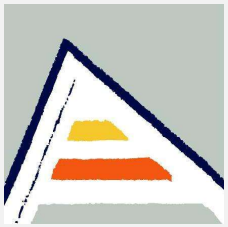


Grado en Ingeniería Informática

Sistemas distribuidos

Criptografía

Víctor Vives
vvives@dtic.ua.es

Departamento de Tecnología Informática y Computación

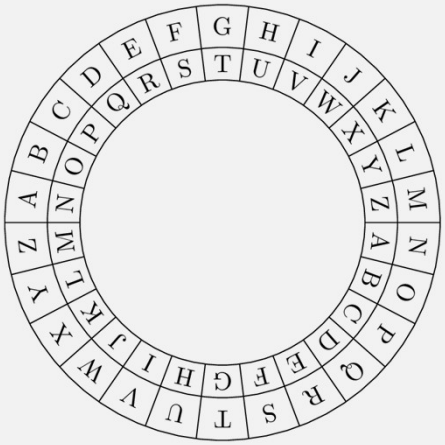
2021 - 2022

1. Introducción

Orígenes de la criptografía

La **criptografía** es una necesidad derivada de realizar **comunicaciones** creada para preservar la **privacidad de la información** que se transmite y garantizando que una persona no autorizada no pueda leer el contenido del mensaje.

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



1. Introducción

Se considera que un algoritmo de cifrado es **resistente** si cumple con la difusión y la confusión

Difusión

La difusión nos dice que si se cambia un bit en el texto sin cifrar, deberían cambiarse la mayor cantidad posible de bits en el texto cifrado. Para conseguir este efecto se realizan las **permutaciones**.

Confusión

En cambio la confusión quiere decir que la relación entre el texto cifrado y la clave sea lo más compleja posible. Para este caso las **sustituciones** cumplen con dicho objetivo.

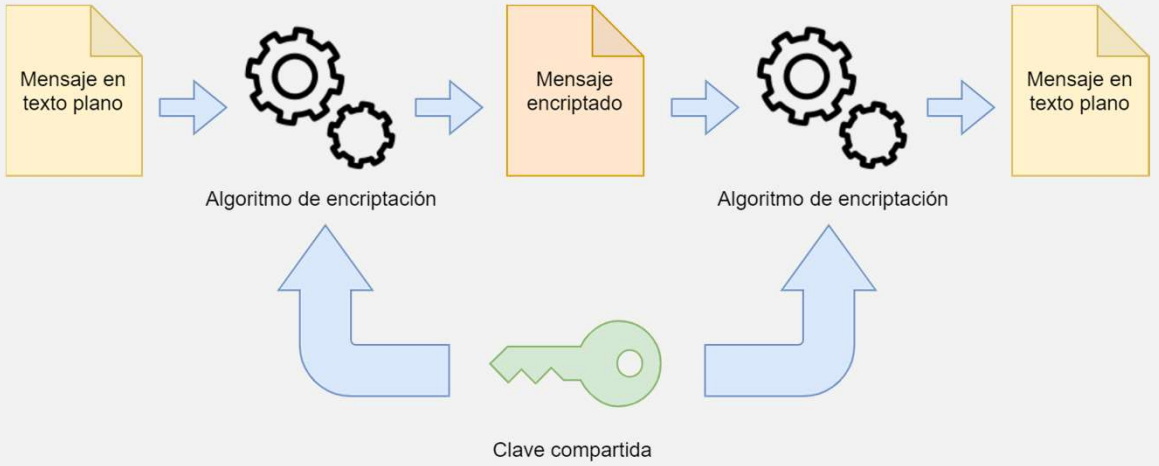
2. Criptografía simétrica

Criptografía simétrica

Solo se utiliza una **clave pública** para **cifrar y descifrar el mensaje**

La clave la tienen que conocer tanto emisor como receptor: **punto débil** del sistema

La **intercepción de la clave** es más sencillo que realizar un ataque de fuerza bruta



2. Criptografía simétrica

Criptografía simétrica

Los algoritmos de clave simétrica son **seguros y altamente eficientes** si se usan adecuadamente. Son útiles para cifrar grandes cantidades de datos sin tener un efecto negativo en el rendimiento. Un buen cifrado pone **toda la seguridad en la clave** y ninguna en el algoritmo.

