

# UT6. Criptografía.

---

---

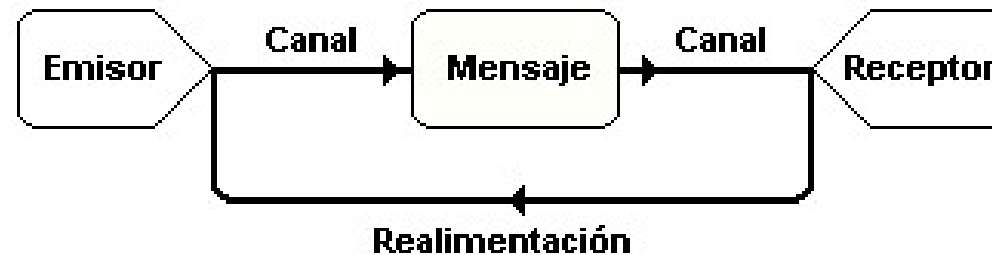
## INTRODUCCIÓN



# ¿Qué es?

**Criptografía:** la técnica que permite transformar un mensaje legible (en claro) en otro ilegible (cifrado) usando claves.

- Esto permite transmitir por un canal inseguro.
- Con ello consigo: C~~I~~DAN.




# ¿Qué es?

---

No confundir con **criptoanálisis**: tratar de descifrar el mensaje cifrado sin conocerlo. Es decir, “romper el código”.

- Ataque de fuerza bruta probando claves.

**Ningún** método de cifrado es 100% seguro:

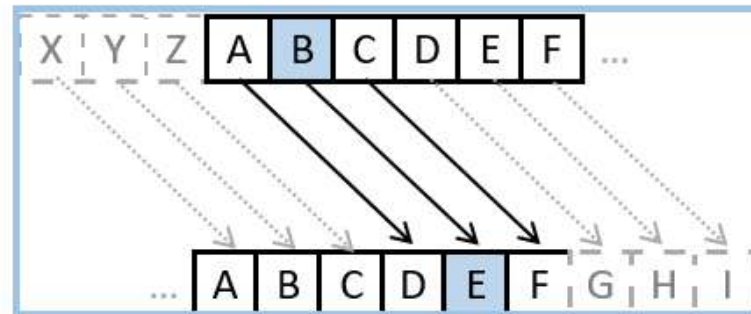
- Todos a la larga son vulnerables al criptoanálisis.
  - Diferencia entre unos y otros: el tiempo que hay que invertir.
  - Con los algoritmos más avanzados de cifrado y el ordenador más potente se tardarían años.
- 

# Ejemplo: cifrado César

¿Cuál es la clave aquí?

¿Cómo rompo el código?

¿Por qué funcionaba si es tan simple?



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

En claro

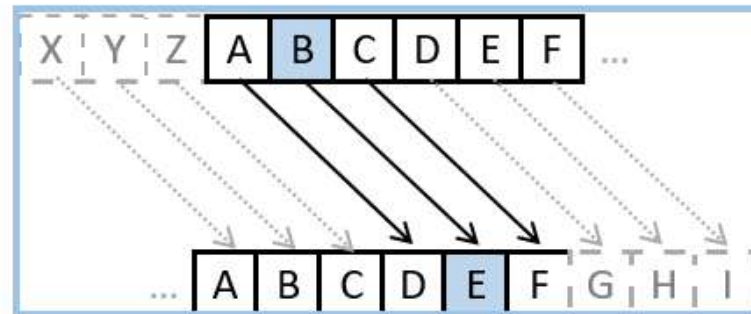
Codificado

# Ejemplo: cifrado César

¿Cuál es la clave aquí? 3

¿Cómo rompo el código? Ataque de frecuencia.

¿Por qué funcionaba si es tan simple? La sociedad no tenía los mismos conocimientos.



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

En claro

Codificado

# Ejemplos

---



---

## SISTEMAS ACTUALES DE CIFRADO





# ¿Qué es una clave y un algoritmo de cifrado?

---

- **El mensaje:** un puñado de bits (que pueden ser texto por ejemplo).
- **La clave:** un puñado de bits.
- **El algoritmo:** un conjunto de pasos en programación, como una función.

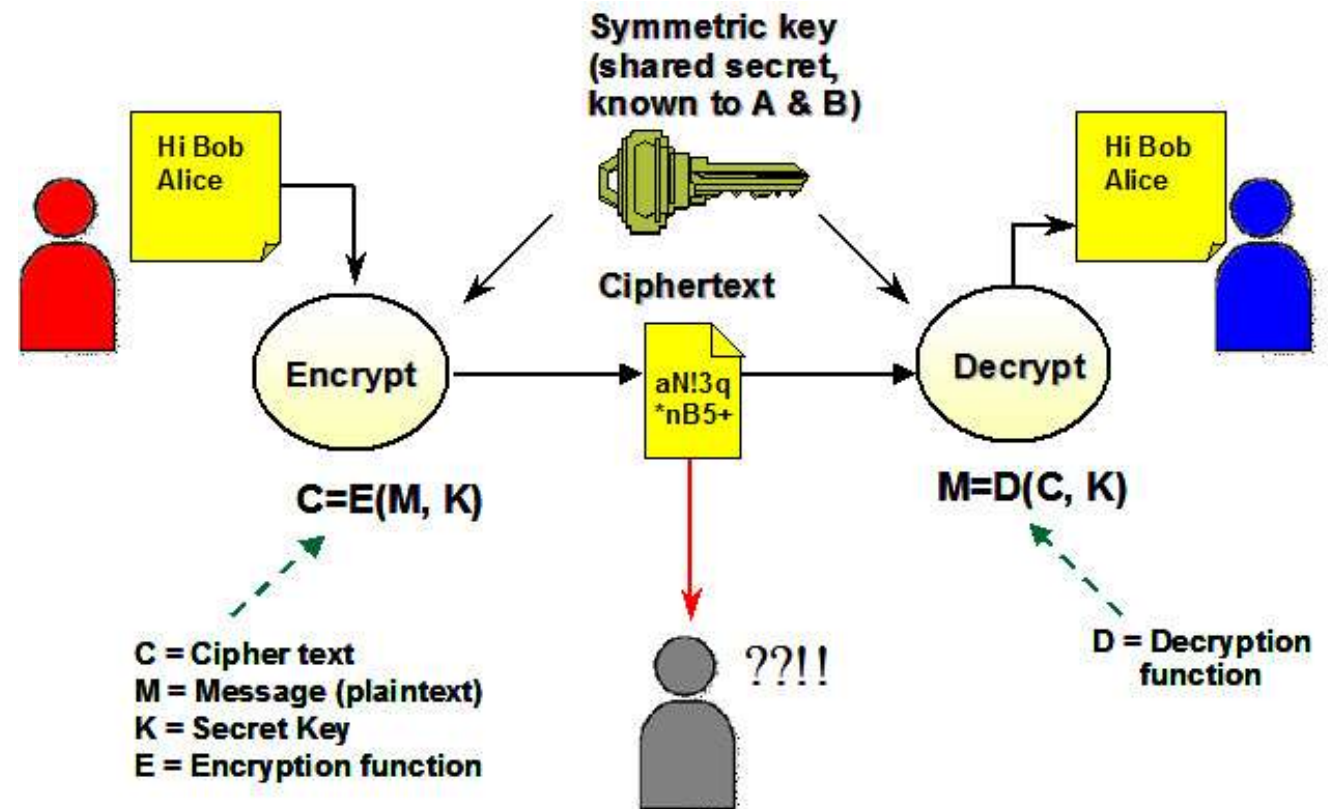
# Criptografía de algoritmos simétricos o de clave privada

Utilizan la misma clave para cifrar y descifrar el mensaje.

