

SEGURIDAD DE LA INFORMACIÓN

TEMA 3

**ESQUEMAS, PROTOCOLOS Y MECANISMOS
DE SOPORTE
(A LA SEGURIDAD DE APLICACIONES Y DE REDES)**

Indice del tema

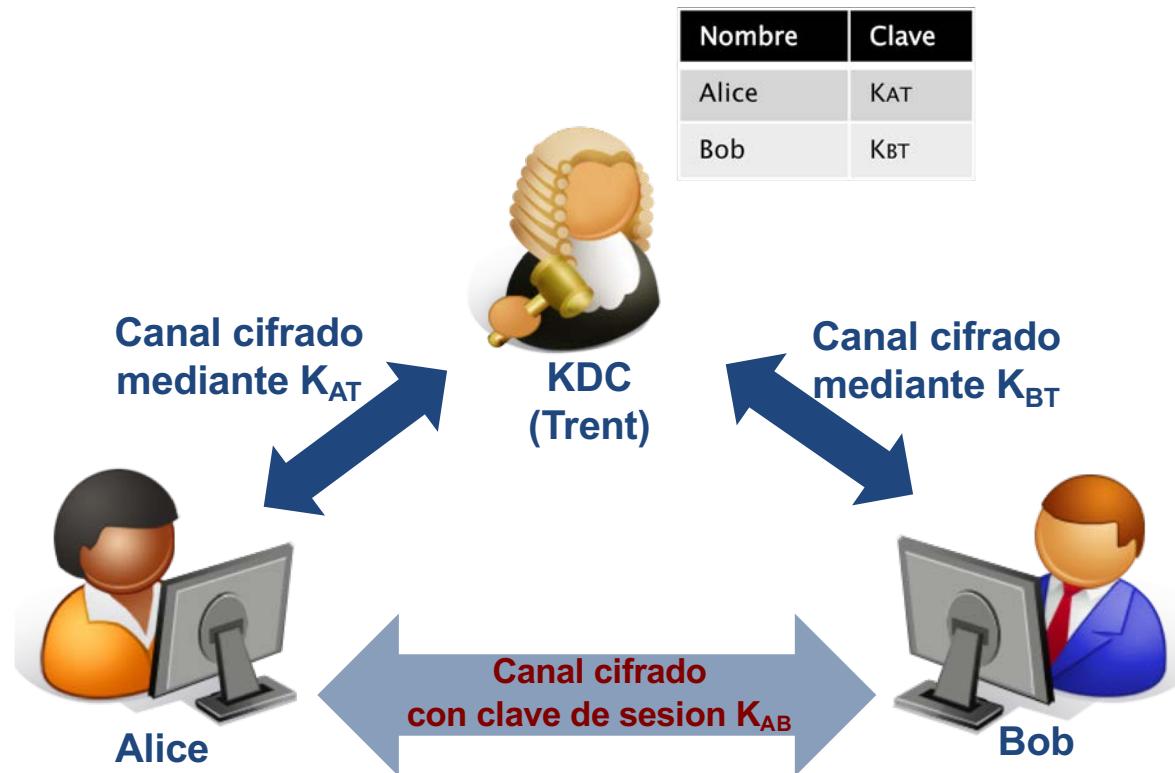
- **Gestión de las Claves**
 - Protocolos de distribución de claves simétricas
 - Mecanismos e infraestructuras de administración de claves públicas
 - El caso del DNI-e
 - Mecanismo de Single Sign-On para Autenticación
- **Mecanismos de Control de Acceso**
 - DAC
 - MAC
 - RBAC
 - ABAC
 - Otros
- **Protocolos criptográficos avanzados**
 - Protocolos de división y compartición de secretos
 - Protocolos de compromiso de bit (bit-commitment)
 - Protocolos de lanzamiento de moneda
 - Protocolo de póker mental
- **Referencias bibliográficas**