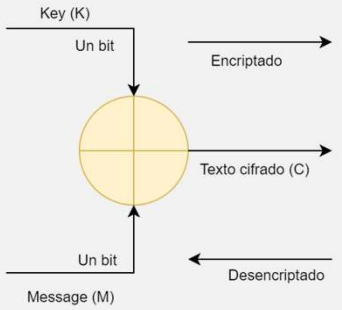
Cifrados de flujo

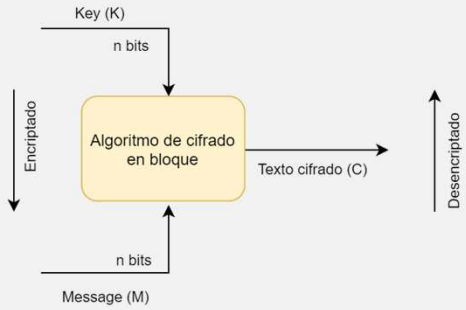
Cifran el mensaje con correspondencia bit a bit sobre el flujo o *stream*.

Cifrados de bloque

Cifran el mensaje dividiendo el flujo en bloques de k bits.

$$E(K, M) = \{M\}_K \qquad D(K, \{M\}_K) = M$$





2. Criptografía simétrica

Cifrados de flujo

- RC4 (Rivest Cipher 4): Este algoritmo cifra un byte a la vez. Una entrada clave es un generador de bits pseudoaleatorio que produce un número de flujo de 8 bits. La salida del generador se llama flujo de claves, se combina un byte a la vez con el cifrado de flujo de texto plano usando la operación XOR.

Cifrados de bloque

- TEA (Tiny Encryption Algorithm): Un simple pero efectivo algoritmo desarrollado en la Universidad de Cambridge (1994). Clave de 128 bits (16 bytes). Bloques de 64 bits.
- DES (Data Encryption Standard): No demasiado fuerte en su formato original (Estados Unidos, 1977). Clave de 56 bits (8 bytes). Bloques de 32 bits.
- Triple DES: Aplica DES tres veces con dos claves distintas (IBM, 1988). Clave de 168 bits (24 bytes). Bloques de 64 bits.
- IDEA (International Data Encryption Algorithm): Parecido al TEA (1990). Clave de 128 bits (16 bytes). Bloques de 64 bits
- AES (Advanced Encryption Standard). Hasta la fecha no se ha encontrado ninguna vulnerabilidad (1997). Clave de 128, 160, 192, 224 o 256 bits (16, 20, 24, 28 o 32 bytes). Bloques de 128 bits.

2. Criptografía simétrica

Ataque sobre AES

Existen actualmente ataques sobre distintas implementaciones que utilizan AES pero **ningún ataque criptográfico sobre el propio algoritmo**.

El único ataque que puede considerarse "efectivo" sobre AES es el ataque biclicuo o *biclique attack*.

No es una amenaza real.

El ataque coge bloques de 4 bits y esto reduce la seguridad a 124 bits.

Un ataque de fuerza bruta sobre el algoritmo de 124 bits lleva aproximadamente 2^{124} operaciones.

El ataque de fuerza bruta sobre el algoritmo de 128 bits lleva 2^{126} operaciones.