- De ahí que se llamen “simétricos”.

La clave solo puede conocerla el emisor y el receptor.

- De ahí que se llamen “de clave privada”.



# Criptografía de algoritmos simétricos o de clave privada

---

## **Ventajas:**

Rapidez de cómputo.

Son generalmente seguros.

## **Desventajas:**

¿?

# Criptografía de algoritmos simétricos o de clave privada

---

## Ventajas:

Rapidez de cómputo.

Son generalmente seguros.

## Desventajas:

Es necesario compartir la clave previamente y esto supone transmitirla en claro. **Necesito un canal seguro.**

Para cada persona con la que hable necesito una clave. **Acabo con demasiadas claves.**

# Criptografía de algoritmos simétricos o de clave privada

---

Ejemplos:

- **DES** (clave de 56 bits). Hay 72.057.594.037.927.936 combinaciones posibles. Un ordenador puede atacar por fuerza bruta en poco tiempo.
- **3DES, Blowfish e IDEA** (128 bits).
- **AES** (128, 192 y 256 bits).

**A priori, ¿cuál es mejor?**

---

Solución...

A solid blue horizontal bar spanning the width of the slide, located at the bottom.

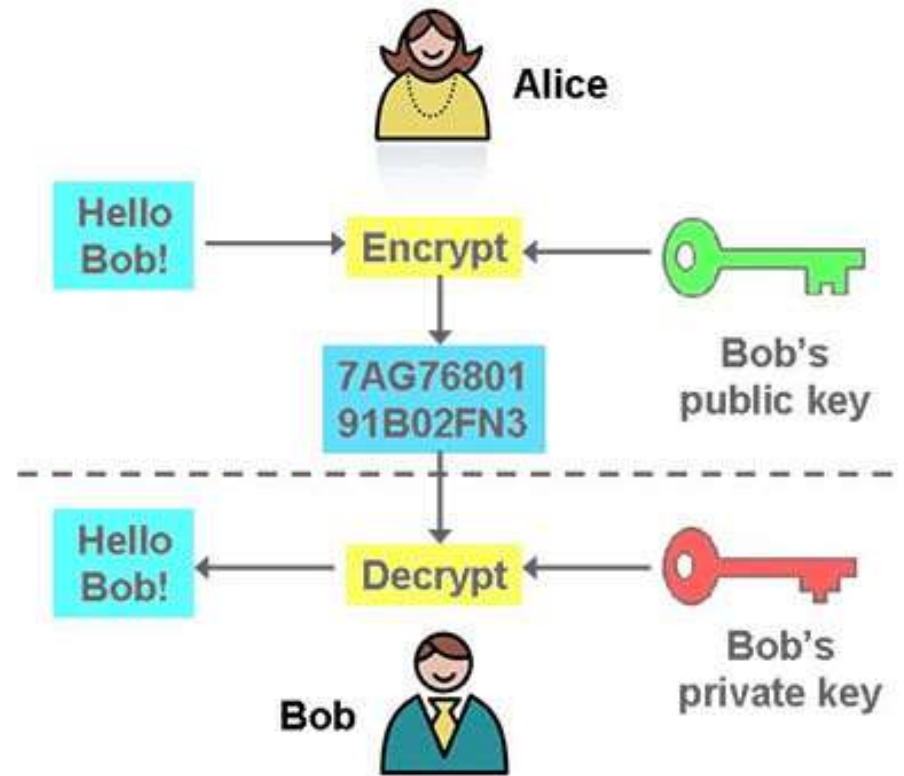
# Criptografía de algoritmos asimétricos o de clave pública

Cada interlocutor tiene **un par de claves**: lo que cifra una SOLO se puede descifrar con la otra.

- De ahí que se llamen algoritmos “asimétricos”.

Dentro de cada par llamamos a una “**clave privada**” y a la otra “**clave pública**”.

- De ahí que se llamen de clave pública.



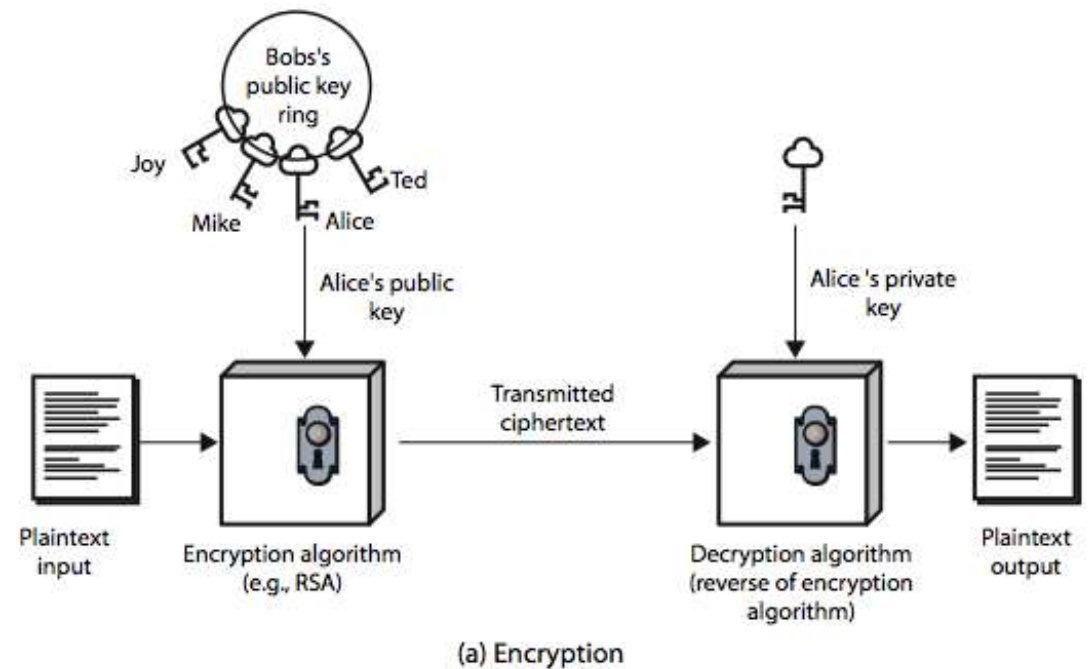
# Criptografía de algoritmos asimétricos o de clave pública

¿Y qué hago con las claves?

1. La clave privada me la guardo.
2. La clave publica la distribuyo a todo el mundo.

¿Qué pasa si alguien usa mi clave pública?

¿Qué pasa si alguien me roba mi clave privada?





# Criptografía de algoritmos asimétricos o de clave pública

---

# Criptografía de algoritmos asimétricos o de clave pública

---

Ejemplos:

- **RSA, Diffie-Hellman, ECC**

Ejemplo de uso: SSH.

La robustez de los algoritmos de clave pública está en:

- Que la clave sea muy larga (muchos bits)
- Que es complicado factorizar números primos.

---

Solución...

A solid blue horizontal bar spanning the width of the slide, located at the bottom.

