# Introducción a la criptografía de clave pública

Fernando Martínez
fernando.martinez@upc.edu

Departament de Matemàtiques • Universitat Politècnica de Catalunya

10 de octubre de 2022

# Introducción a la clave pública

- Los sistemas de clave secreta tienen el inconveniente de que, antes de establecer la comunicación cifrada, los participantes deben enviarse previamente, por un canal seguro, la clave.

- Los sistemas de clave pública permiten establecer comunicación cifrada sin necesidad de intercambio previo de claves.

# Los inicios (públicos) de la criptografía de clave pública

Diffie, W., and Hellman, M. **(1976)**
*New directions in cryptography.* IEEE Trans. Inform. Theory IT-22, 644–654.

# Introducción a la clave pública

Los sistemas de clave secreta se basan en la dispersión y confusión de la información mientras que la filosofía de los sistemas de clave pública es totalmente diferente:

1. se elige un problema que sea difícil de resolver en el caso general,
2. se toma un caso particular del anterior que sea fácil de resolver,
3. se modifica el caso particular para que parezca el caso general.

# Un ejemplo: Criptosistema de la mochila (I)
## Problema de la mochila.

Consideremos la sucesión

$$\{1, 5, 8, 9, 10, 15, 16, 19, 20, 23\}$$

y pensemos cada letra del alfabeto como una cadena binaria de cinco bits:

A=00000,..., I=01001,..., N=01110, O=01111,..., S=10011,...

Para cifrar la palabra `SI` el proceso a seguir sería:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 8 | 9 | 10 | 15 | 16 | 19 | 20 | 23 | |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | |
| 1 | + 0 | + 0 | + 9 | + 10 | + 0 | + 16 | + 0 | + 0 | + 23 | = 59 |

# Un ejemplo: Criptosistema de la mochila (II)

- ¿Cuál sería el texto en claro correspondiente al cifrado 100?
- Por fuerza bruta tendríamos que probar

$$\binom{10}{1} + \binom{10}{2} + \cdots + \binom{10}{10} = \sum_{k=1}^{10} \binom{10}{k} = 2^{10} - 1$$

  combinaciones.
- En general, si tenemos $n$ términos en la lista el número de combinaciones posibles es $2^n - 1$.
- Además, si no elegimos cuidadosamente la sucesión, la solución podría no ser única.

# Un ejemplo: Criptosistema de la mochila (III)
# Problema fácil de la mochila.

Sucesión supercreciente $a_n > \sum_{i=1}^{n-1} a_i$

{1,3,5,11,22,45,88,180,357,712}

Si queremos descifrar el criptograma **1356** el proceso a seguir es:

| | | |
|---|---|---|
| **1356** | mayor que 712, | 712 forma parte solución |
| 1356 - 712 = **644** | entre 357 y 712, | 357 forma parte solución |
| 644 - 357 = **287** | entre 180 y 357, | 180 forma parte solución |
| 287 - 180 = **107** | entre 88 y 180, | 88 forma parte solución |
| 107 - 88 = **19** | entre 11 y 22, | 11 forma parte solución |
| 19 - 11 = **8** | entre 5 y 11, | 5 forma parte solución |
| 8 - 5 = **3** | entre 3 y 5, | 3 forma parte solución |
| 3 - 3 = **0** | | |

El texto es `NO`:

| 1 | 3 | 5 | 11 | 22 | 45 | 88 | 180 | 357 | 712 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

El coste es $n$ y la solución es única

# Un ejemplo: Criptosistema de la mochila (IV) Transformación para que no parezca sencillo.

Multiplicamos la sucesión por un entero $t$ módulo $M$ ($M > \sum_{i=1}^{n} a_i$ y $\text{mcd}(t, M) = 1$) y la reordenamos.

Si tomamos $t = 100$ y $M = 1511$, $\{1, 3, 5, 11, 22, 45, 88, 180, 357, 712\} \rightarrow$
$\{100, 300, 500, 1100, 689, 1478, 1245, 1379, 947, 183\}$ .
Reordenando $\{100, 183, 300, 500, 689, 947, 1100, 1245, 1379, 1478\}$.
Para descifrar el criptograma 6185 el proceso a seguir sería:

1. $100^{-1} \equiv 136 \mod 1511$.
2. $6185 \cdot 136 \equiv 1044 \mod 1511$.
3. Se resuelve el problema de la mochila para 1044 y la sucesión supercreciente, 1044=712+180+88+45+11+5+3.
4. Se aplica la permutación correspondiente para obtener el mensaje,

| 100 | 183 | 300 | 500 | 689 | 947 | 1100 | 1245 | 1379 | 1478 |
|---|---|---|---|---|---|---|---|---|---|
| | 712 | 3 | 5 | | | 11 | 88 | 180 | 45 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

que es `NO`.