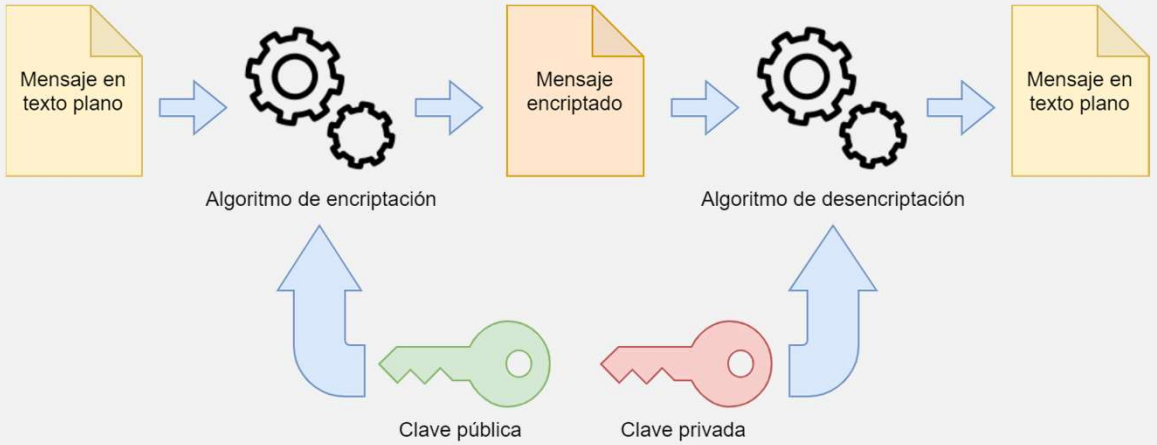
3. Criptografía asimétrica

Criptografía asimétrica

Se utilizan **dos claves**: la **pública** y la **privada**

La clave pública se puede difundir sin problema a quienes quieren mandarte un mensaje cifrado

La clave privada no debe ser revelada nunca



3. Criptografía asimétrica

Cifrados asimétricos

- RSA (Rivest, Shamir y Adleman, 1978): El primer algoritmo práctico y el que se utiliza con mayor frecuencia. Su seguridad radica en el problema de factorización de números enteros y se basa en el producto entre dos números primos grandes. Clave entre 512 y 2048 bits.
- ECC (Elliptic Curve Criptography, 1985) o curvas elípticas: Se basa en las matemáticas de las curvas elípticas. Es más rápida y utiliza claves más cortas (256 bits) que otros métodos como RSA mientras proporcionan un nivel de seguridad equivalente. Una clave de 256 bits ofrece la misma seguridad que una clave de 3072 bits.
- DSA (Digital Signature Algorithm, 1991): Estándar del Gobierno Federal de los EEUU para firmas digitales. Sirve para firmar pero no para autenticar. La desventaja frente al RSA es que requiere mucho más tiempo de cómputo.

Los **certificados SSL y TLS** se emiten con **RSA de 2048 bits** (617 dígitos decimales).

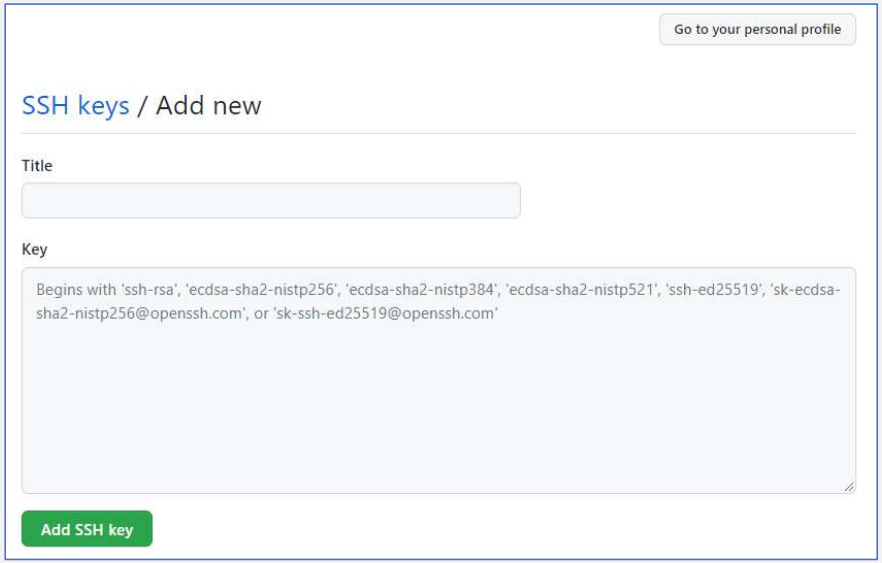
El **problema RSA** hace referencia a **la dificultad de efectuar una operación de clave privada conociendo tan solo la clave pública**.