# Criptografía híbrida

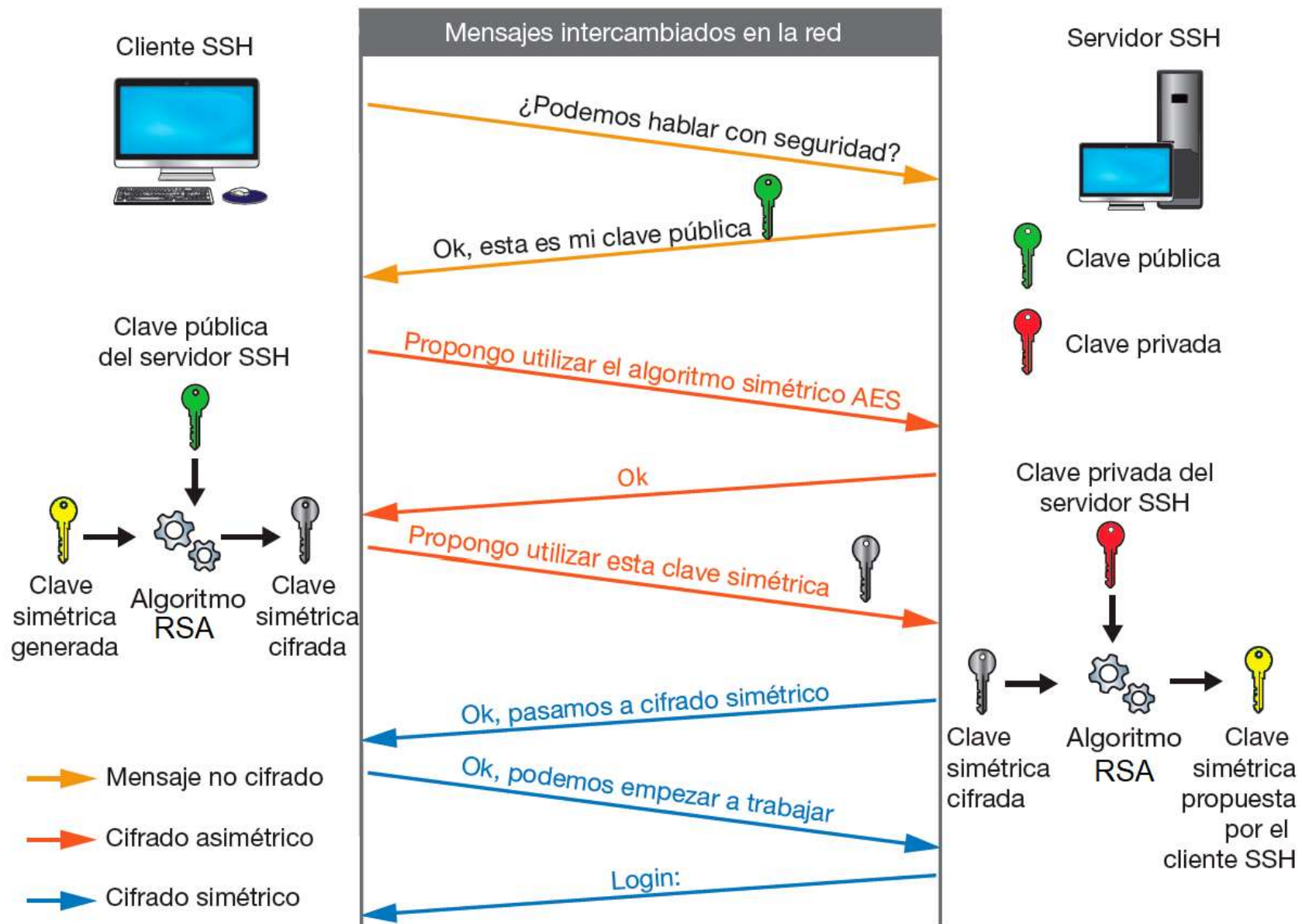
---

Usamos **criptografía de clave pública** al comienzo nada más.

- Creamos así un canal seguro.
- Lo usamos para intercambiar una clave privada.

Usamos la **criptografía de clave privada** el resto del tiempo.

- Y así el coste computacional es óptimo.



# Resumen de algoritmos

---

Idealmente la fortaleza de un sistema de cifrado debe recaer en la **clave**, **NO** en el **algoritmo**.

- Aunque el algoritmo sea público (la mayoría lo son), si no conocemos la clave, no podemos descifrar.
- En general, cuanto más larga la clave, mejor.