Gestión de las Claves

Protocolos de distribución de claves simétricas

- Hay escenarios donde la utilización de la criptografía de clave pública para el intercambio de una clave de sesión K_{AB} no es posible, o simplemente no es conveniente
 - pero a pesar de ello, *Alice* y *Bob* van a seguir necesitando de alguna solución que les permitan, aún estando geográficamente lejanos, decidir esa clave de sesión K_{AB}
- En estos casos, la solución pasa por algún protocolo de **distribución centralizada de claves** para los usuarios del sistema
 - consiste en hacer uso de una tercera parte confiable (TTP), que en este caso se denomina **Centro de Distribución de Claves** (o **KDC** – *Key Distribution Center*)
- Existen diferentes protocolos que proporcionan una solución para ese escenario:
 - Yahalom, Needam-Schroeder, Otway-Rees, Kerberos, ...

- En el modus operandi general de este tipo de protocolos, cada usuario del sistema comparte, de inicio, una **clave secreta** con el KDC
 - mediante algún proceso de registro o inscripción del usuario ante el KDC



KDC: modelos y protocolos

- El uso de un KDC se basa en el uso de claves jerárquicas, de manera que se requieren al menos dos niveles de claves

Canal cifrado
mediante K_{AT}, K_{BT}

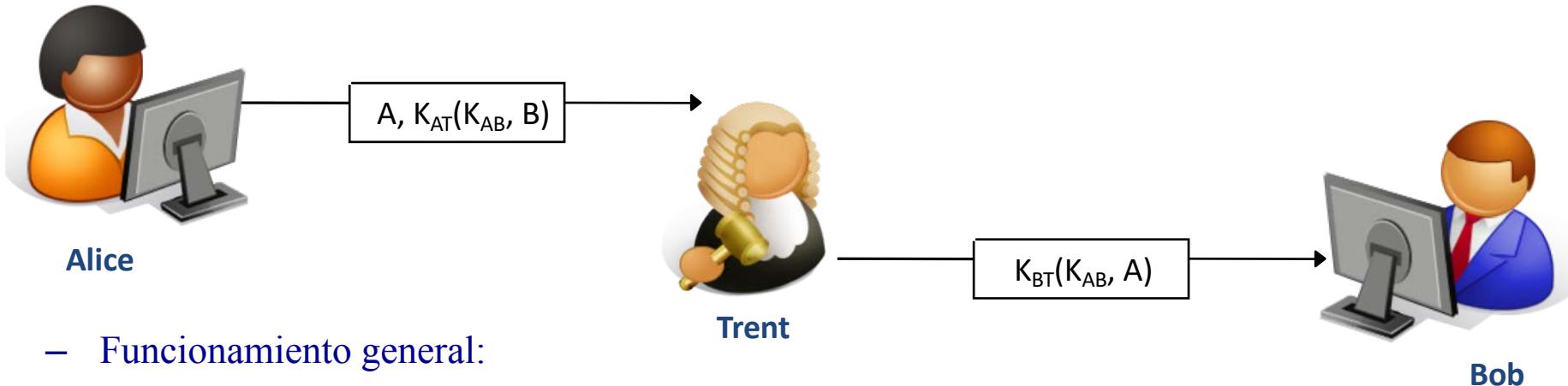


Canal cifrado
mediante K_{AB}

- La mayoría de las técnicas de distribución de claves se adaptan a situaciones, escenarios y aplicaciones específicas, de manera que son diversos los esquemas que se integran a entornos locales donde todos los usuarios tienen acceso a un **servidor común de confianza**
- Hay muchos modelos de distribución de claves:
 - **Simples**
 - **Genéricos**, y dentro de los genéricos, nos podemos encontrar:
 - Los modelos PULL o modelos PUSH, o sus combinaciones