¿Por qué no se utiliza ECC si es igual de seguro pero más rápido y con claves más cortas? (trabajo opcional)

3. Criptografía asimétrica

Ejemplo de clave RSA

Github emplea RSA para el protocolo SSH.




```
C:\Users\vvives-nouss
λ  ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\vvives-nouss/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\vvives-nouss/.ssh/id_rsa.
Your public key has been saved in C:\Users\vvives-nouss/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Vii+MKBpgUuYHp+WUNwjlZAhhcoL9EmtGbjtM6EfyIM vvives-nouss@DESKTOP-S9KIC61
The key's randomart image is:
+---[RSA 3072]----+
| .==B..          |
|o=o* =   .       |
|B=* * o . .      |
|=*=Oo. . .       |
|Bo==+ . S        |
|E*.= o o         |
|  o + .          |
|   .             |
|                 |
+----[SHA256]-----+
C:\Users\vvives-nouss
λ  cat C:\Users\vvives-nouss\.ssh\id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDbXTn2RR66kEEC5GphLEFKK6ghdENeIj5PkrDK4bqXb
hbAvL4qiU/MWDXg6lEqYKVpivG/lrh8puDdwOpX/wHr7WCdidwR9RlIa7nDDOY/HbiXE+HAHnWwGS9v4Q
C:\Users\vvives-nouss
λ  ls C:\Users\vvives-nouss\.ssh\


    Directorio: C:\Users\vvives-nouss\.ssh


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        08/11/2021     19:19           2675 id_rsa
-a----        08/11/2021     19:19            583 id_rsa.pub
```

3. Criptografía asimétrica

Cifrados asimétricos

Claves de encriptación (K_e) y desencriptación (K_d)

$D(K_d \cdot E(K_e, M)) = M$

Ataques al algoritmo RSA

1. Búsqueda del espacio de mensajes: cifrar todos los bloques hasta que se encuentre una coincidencia con el texto cifrado.
2. Ataque de texto cifrado o *Guessing d*: el atacante conoce el texto sin formato y cifrado e intenta descifrar el exponente *d*.
3. Ataque de ciclo: cifrar el texto rápidamente hasta encontrar coincidencias.
4. Módulo común: distribución de claves públicas y privadas a los usuarios de una organización.
5. Cifrado defectuoso: el atacante tiene acceso a la comunicación y la modifica al invertir un bit en el exponente público.
6. Exponente bajo: si el mensaje se cifra tres veces con diferentes claves se puede recuperar. Teorema del resto chino.
7. Factorizar la clave pública: la mejor forma actualmente de descifrar el algoritmo RSA.

4. Criptografía híbrida

Criptografía híbrida

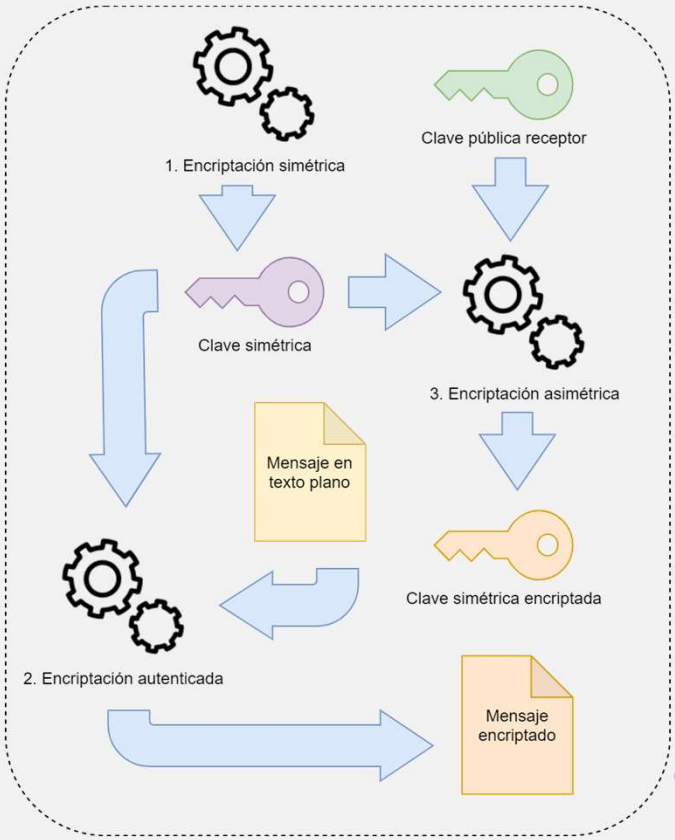
La criptografía simétrica es **menos segura** que la asimétrica