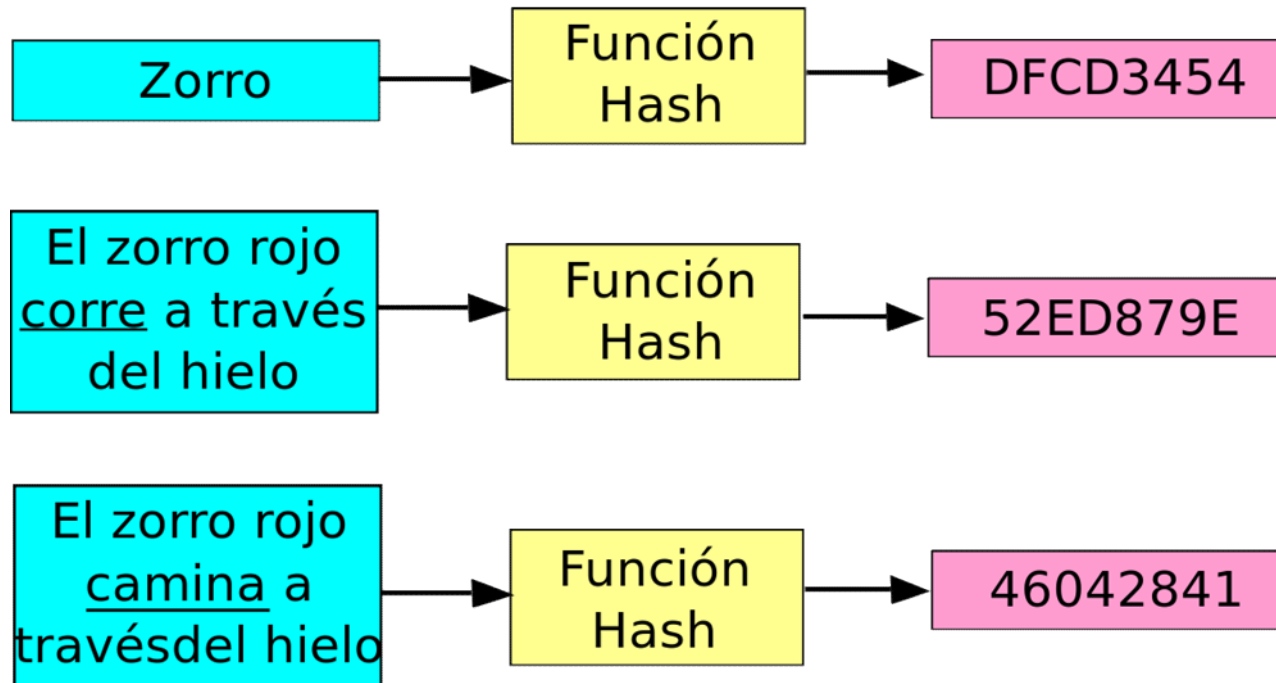
---

## FUNCIONES HASH



# Funciones hash o resumen

---

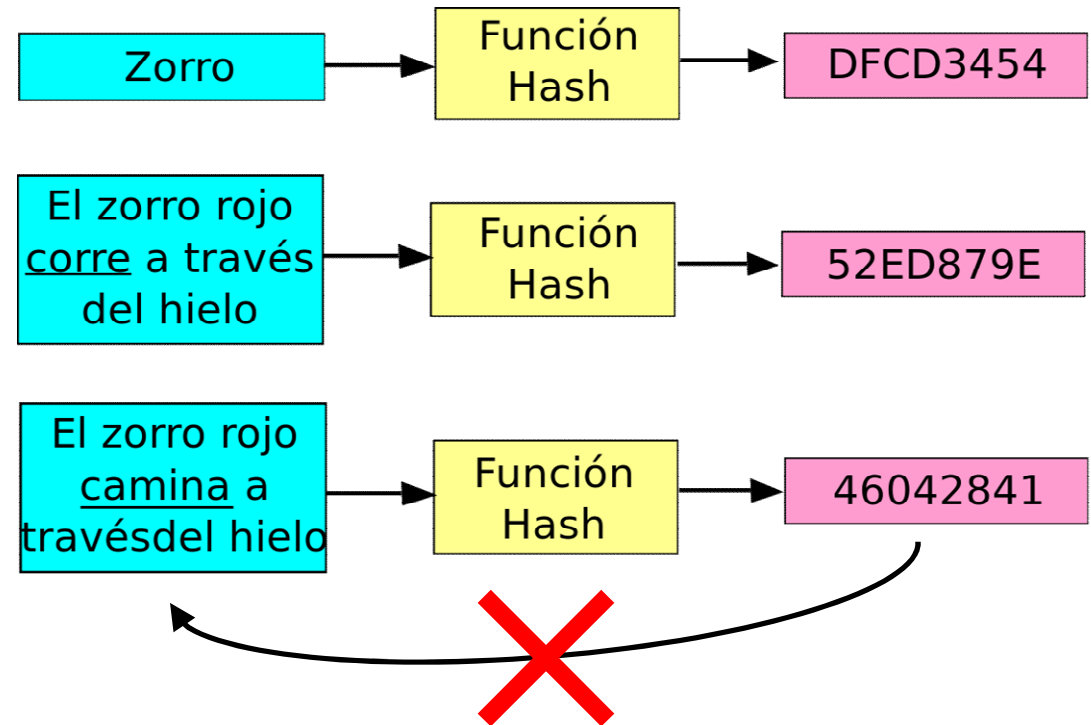




# Funciones hash o resumen

## Características:

- Ante la misma entrada producen siempre la misma salida.
- A partir de la salida **NO** se puede conocer la entrada.
- La salida tiene una longitud fija (como 64 bits, por ejemplo).
- Un pequeño cambio a la entrada altera mucho la salida.



# Funciones hash o resumen

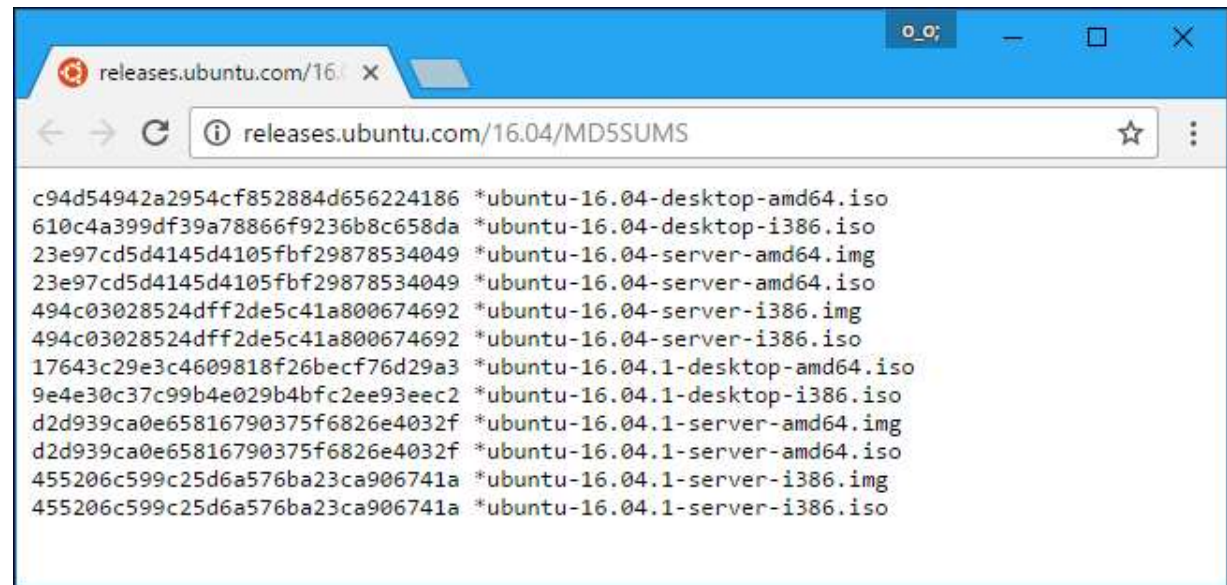
---

**Usos:**

# Funciones hash o resumen

## Usos:

- Comprobación integridad (*checksum*).
- Guardado de contraseñas
- Huellas (comparar documentos, detección de virus, etc.)
- Firma digital.



The screenshot shows a web browser window with the address bar displaying "releases.ubuntu.com/16.04/MD5SUMS". The page content lists MD5 checksums for various Ubuntu 16.04 release images, including desktop and server versions for amd64 and i386 architectures. The text is as follows:

```
c94d54942a2954cf852884d656224186 *ubuntu-16.04-desktop-amd64.iso
610c4a399df39a78866f9236b8c658da *ubuntu-16.04-desktop-i386.iso
23e97cd5d4145d4105fbf29878534049 *ubuntu-16.04-server-amd64.img
23e97cd5d4145d4105fbf29878534049 *ubuntu-16.04-server-amd64.iso
494c03028524dff2de5c41a800674692 *ubuntu-16.04-server-i386.img
494c03028524dff2de5c41a800674692 *ubuntu-16.04-server-i386.iso
17643c29e3c4609818f26becf76d29a3 *ubuntu-16.04.1-desktop-amd64.iso
9e4e30c37c99b4e029b4bfc2ee93eec2 *ubuntu-16.04.1-desktop-i386.iso
d2d939ca0e65816790375f6826e4032f *ubuntu-16.04.1-server-amd64.img
d2d939ca0e65816790375f6826e4032f *ubuntu-16.04.1-server-amd64.iso
455206c599c25d6a576ba23ca906741a *ubuntu-16.04.1-server-i386.img
455206c599c25d6a576ba23ca906741a *ubuntu-16.04.1-server-i386.iso
```

# Funciones hash o resumen

---

**Ejemplos:** MD5, SHA, SHA-1

## **Ataques y problemas:**

- Dos entradas diferentes pueden generar la misma salida (a esto lo llamamos **colisión**). <https://www.mscs.dal.ca/~selinger/md5collision/>
- La gente crea sus diccionarios metiendo en la función hash las contraseñas más usadas y obtiene una lista con sus hashes.  
<https://md5.gromweb.com/>

