada es a.

### Cifrado y Descifrado

B cifra un mensaje para A

1. Obtiene la clave publica de A (p, g, β).

2. Representa el mensaje como un conjunto de enteros m1m2 · · · en el intervalo [0, p − 1].

3. Para cada $m_i$ elige al azar un entero $k_i$ , $1 < k_i \leq p - 2$.

4. Calcula $\gamma_i = g^{k_i} \bmod p$ y $\delta_i = m_i \cdot \beta^{k_i} \bmod p$.

5. Envía el texto cifrado $c_i = (\gamma_i , \delta_i)$ a A.

Este proceso de descifrado nos devuelve el mensaje original, puesto que, por el Teorema de Fermat, se tiene que en Zp, $\gamma^{p-1} = 1$

Con lo que $\gamma^{p-1-a} = \gamma^{p-1} \cdot \gamma^{-a} = g^{-ak}$.

Por tanto: $\gamma^{-a} \cdot \delta = g^{-ak} \cdot m \cdot g^{ak} = m$

## Eficiencia y seguridad

**Desventajas**

Una desventaja es que el mensaje cifrado es el doble de largo que el original.

El proceso de descifrado hace sólo una potencia, pero el de cifrado requiere dos, de exponente k, que pueden hacerse más rápidas eligiendo adecuadamente el exponente.

**Ventajas**

Una gran ventaja es que, al elegirse k de forma aleatoria, el cifrado de un mismo texto dará resultados diferentes, aunque esto es transparente para el descifrado.

Es muy flexible y puede adaptarse al uso de otro grupo diferente de $Z^*_p$.

Resulta critico que se usen diferentes enteros aleatorios k para cada mensaje. Si se usara el mismo k para m1 y m2, dando lugar a los cifrados (γ1, δ1) y (γ2, δ2), entonces δ1/δ2 = m1/m2 y el conocimiento de uno de los mensajes revelaría el otro.

## Firma digital ElGamal

Los parámetros utilizados por el esquema ElGamal son la función de resumen h resistente a colisiones, un número primo p muy grande tal que el cómputo de logaritmos discretos módulo p sea difícil y un generador pseudoaleatorio g para el grupo multiplicativo $Z^*_p$.

### A firma un mensaje m:

- Sus claves ElGamal son: publica = (p, g, β), privada = a.
- Elige un entero secreto k, $1 < k \leq p - 2$, primo con $p - 1$.
- Calcula $r = g^k \bmod p$.
- Calcula $s = k^{-1} (h(m) - ar) \bmod (p - 1)$.
- La firma del mensaje m es el par (r, s).

### B verifica la firma:

- Calcula $v_1 = \beta^r r^s \bmod p$.

• Calcula $v_2 = g^{h(m)} \bmod p$.

• La firma es válida si $v_1 = v_2$.

Ej de firma y verificación

En la r sería $2^{1529}$, hay un error.

Firma:

- A tiene como clave pública $(p = 2357, g = 2, \beta = 1185)$, clave privada $a = 1751$ y firma un mensaje $m$ de resumen $h(m) = 1463$.
- A selecciona $k = 1529$ y calcula
  $r = 21529 \ mod \ 2357 = 1490$
  $s = 1529^{-1}(1463 - 1751 \cdot 1490) \ mod \ 2356 = 1777.$
- La firma es $(1490, 1777)$.

Verificación:
B verifica la firma calculando
$\nu_1 = 1185^{1490} \cdot 1490^{1777} \ mod \ 2357 = 1072$
$\nu_2 = 2^{1463} \ mod \ 2347 = 1072.$

# Tema 6 Aplicaciones

Los certificados digitales sirven para:

• Autenticar la identidad del propietario ante terceros.

• Firmar digitalmente de forma que se garantice la integridad de los datos trasmitidos y su procedencia.

• Cifrar datos para que sólo el destinatario de la información pueda acceder a ella.

El estándar para certificados digitales es el X.509. Una forma de poder confiar en el certificado digital de un interlocutor al que no conocemos es que el certificado esté avalado por una tercera parte en la que sí confiamos (autoridad de certificación). Esa tercera parte (TTP, Trusted Third Party) avalará con su firma digital que el certificado es de fiar.

La principal función de una autoridad de certificación es firmar la clave pública y los datos del usuario para que otros usuarios puedan validar su autenticidad.

Un certificado digital solo contiene datos del usuario y su clave pública, no la privada.

En un sistema de comunicación seguro (SSL, por ejemplo) se usa un cifrado asimétrico para intercambio de claves y firma digital y uno simétrico para el cifrado de datos

Un certificado digital contiene la clave pública del usuario, no la privada, firmada con la clave privada de una autoridad certificadora.

Una PKI es un conjunto de protocolos, servicios y estándares que soportan aplicaciones basadas en criptografía de clave pública. La base de las PKI es el modelo de confianza basado en Terceras Partes de Confianza (TTP).