La criptografía asimétrica es **más lenta** que la simétrica

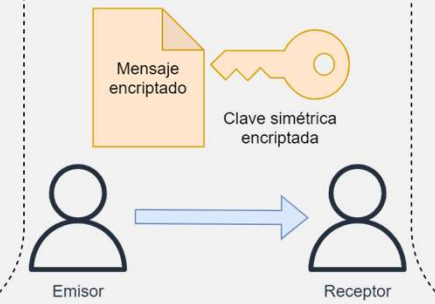
La criptografía híbrida surge del problema de ambos sistemas criptográficos

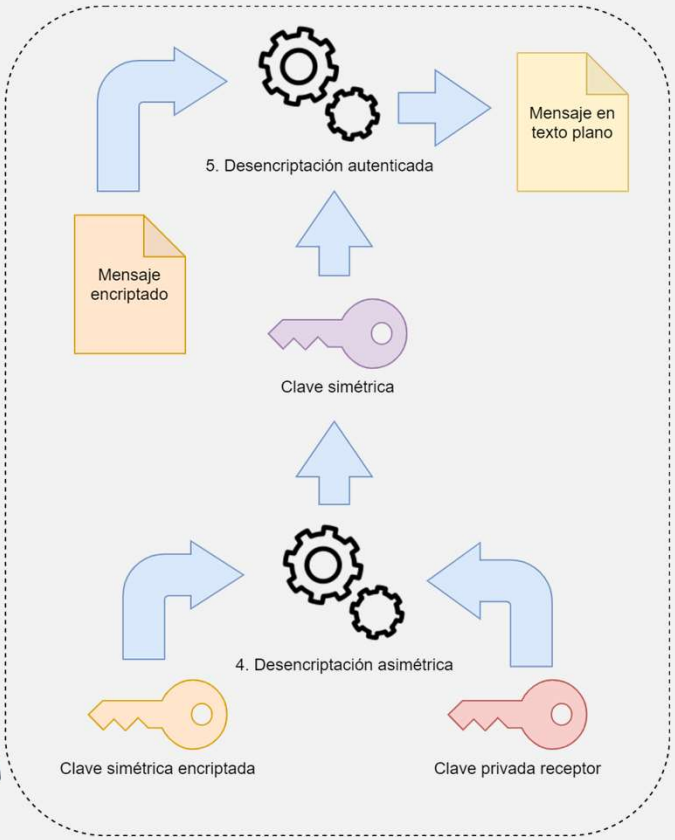
1. El **receptor** genera una **clave pública** y una **clave privada**
2. EL emisor cifra el archivo de forma **síncrona**
3. El receptor envía su clave pública
4. Se cifra la clave utilizada para encriptar el archivo con la clave pública del receptor
5. Se envía el **archivo cifrado síncronamente** y la **clave del archivo cifrada asíncronamente**

4. Criptografía híbrida



Criptografía híbrida





4. Criptografía híbrida

¿Cuándo usar criptografía simétrica, asimétrica o híbrida?

Criptografía simétrica

- **Banca**: cifrado de la información de la tarjeta de crédito u otra información de identificación personal requerida para realizar una transacción.
- **Almacenamiento de datos**: cifrado de datos almacenados en un dispositivo cuando esos datos no se están transfiriendo.

Criptografía asimétrica

- **Firmas digitales**: confirmación de la identidad para que alguien firme un documento.
- **Blockchain**: confirmación de identidad para autorizar transacciones.
- **Infraestructura de clave pública**: gobierno de claves de cifrado mediante la emisión y gestión de certificados digitales.