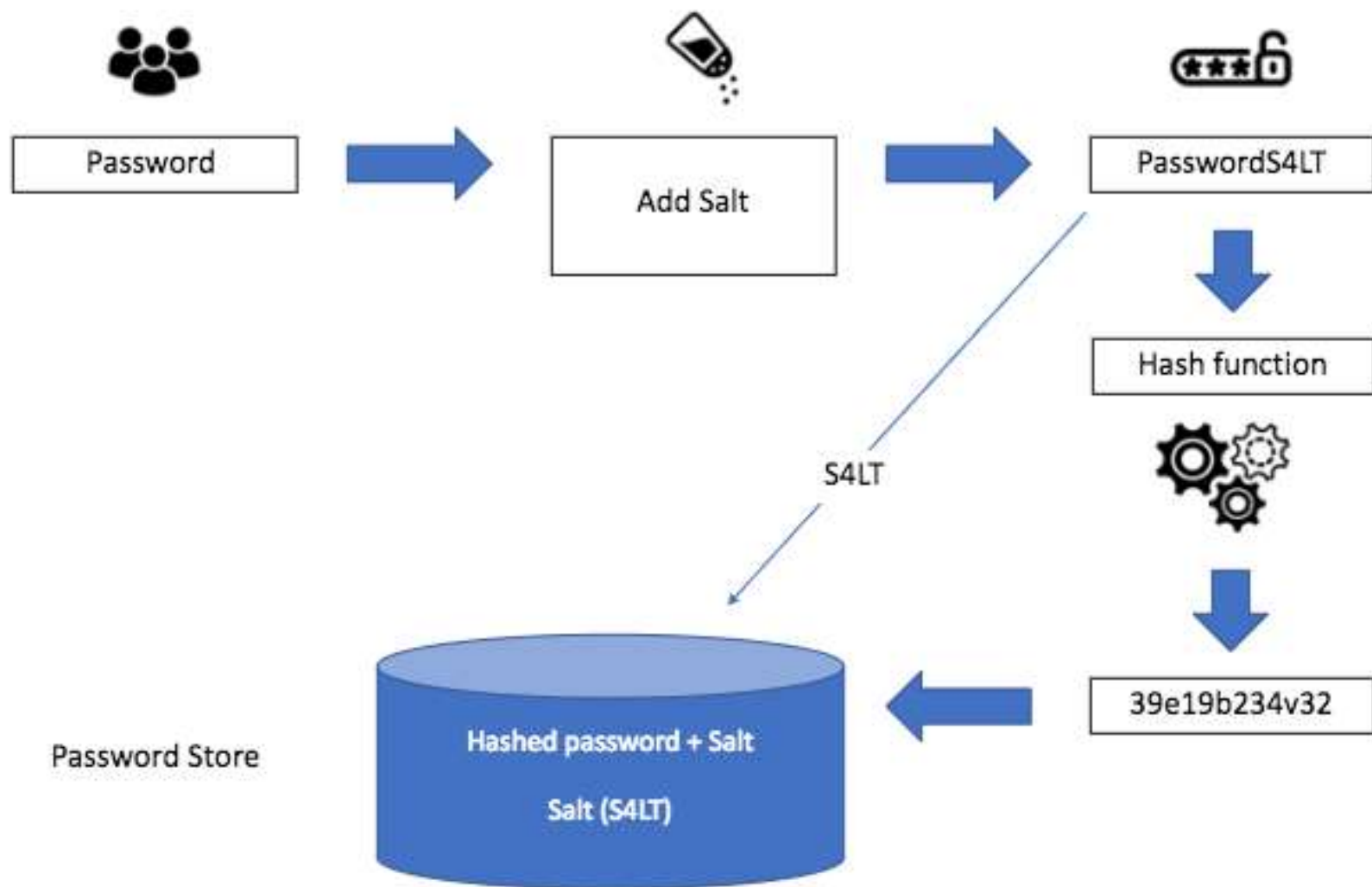
# Funciones hash o resumen

---

**Ejemplos:** MD5, SHA, SHA-1

## **Ataques y problemas:**

- Dos entradas diferentes pueden generar la misma salida (a esto lo llamamos **colisión**). <https://www.mscs.dal.ca/~selinger/md5collision/>
- La gente crea sus diccionarios metiendo en la función hash las contraseñas más usadas y obtiene una lista con sus hashes.  
<https://md5.gromweb.com/>
  - **Solución:** “saltar” las contraseñas. Añadir caracteres adicionales por nuestra cuenta.



---

## FIRMA DIGITAL

---

# CERTIFICADOS DIGITALES



# Por qué

---

## Problemas:

1. A la hora de firmar necesitamos asegurarnos de que la **clave privada** del usuario que firma SOLO la tiene él.
2. Necesitamos distinguir de manera fácil DE QUIÉN es cada **clave pública**.

### Search results for 'iesclaradelrey'

Type	bits/keyID	Date	User ID
pub	1024R/ <a href="#">57930C9D</a>	2021-09-30	<a href="#">AlvaroP IESClaraDelRey &lt;AlvaroP@IESClaraDelRey.es&gt;</a>
pub	1024R/ <a href="#">FC64DEA5</a>	2021-09-30	<a href="#">IvanG IESClaraDelRey &lt;IvanG@IESClaraDelRey.es&gt;</a>
pub	1024R/ <a href="#">A9F83A70</a>	2021-09-30	<a href="#">AliciaR IESClaraDelRey &lt;AliciaR@IESClaraDelRey.es&gt;</a>
pub	1024R/ <a href="#">D978E27B</a>	2021-09-30	<a href="#">AndresD IESClaraDelRey &lt;AndresD@IESClaraDelRey&gt;</a>

# Por qué

---

**Solución:** que un tercero de confianza firme digitalmente la clave pública (y datos asociados a la misma).

# Por qué

---

**Solución:** que un tercero de confianza firme digitalmente la clave pública (y datos asociados a la misma).

**Camerfirma**  
Certificado Digital

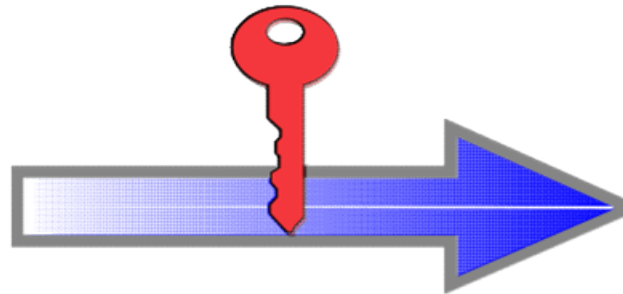


 **Real Casa de la Moneda**  
Fábrica Nacional  
de Moneda y Timbre

## Identity Information and Public Key of Mario Rossi



Certificate Authority  
verifies the identity of Mario Rossi  
and encrypts with its Private Key



## Certificate of Mario Rossi



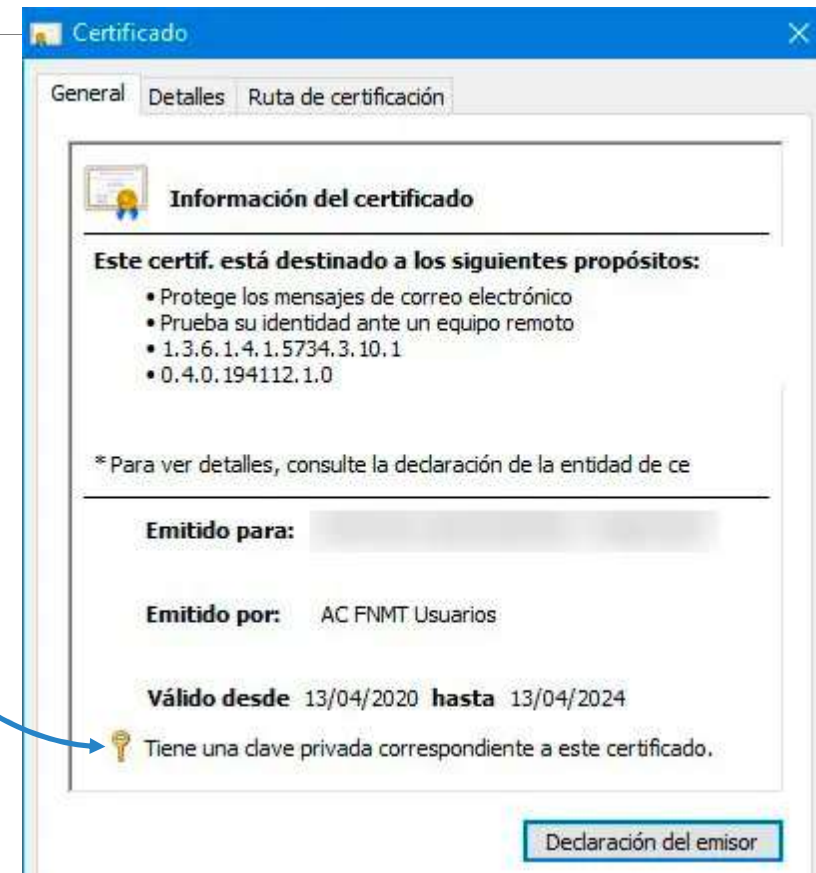
Digitally Signed by  
Certificate Authority

# Certificado digital

Suele llevar:

- La clave pública del usuario.
- **Opcionalmente** la privada.
- Datos (DNI, nombre, email, etc.)
- Quién emite el certificado.
- Validez.