KDC: modelos y protocolos - Modelo Simple

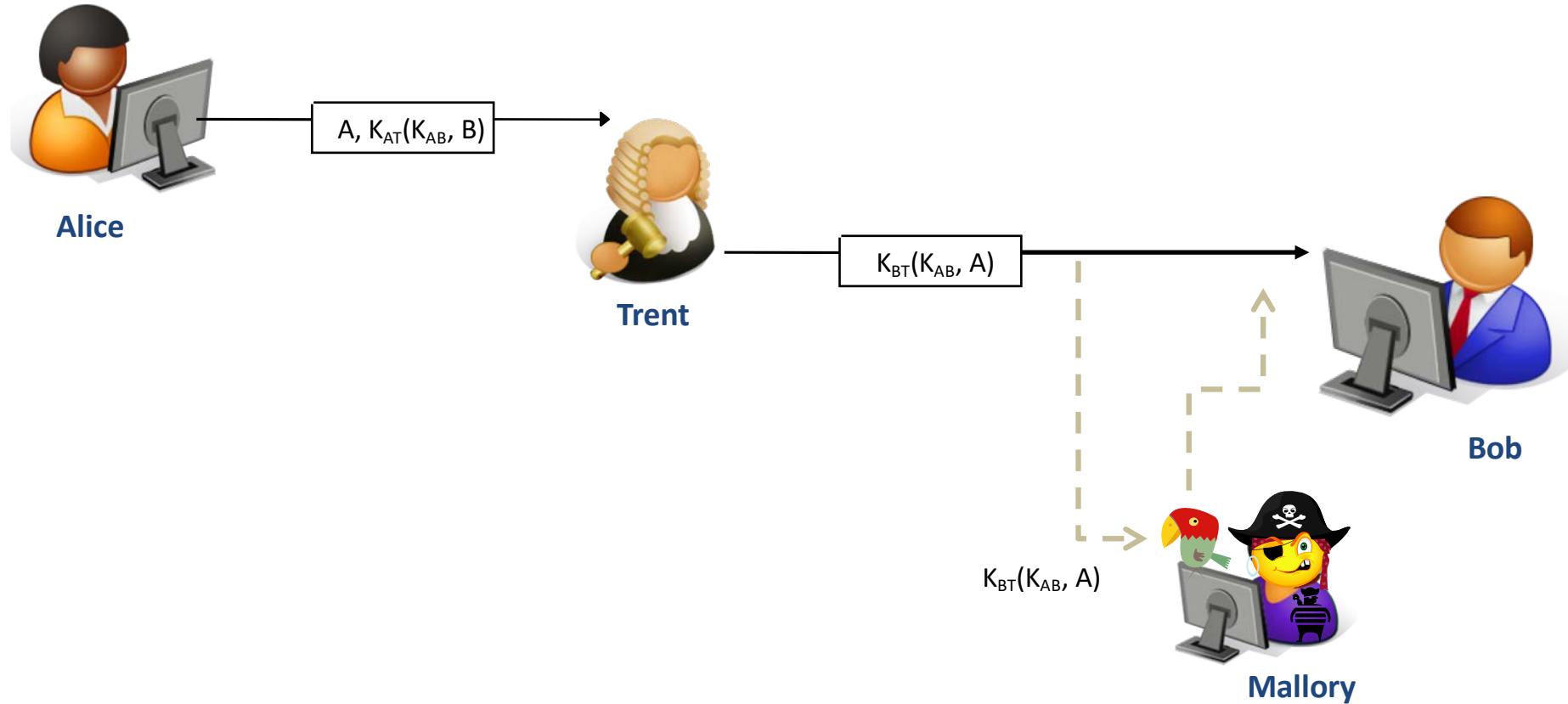
- El protocolo “**La Rana de la Boca Grande**” es un ejemplo de modelo simple para la distribución de claves:



- Funcionamiento general:
 - **Paso 1:** A genera una clave de sesión K_{AB} y se la envía al KDC
 - El mensaje incluye la identidad de A, la identidad de B y la clave de sesión cifrada con el K_{AT}
 - **Paso 2:** El KDC verifica la identidad de A y reenvía la K_{AB} a B cifrado con K_{BT}
 - **Paso 3:** B verifica la identidad de KDC por la K_{BT} y obtiene la clave de sesión
- Como se puede observar existe **validación de identidad**:
 - Las claves con el KDC son secretas, por lo que nadie más habría sido capaz de cifrar la clave secreta K_{AB} , además existe autenticación de cada parte involucrada

KDC: modelos y protocolos - Modelo Simple

- Sin embargo, existe un **fallo de seguridad**:



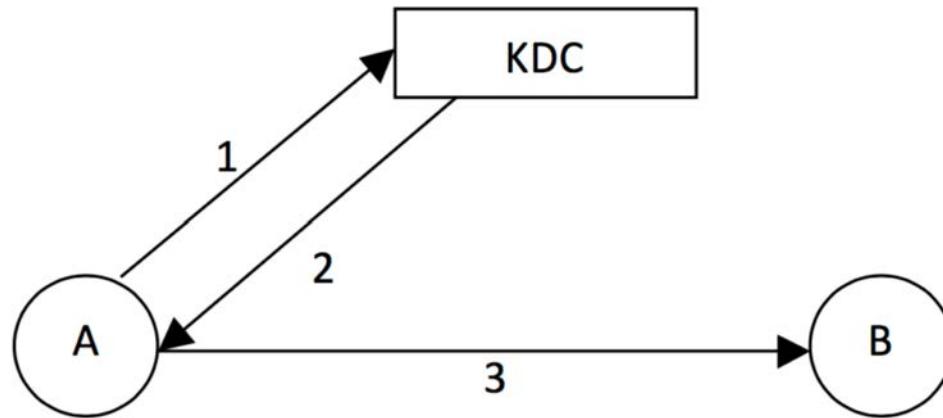
Si Mallory intercepta el canal y captura todos los mensajes de KDC a B, entonces es posible que Mallory cause un ataque de repetición (ataque replay), y, por consiguiente un ataque de Denegación de Servicio (DoS) sin necesidad de que éste derive K_{AB} o K_{BT}

KDC: modelos y protocolos - Modelo Simple

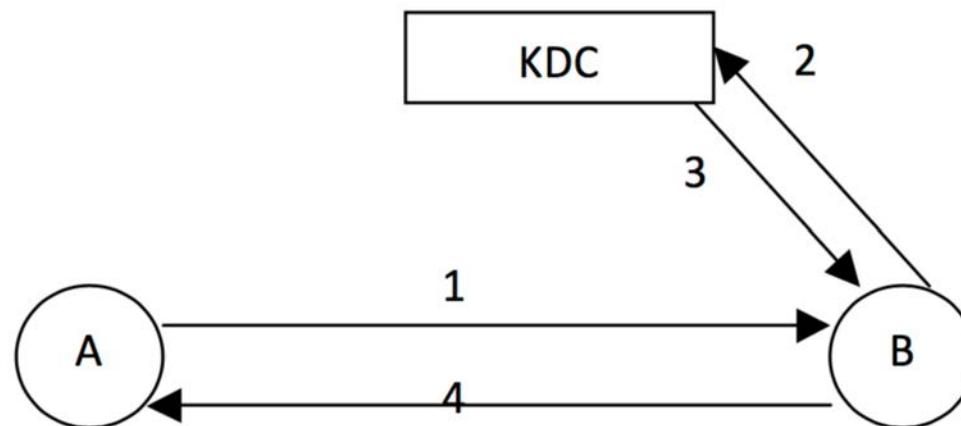
- Para resolver el problema anterior, se pueden hacer uso de alguno de los mecanismos existentes:
 - **Marca de tiempo:** incluir en cada mensaje una marca de tiempo (un sello de tiempo) de forma que pueda descartar mensajes obsoletos
 - Problema: los relojes nunca están perfectamente sincronizados en toda una red
 - **Nonce / único:** incluir un número aleatorio único para cada mensaje enviado, de forma que cada parte de la comunicación debe siempre recordar todos los únicos enviados o recibidos, y rechazar cualquier mensaje que contenga un único previamente usado
 - Problema: si una de las partes pierde la lista de nonce / únicos, es susceptible a ataques replays
 - **Combinación de ambas** estrategias para limitar el tiempo que pueden recordarse los únicos, pero el protocolo se volverá más complicado