Criptografía híbrida

- **SSL/TLS**: criptografía asimétrica para cifrar una clave simétrica de un solo uso, que a su vez se utiliza para cifrar o descifrar el contenido de esa sesión de navegación por internet.
- **Sistemas de chat móvil**: uso de criptografía asimétrica para verificar la identidad de los participantes al inicio de la conversación y criptografía simétrica para cifrar el contenido en curso de dicha conversación.

5. Funciones resumen o hash

Diferencias entre un cifrador y un hash

Un **cifrador** es un **algoritmo que puede cifrar o descifrar datos** basados en claves públicas y privadas.
Un **hash** es una **conversión irreversible de los datos** u otra fuente de entrada.

Se utilizan para asegurar **la integridad de los mensajes**. Por ejemplo: contraseñas o checksums.

También se utilizan para la **detección de malware**.

- MD5 (Message-Digest Algorithm 5): Fue desarrollado en 1991 por Rivest en el MIT. Ampliamente utilizado hasta el descubrimiento de colisiones de hash en este algoritmo.
- SHA (Secure Hash Algorithm): SHA-3 es el algoritmo más reciente de la familia SHA (2015). Se basa en la construcción de esponjas, que consiste en una permutación aleatoria de los datos. Se utiliza en protocolos como TLS o VPN.

5. Funciones resumen o hash

Ataque de cumpleaños

El ataque de cumpleaños es un tipo de **ataque criptográfico** que rompe los algoritmos al encontrar **coincidencias en la función hash**. El método se basa en la paradoja del cumpleaños por la que la posibilidad de que dos personas compartan un cumpleaños es bastante mayor de lo que parece.

¿Cuál es el problema de la paradoja del cumpleaños?

La teoría de la probabilidad describe este problema como la probabilidad de que, dada una cierta cantidad de personas en una habitación, una o varias de esas personas compartan su cumpleaños.

¿Cuántas personas se necesitan para que haya una colisión de cumpleaños?

5. Funciones resumen o hash

Ataque de cumpleaños

La mayoría asume que se necesitan al menos 183 personas.

- No se consideran los años bisiestos.
- Se considera que cada cumpleaños tiene la misma probabilidad de ocurrencia.
- Es sencillo calcular la probabilidad de que ninguna otra persona comparta el cumpleaños.

Posibilidad de al menos una coincidencia = 1 – probabilidad de no coincidencia

- Para la primera persona hay una posibilidad de 365/365 de que no comparta cumpleaños.