Formato habitual: X.509



## Formato X.509:

### Certificate:

#### Data:

Version: 1 (0x0)  
Serial Number: 7829 (0x1e95)  
Signature Algorithm: md5WithRSAEncryption  
Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,  
OU=Certification Services Division,  
CN=Thawte Server CA/Email=server-certs@thawte.com

#### Validity

Not Before: Jul 9 16:04:02 1998 GMT

Not After : Jul 9 16:04:02 1999 GMT

Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,  
OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org

#### Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:  
33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:  
66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:  
70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:  
16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:  
c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:  
8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:  
d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:  
e8:35:1c:9e:27:52:7e:41:8f

Exponent: 65537 (0x10001)

#### Signature Algorithm: md5WithRSAEncryption

93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:  
92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:  
ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:  
d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:  
0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:  
5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:  
8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:  
68:9f



## Formato X.509:

## Parte de datos

## La firma de la parte de datos

### Certificate:

#### Data:

```
Version: 1 (0x0)
Serial Number: 7829 (0x1e95)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
        OU=Certification Services Division,
        CN=Thawte Server CA/Email=server-certs@thawte.com

Validity
  Not Before: Jul  9 16:04:02 1998 GMT
  Not After : Jul  9 16:04:02 1999 GMT
Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
        OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (1024 bit)
    Modulus (1024 bit):
      00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
      33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
      66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
      70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
      16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
      c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
      8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
      d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
      e8:35:1c:9e:27:52:7e:41:8f
    Exponent: 65537 (0x10001)
```

```
Signature Algorithm: md5WithRSAEncryption
93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
68:9f
```

Formato X.509:

Parte de datos

La firma de la  
parte de datos

Certificate:

Data:

Version: 1 (0x0)

Serial Number: 7829 (0x1e95)

Signature Algorithm: md5WithRSAEncryption

Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,  
OU=Certification Services Division,  
CN=Thawte Server CA/Email=server-certs@thawte.com

Validity

Not Before: Jul 9 16:04:02 1998 GMT

Not After : Jul 9 16:04:02 1999 GMT

Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,

OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:  
33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:  
66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:  
70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:  
16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:  
c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:  
8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:  
d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:  
e8:35:1c:9e:27:52:7e:41:8f

Exponent: 65537 (0x10001)

Signature Algorithm: md5WithRSAEncryption

93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:  
92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:  
ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:  
d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:  
0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:  
5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:  
8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:  
68:9f

Quién firma



Formato X.509:

Parte de datos

La firma de la parte de datos

```
Certificate:
Data:
  Version: 1 (0x0)
  Serial Number: 7829 (0x1e95)
  Signature Algorithm: md5WithRSAEncryption
  Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
         OU=Certification Services Division,
         CN=Thawte Server CA/Email=server-certs@thawte.com
  Validity
    Not Before: Jul  9 16:04:02 1998 GMT
    Not After : Jul  9 16:04:02 1999 GMT
  Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
         OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
      Modulus (1024 bit):
        00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
        33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
        66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
        70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
        16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
        c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
        8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
        d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
        e8:35:1c:9e:27:52:7e:41:8f
      Exponent: 65537 (0x10001)
  Signature Algorithm: md5WithRSAEncryption
    93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
    92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
    ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
    d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
    0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
    5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
    8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
    68:9f
```

Quién firma

Validez

Formato X.509:

Parte de datos

La firma de la parte de datos

```
Certificate:
Data:
  Version: 1 (0x0)
  Serial Number: 7829 (0x1e95)
  Signature Algorithm: md5WithRSAEncryption
  Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
         OU=Certification Services Division,
         CN=Thawte Server CA/Email=server-certs@thawte.com
  Validity
    Not Before: Jul  9 16:04:02 1998 GMT
    Not After : Jul  9 16:04:02 1999 GMT
  Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
         OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
      Modulus (1024 bit):
        00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
        33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
        66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
        70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
        16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
        c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
        8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
        d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
        e8:35:1c:9e:27:52:7e:41:8f
      Exponent: 65537 (0x10001)
  Signature Algorithm: md5WithRSAEncryption
    93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
    92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
    ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
    d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
    0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
    5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
    8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
    68:9f
```

Quién firma

Validez

Quién soy

Formato X.509:

Parte de datos

La firma de la parte de datos

Certificate:

Data:

Version: 1 (0x0)

Serial Number: 7829 (0x1e95)

Signature Algorithm: md5WithRSAEncryption

Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,  
OU=Certification Services Division,  
CN=Thawte Server CA/Email=server-certs@thawte.com

Validity

Not Before: Jul 9 16:04:02 1998 GMT

Not After : Jul 9 16:04:02 1999 GMT

Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,  
OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:  
33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:  
66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:  
70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:  
16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:  
c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:  
8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:  
d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:  
e8:35:1c:9e:27:52:7e:41:8f

Exponent: 65537 (0x10001)

Signature Algorithm: md5WithRSAEncryption

93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:  
92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:  
ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:  
d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:  
0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:  
5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:  
8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:  
68:9f

Quién firma

Validez

Quién soy

Mi clave pública



# ¿Y si me lo firmo yo?

## Certificado autofirmado:

El propietario del certificado se lo firma él mismo con su clave privada.