# Un ejemplo: Criptosistema de la mochila (y V)

- En el criptosistema de la mochila la clave pública es la sucesión obtenida a partir de la sucesión supercreciente multiplicando por $t$, tomando módulo $M$ y aplicándole una permutación $\pi$. La clave privada es $t$, $M$, $\pi$ y la sucesión supercreciente.
- Este criptosistema es simple, rápido (comparable a los de clave secreta) y muy fácil de implementar.
- Tiene un inconveniente: no es seguro.

A. Shamir. *A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem.* IEEE Trans. Inform. Theory, **IT-30**:699-704, 1984.

# Criptosistemas de clave pública

- Los criptosistemas de clave pública, además de permitir el establecimiento de comunicación cifrada sin el intercambio previo de claves, hacen posible la firma de documentos y su posterior verificación.
- Estos sistemas constan de dos **claves**, una pública que se utiliza para cifrar ~~y~~/ o verificar firmas, y otra privada usada para descifrar ~~y~~/ o firmar. La clave privada es *computacionalmente* imposible de obtener a partir de la clave pública. La existencia de estas dos claves es la razón por la cual estos criptosistemas reciben el nombre de asimétricos.

# Un ejemplo: Criptosistema de la mochila (Firma)

Si deseamos firmar el mensaje NO=[0, 1, 1, 1, 0, 0, 1, 1, 1, 1] con la clave privada {1,3,5,11,22,45,88,180,357,712}, el proceso sería:

1. Calculamos su hash, por ejemplo:
   $0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 + 1 \cdot 2^7 + 1 \cdot 2^8 + 1 \cdot 2^9 = 974$
2. *Desciframos* el hash: 974. En este caso obtendríamos:
   [0, 1, 0, 1, 1, 0, 0, 1, 1, 0].
3. La firma del mensaje [0, 1, 1, 1, 0, 0, 1, 1, 1, 1] será
   [0, 1, 0, 1, 1, 0, 0, 1, 1, 0]

Para verificar que la firma es correcta *ciframos* [0, 1, 0, 1, 1, 0, 0, 1, 1, 0] con la clave pública {100,183,300,500,689,947,1100,1245,1379,1478} y el resultado ha de ser igual que el hash de [0, 1, 1, 1, 0, 0, 1, 1, 1, 1] módulo 1511 (para firmar se ha de hacer público $M$ también).

# Criptosistemas de clave pública

Estos criptosistemas se basan en la utilización de funciones unidireccionales (*one-way functions*) con puerta trasera (*trap-door*).

- Función unidireccional: Aplicación $f$ tal que existe un algoritmo polinómico para calcular $f(x)$ pero no existen algoritmos polinómicos para calcular $f^{-1}(y)$.
- Puerta trasera para una función unidireccional $f$ es una información que permite diseñar un algoritmo polinómico para calcular $f^{-1}(y)$.

# Un ejemplo real: RSA

Rivest, R.; A. Shamir; L. Adleman **(1978)**.[1]
*A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.* Communications of the ACM 21 (2): 120–126

Los mensajes y criptogramas son elementos de $\mathbb{Z}_n$, $n = pq$, $p$, $q$ primos.

- Clave pública: $\{e, n\}$, $e$ arbitrario tal que $(e, \varphi(n)) = 1$.
- Clave privada: $\{d, p, q\}$, $d$ inverso de $e$ módulo $\varphi(n)$.
- Cifrado: $c \equiv m^e \mod n$.
- Descifrado: $m \equiv c^d \mod n$.
- Firma: $f \equiv m^{\tilde{d}} \mod n$.
- Verificación de la firma: $v = f^{\tilde{e}} \equiv m \mod n$.

[1] The Original RSA Patent as filed with the U.S. Patent Office by Rivest; Ronald L. (Belmont, MA), Shamir; Adi (Cambridge, MA), Adleman; Leonard M. (Arlington, MA), **December 14, 1977**, U.S. Patent 4,405,829.