KDC: modelos y protocolos - Modelos Genéricos

- Modelo **PULL** para la distribución de claves:

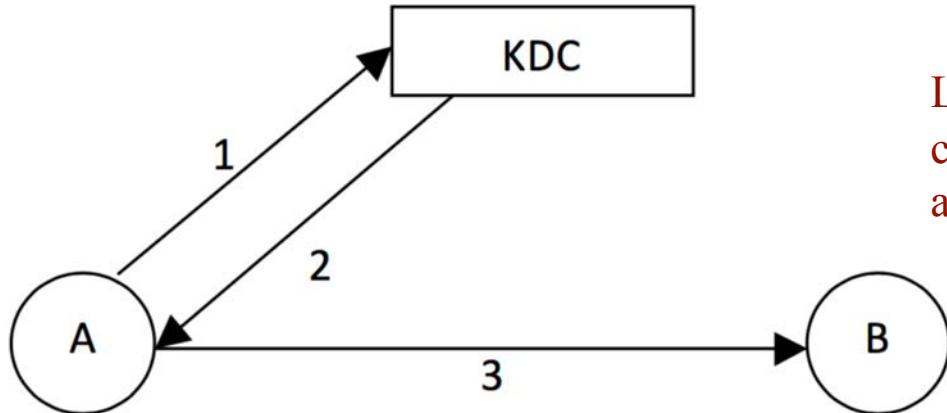


- Modelo **PUSH** para la distribución de claves:



KDC: modelos y protocolos

- Modelo **PULL** para la distribución de claves:



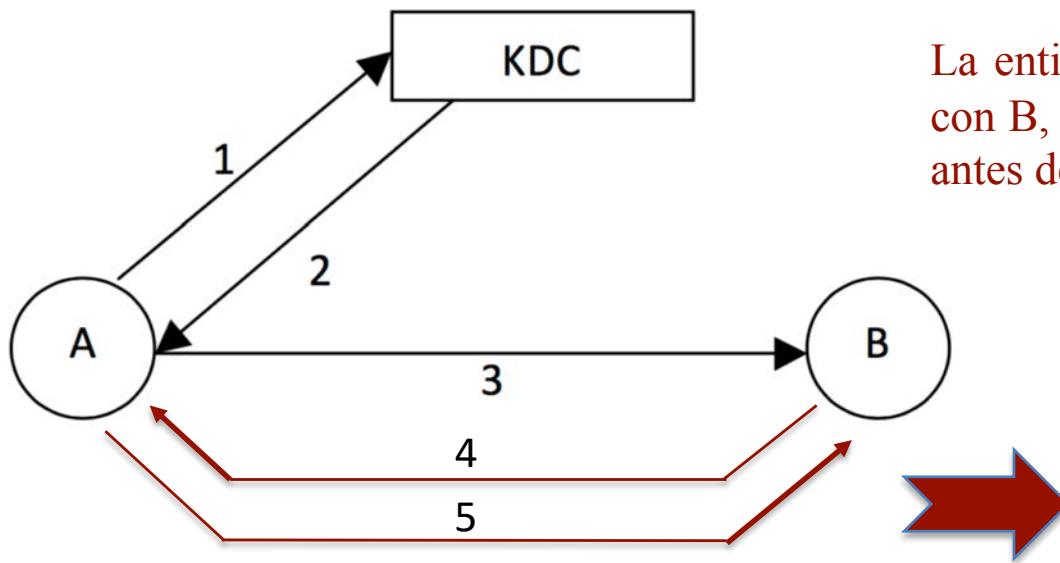
La entidad A desea tener comunicación segura con B, por lo que contacta con el KDC primero antes de comunicarse con B