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
           OU=Certification Services Division,
           CN=Thawte Server CA/Email=server-certs@thawte.com
    Validity
      Not Before: Aug  1 00:00:00 1996 GMT
      Not After : Dec 31 23:59:59 2020 GMT
    Subject: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
           OU=Certification Services Division,
           CN=Thawte Server CA/Email=server-certs@thawte.com
```

Iguales

# ¿Qué se hace con él?

Se extrae la clave (o claves) que contiene y se utiliza de manera normal, como hemos visto hasta ahora.



## Certificate:

### Data:

Version: 1 (0x0)  
Serial Number: 7829 (0x1e95)  
Signature Algorithm: md5WithRSAEncryption  
Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,  
OU=Certification Services Division,  
CN=Thawte Server CA/Email=server-certs@thawte.com

### Validity

Not Before: Jul 9 16:04:02 1998 GMT

Not After : Jul 9 16:04:02 1999 GMT

Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,  
OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org

### Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):



00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:  
33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:  
66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:  
70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:  
16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:  
c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:  
8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:  
d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:  
e8:35:1c:9e:27:52:7e:41:8f

Exponent: 65537 (0x10001)

### Signature Algorithm: md5WithRSAEncryption

93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:  
92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:  
ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:  
d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:  
0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:  
5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:  
8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:  
68:9f

# ¿Cómo se verifica si es válido?

1. Hago el hash a la parte de “Data”. 
2. Aplico la clave pública del emisor a la firma. 
3. ¿Son iguales?

**La clave pública del emisor la saco de otro certificado.**

## Certificate:

### Data:

Version: 1 (0x0)  
Serial Number: 7829 (0x1e95)  
Signature Algorithm: md5WithRSAEncryption  
Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,  
OU=Certification Services Division,  
CN=Thawte Server CA/Email=server-certs@thawte.com

### Validity

Not Before: Jul 9 16:04:02 1998 GMT

Not After : Jul 9 16:04:02 1999 GMT

Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,  
OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org

### Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:  
33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:  
66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:  
70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:  
16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:  
c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:  
8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:  
d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:  
e8:35:1c:9e:27:52:7e:41:8f

Exponent: 65537 (0x10001)

### Signature Algorithm: md5WithRSAEncryption

93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:  
92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:  
ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:  
d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:  
0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:  
5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:  
8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:  
68:9f

---

INFRAESTRUCTURA DE CLAVE PÚBLICA



# RA: autoridad de registro

---

Una **RA** (autoridad de registro) es el organismo que se encarga de comprobar la identidad de la persona que va a pedir un certificado.

Ejemplo: <https://www.sede.fnmt.gob.es/certificados/persona-fisica/obtener-certificado-software/acreditar-identidad>



# CA: autoridad de certificación

---

Una **CA** (autoridad de certificación) es un organismo que vincula una clave con una entidad (persona, empresa, dominio web, etc.)

Es decir:

- Es un organismo que tiene su propio par de claves pública y privada.
- Se dedica a firmar las claves de otros (o sea, emitir certificados).



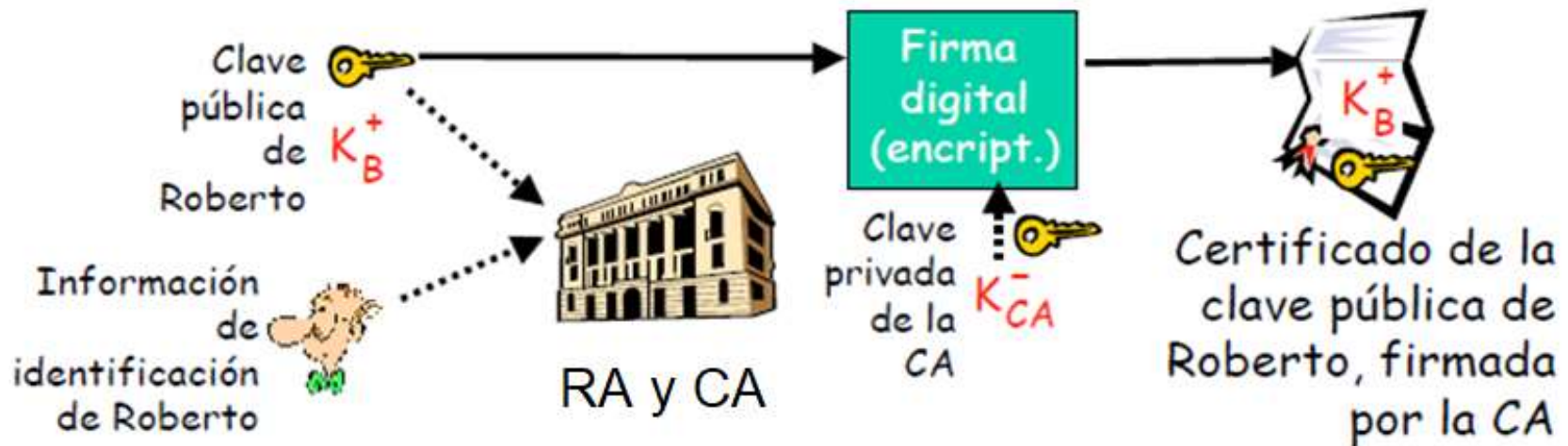
# Solicitando el certificado

1º Ir a una RA a verificar mi identidad.

2º La RA le dice a la CA que efectivamente soy yo.

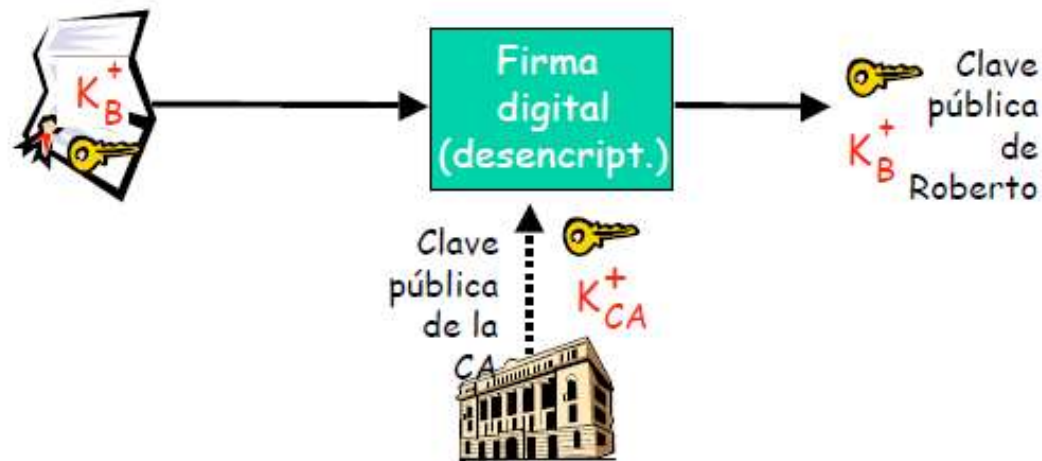
3º La CA me emite el certificado:

- **Opción A:** ya tengo un par de claves → me firma la pública.
- **Opción B:** no tengo ningún par de claves (lo normal) → me las genera y me las firma.



# Usando los certificados

- Cuando Alicia quiere la clave pública de Roberto, ya no tiene que pedírsela a él, sino que obtiene el certificado (de Roberto, de Internet, de otra persona...) .
- Tan sólo tiene que aplicar **la clave pública de la CA** al certificado de Roberto para obtener la clave pública de Roberto. .



# Usando los certificados

La CA tiene su propio certificado público que se denomina **certificado raíz** y es **autofirmado**:

El ordenador lo usa para “fiarse” de los certificados que ha generado esa CA.

- Contiene la clave de la CA.



## Certificado de entidad final

Nombre del propietario
Clave pública del propietario
Nombre del emisor (CA)
Firma del emisor

*referencia*

## Certificado Intermedio

Nombre del propietario (CA)