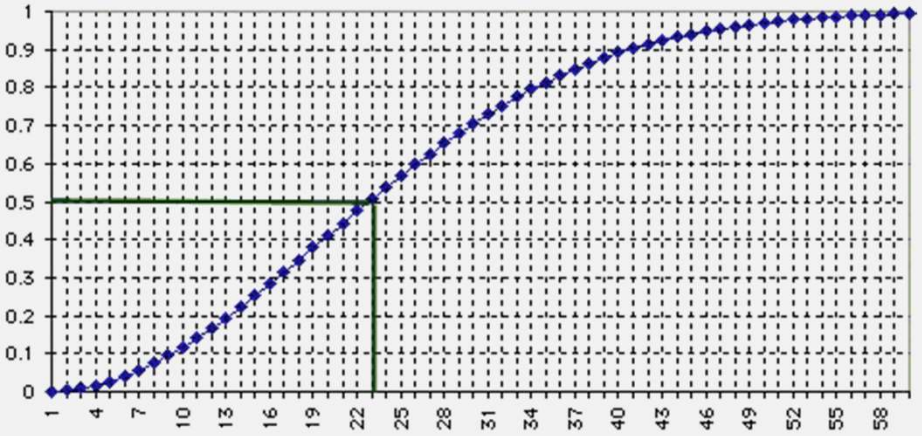
$$1 - (365 / 365) = 0\%$$

- Para tres personas sería

$$1 - (365 / 365) * (364 / 365) * (363 / 365) = 0.0082\ (0{,}82\%)$$

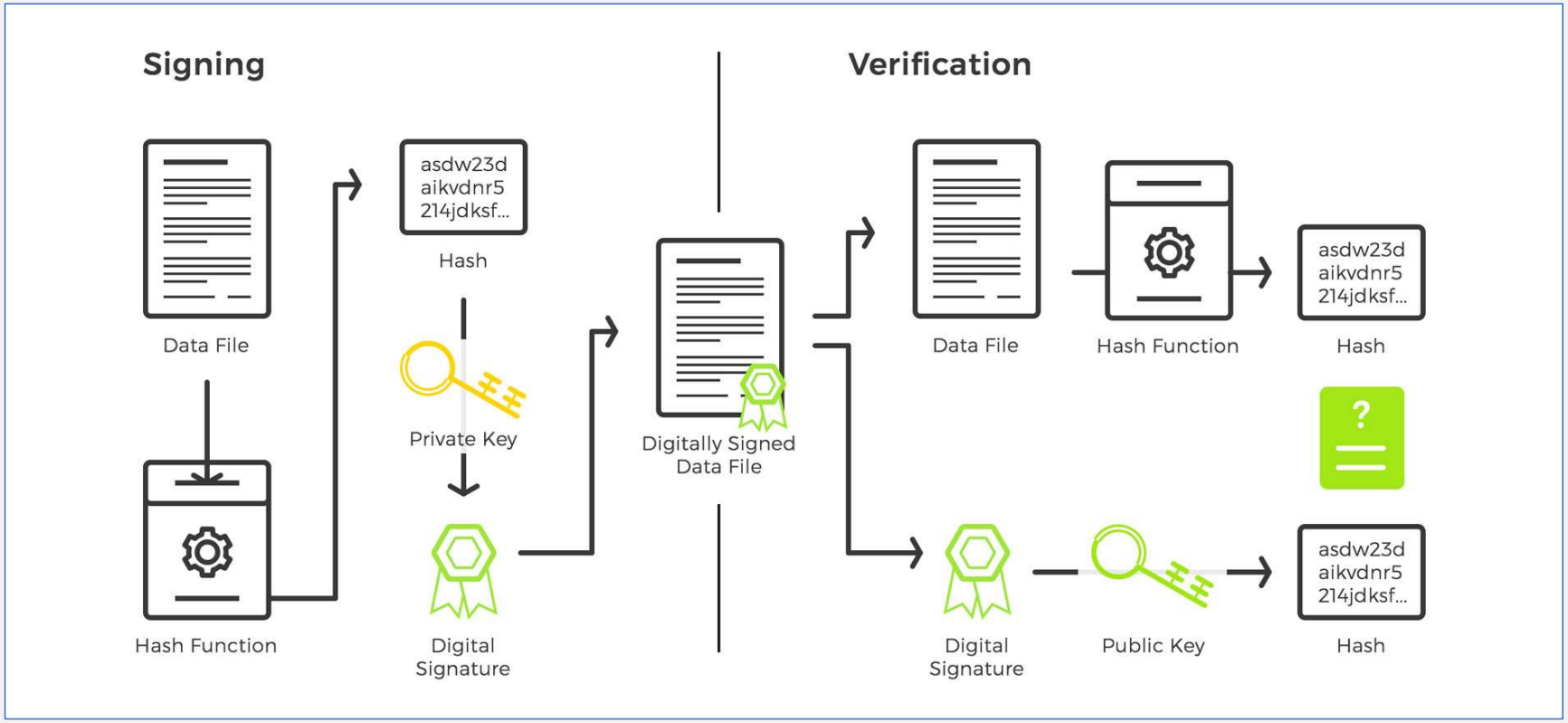
5. Funciones resumen o hash

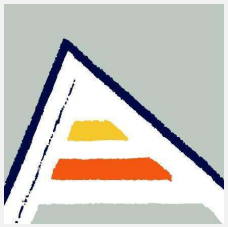
Ataque de cumpleaños



n	prob		n	prob
5	0.027		30	0.706
10	0.117		35	0.814
15	0.253		40	0.891
18	0.347		50	0.970
20	0.411		60	0.9951
23	0.507		70	0.99916
25	0.569		80	0.99991
27	0.627		90	0.99999

6. Firma digital





Grado en Ingeniería Informática

Sistemas distribuidos

Criptografía

Víctor Vives
vvives@dtic.ua.es

Departamento de Tecnología Informática y Computación

2021 - 2022