# RSA: Observaciones (I)

- El RSA funciona como consecuencia del teorema de Fermat y del teorema chino de los restos.
- Conocido $\varphi(n)$ se puede calcular $d$ utilizando la indentidad de Bezout.
- A partir de $p$ y $q$ hallar $\varphi(n)$ es fácil; sin ellos, es necesario factorizar $n$.
- Para romper el RSA tenemos que saber calcular raíces en $\mathbb{Z}_n$ o bien hallar la descomposición de $n$ en factores primos.
- La función unidireccional es $x^e \mod n$, su función inversa es el cálculo de raíces en $\mathbb{Z}_n$ y la puerta trasera son los factores primos de $n$.

# RSA: Observaciones (y II)

- En la práctica los criptosistemas asimétricos no se utilizan para cifrar mensajes sino que se usan para cifrar claves temporales utilizadas por criptosistemas simétricos con los que se cifra el mensaje ya que éstos son mucho más eficientes.

- Por la misma razón lo que se firma en realidad es un *hash* del mensaje.

# RSA: Algunas cuestiones

1. Generación de números aleatorios: Si solamente genera una pequeña cantidad o es predecible, un atacante podría reproducir la secuencia y hallar $p$ y $q$ fácilmente. *(O las claves de sesión.)* [2,3]
2. Generación de números primos: El coste de factorización de un número depende de la cantidad de factores primos en que descompone. Si $n = pq$ habrá que *probar* del orden de $\sqrt{n}$ candidatos para factorizar, pero si $n = pqr$ entonces sólo es necesario *probar* $\sqrt[3]{n}$.
3. ¿Cómo podemos estar seguro que una clave pública corresponde a quién dice corresponder?

[2] Debian Security Advisory DSA-1571-1 openssl – predictable random number generator, http://www.debian.org/security/2008/dsa-1571

[3] Ron was wrong, Whit is right, http://eprint.iacr.org/2012/064.pdf

# Clave pública

## Presente

1. Factorización de enteros
   - RSA
2. Logaritmo discreto
   - Cuerpos finitos
   - Curvas elípticas

## Futuro (Post-Quantum)

1. Retículos (Lattices)
2. Hash-based signatures
3. Códigos
4. Polinonios cuadráticos de varias variables[4]
5. Isogenias[5,6]

[4] W. Beullens, Breaking Rainbow Takes a Weekend on a Laptop, Cryptology ePrint Archive, Paper 2022/214

[5] W. Castryck, T. Decru, An efficient key recovery attack on SIDH (preliminary version), Cryptology ePrint Archive, Paper 2022/975

[6] L. Maino, C. Martindale, An attack on SIDH with arbitrary starting curve, Cryptology ePrint Archive, Paper 2022/1026

# Seguridad

Las longitudes de las claves (en bits) son orientativas, pueden variar dependiendo de los avances en *hard* y técnicas de criptoanálisis.

| Date | Minimum of Strength | Symmetric Algo-rithms | Factoring Modulus | Discrete Logarithm | | Elliptic Curve | Hash Signature | Hash HMAC, key derivation |
|---|---|---|---|---|---|---|---|---|
| | | | | Key | Group | | | |
| Legacy | 80 | 2TDEA | 1024 | 160 | 1024 | 160 | SHA-1 | |
| 2016<br>2030 | 112 | 3TDEA | 2048 | 224 | 2048 | 224 | SHA-224<br>SHA-512/224<br>SHA3-224 | |
| 2016<br>2030 | 128 | AES-128 | 3072 | 256 | 3072 | 256 | SHA-256<br>SHA-512/256<br>SHA3-256 | SHA-1 |
| 2016<br>2030 | 192 | AES-192 | 7680 | 384 | 7680 | 384 | SHA-384<br>SHA3-384 | SHA-224<br>SHA-512/224 |
| 2016<br>2030 | 256 | AES-256 | 15360 | 512 | 15360 | 512 | SHA-512<br>SHA3-512 | SHA-256<br>SHA-512/256<br>SHA-384<br>SHA-512<br>SHA3-512 |

NIST SP 800-57 Part 1 Revision 5 Recommendation for Key Management: Part 1 – General
Algorithms, Key Size and Protocols Report 2018, H2020-ICT-2014 – Project 645421, D5.4 ECRYPT-CSA