Clave pública del propietario
Nombre del emisor (CA raíz)
Firma del emisor

*firma*

*referencia*

*firma*

*auto-firmado*

Nombre de la CA raíz
Clave pública de la CA raíz
Firma de la CA raíz

## Certificado raíz

# Caducidad y revocación

---

Los certificados tienen un **período de validez**.

Si dejan de ser válidos dentro de ese período, es necesario revocarlos.

Razones para la revocación:

- Se sospecha que la clave privada del usuario está comprometida.
- Se sospecha que el certificado de la AC está comprometido.

# 4 Protocolos seguros.

---

SSL

TLS

HTTPS

# 4.SSL Secure Sockets Layer

---

Que es un certificado SSL:

- <https://www.genbeta.com/seguridad/que-es-un-certificado-ssl-y-por-que-deberia-importarte>

**SSL v1** nunca se llego a publicar.

**SSL v2** se presentó en febrero de 1995 pero "contenía una cantidad de fallas de seguridad que al final llevaron al diseño de la versión SSL 3.0"

**SSL v3** las versiones má

En octubre de 2014, se detectó una nueva vulnerabilidad sobre el protocolo SSL en su versión 3.0, la Vulnerabilidad de Poodles nuevas de SSL/TLS están basadas en SSL 3.0.



## 4.SSL Secure Sockets Layer (II)

---

SSL proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, solo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar.

SSL implica una serie de fases básicas:

- Negociar entre las partes el algoritmo que se usará en la comunicación
- Intercambio de claves públicas y autenticación basada en certificados digitales.
- Cifrado del tráfico basado en cifrado simétrico.

Durante la primera fase, el cliente y el servidor negocian qué algoritmos criptográficos se van a usar. Las implementaciones actuales proporcionan las siguientes opciones:

- Para criptografía de clave pública: RSA, Diffie-Hellman, DSA (*Digital Signature Algorithm*) o Fortezza.
- Para cifrado simétrico: RC2, RC4, IDEA (*International Data Encryption Algorithm*), DES (*Data Encryption Standard*), Triple DES y AES (*Advanced Encryption Standard*).
- Con funciones hash: MD5 o de la familia SHA.

# 4.TLS Transport Layer Security

---

Sucede a SSL tras su varios fallos de seguridad.

**TLS v1.0** fue definido en 1999 y es una actualización de SSL versión 3.0. Como dice el RFC, "las diferencias entre este protocolo y SSL 3.0 no son dramáticas, pero son suficientemente significativas como para impedir la integración entre ambos sistemas.

**TLS v1.1 v1.2 v1.3** son actualizaciones del protocolos en el que modifican diferentes parámetros y procedimientos, siendo estas 3 versiones funcionales.

El TLS es la siguiente generación del Certificado SSL : permite y garantiza el intercambio de datos en un entorno securizado y privado entre dos entes, el usuario y el servidor, mediante aplicaciones como HTTP, POP3, IMAP, SSH, SMTP o NNTP. Nos referimos al TLS como la evolución del SSL dado que está basado en éste último certificado y funciona de manera muy similar, básicamente: encripta la información compartida.

# 4.TLS Transport Layer Security (II)

---

## **HTTPS**

- Simplemente es una combinación del protocolo HTTP (usado en cada transacción web) con el protocolo SSL/TLS usada para establecer comu

## **CERTIFICADO DIGITAL SSL/TLS**

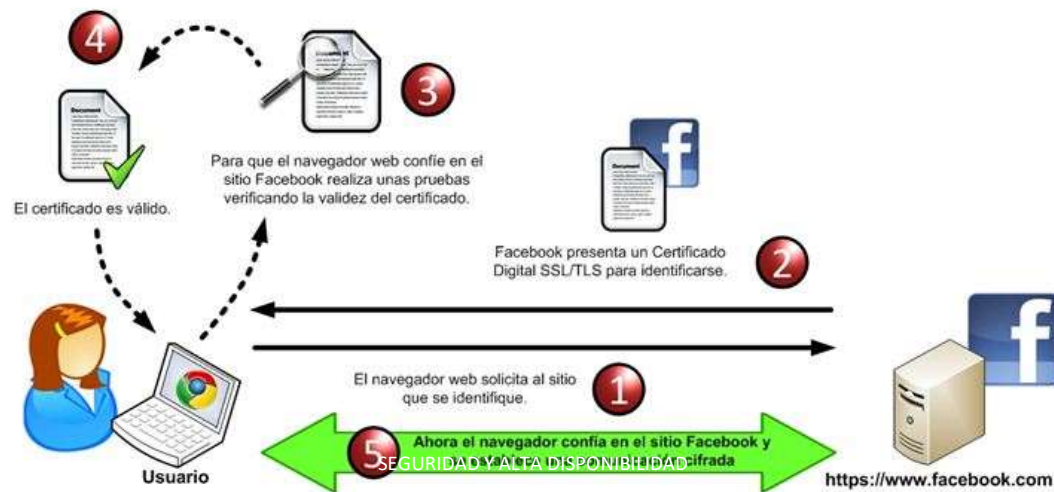
- Es un documento digital único que garantiza la vinculación entre una persona o entidad con su llave pública.
- Contiene información de su propietario como nombre, dirección, correo electrónico, organización a la que pertenece y su llave pública, así como información propia del certificado por mencionar: periodo de validez, número de serie único, nombre de la AC que emitió, firma digital de la AC cifrada con su llave privada y otros datos más que indican cómo puede usarse ese certificado.

## 4 TLS Transport Layer Security (III)

El navegador hace una petición al sitio seguro de Facebook, éste envía un mensaje donde indica que quiere establecer una conexión segura y envía datos sobre la versión del protocolo SSL/TLS que soporta y otros parámetros necesarios.

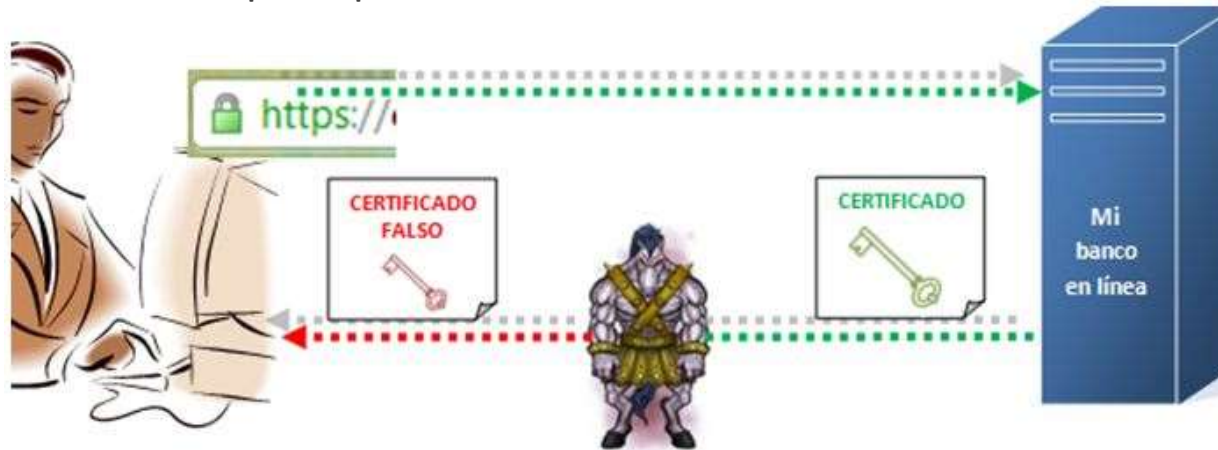
En base a esta información enviada por el navegador, el servidor web de Facebook responde con un mensaje informando que está de acuerdo en establecer la conexión segura con los datos de SSL/TLS proporcionados.

Una vez que ambos conocen los parámetros de conexión, el sitio de Facebook presenta su certificado digital al navegador web para identificarse como un sitio confiable.



## 4. Ataque a conexiones SSL/TLS

Supongamos que deseas realizar una operación bancaria en línea. Al ingresar a la página web de tu banco, durante el proceso de conexión SSL/TLS, el banco envía a tu navegador su certificado y su llave pública firmados, elementos que utilizará para cifrar la información a transmitir. El troyano se interpone entre el servidor del banco y tu navegador tomando la llave pública y la información del certificado para cifrar su propio canal de comunicación, mientras tanto, del lado del navegador el troyano inserta su certificado auto-firmado (certificado falso) de tal manera que el “*candadito de seguridad*” siempre está visible durante la conexión y así la presencia del troyano resulta imperceptible.



## 4. Ataque a conexiones SSL/TLS (II)

---

Evita hacer uso de computadoras y redes públicas, sobre todo si vas a utilizar el servicio de banca en línea.

Usar un antivirus y procura mantenerlo actualizado.

Es importante mantener actualizado tu navegador, ya que si no lo haces, eres más susceptible a ataques

Cuando utilices HTTPS, verifica la vigencia del certificado, esto lo puedes hacer observando en el periodo de validez del mismo.