- Funcionamiento general:

- **Paso 1:** A solicita una clave de sesión K_{AB} al KDC
 - El mensaje incluye la identidad de A, la identidad de B y un valor **N1 (sello de tiempo, valor aleatorio)**
- **Paso 2:** El KDC le contesta a A con un mensaje cifrado mediante la clave maestra K_{AT} , de manera que solamente A puede leer dicho mensaje y con ello, sabe, además, que el KDC es el único que pudo haberlo generado
 - El mensaje contiene la clave K_{AB} , N1, y un mensaje cifrado para B con el K_{AB}
- **Paso 3:** A obtiene la información recibida y reenvía el mensaje a B para que pueda obtener el K_{AB} también

KDC: modelos y protocolos

- Modelo **PULL** para la distribución de claves:



La entidad A desea tener comunicación segura con B, por lo que contacta con el KDC primero antes de comunicarse con B

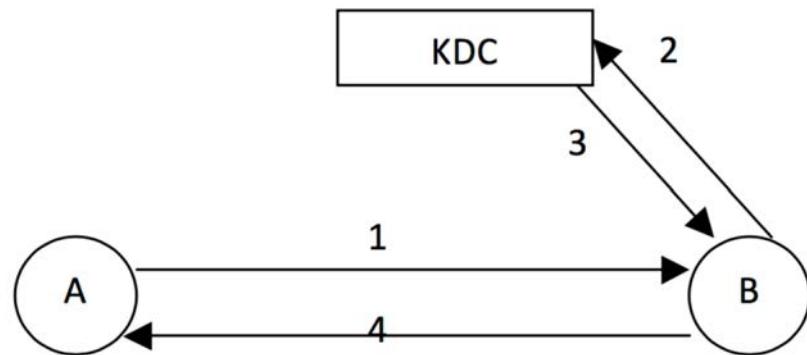
desafío-respuesta
“challenge-response”

- Funcionamiento general:

- **Paso 4:** B utiliza la K_{AB} para cifrar un valor único N_2 y se lo envía a A
- **Paso 5:** A recibe el valor N_2 , le aplica una transformación $f(N_2)$, lo cifra con K_{AB} y lo transmite a B

KDC: modelos y protocolos

- Modelo **PUSH** para la distribución de claves:

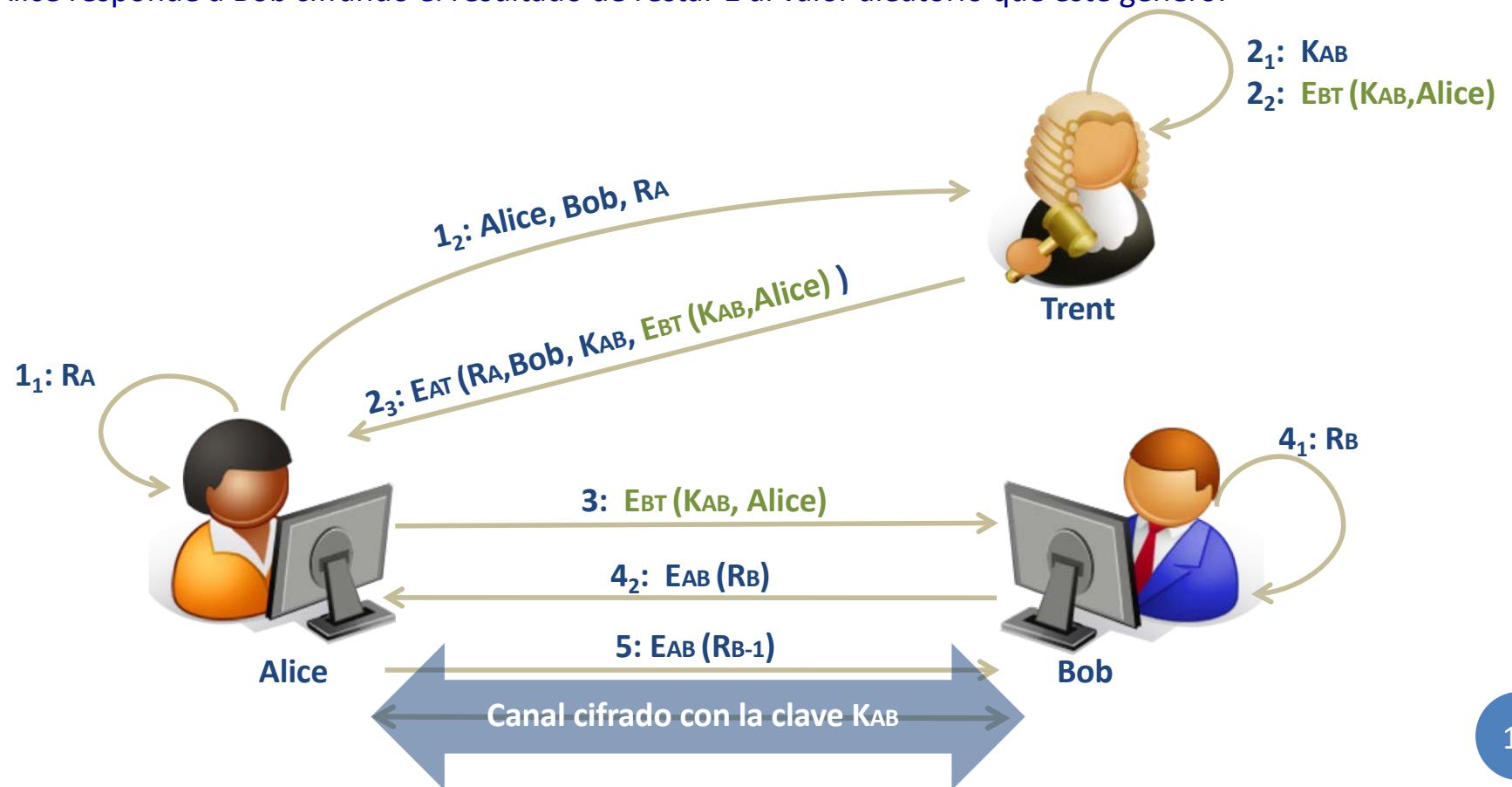


La entidad A contacta primero con la entidad B a fin de que éste última sea la encargada de solicitar al KDC la clave correspondiente

- Funcionamiento general:
 - **Paso 1:** A solicita conexión segura con B, a B.
 - Le manda, como mínimo, su identidad, la identidad de B y un nonce
 - **Paso 2:** B reenvía dicha solicitud a KDC para que éste genere la K_{AB}
 - **Paso 3:** KDC verifica las identidades y el freshnesses de los mensajes, genera la K_{AB} , y dicha información se reenvía a B cifrada con la correspondiente K_{xT}
 - **Paso 4:** B reenvía dicha solicitud a A para que obtenga K_{AB}
 - **Paso 5 (opcional):** A establece un desafío y respuesta

• Protocolo de Needham-Schroeder

- 1: Alice genera el valor aleatorio $R_A <1_1>$, y se lo envía a Trent $<1_2>$.
- 2: Trent genera la clave de sesión $K_{AB} <2_1>$, y se la envía a Alice, junto a un mensaje para Bob $<2_3>$.
- 3: Alice envía a Bob el mensaje que ella ha recibido de Trent.
- 4: Bob genera valor aleatorio R_B y se lo envía a Alice usando la clave K_{AB} (proceso de “challenge-response”).
- 5: Alice responde a Bob cifrando el resultado de restar 1 al valor aleatorio que éste generó.



Needham-Schroeder

- Diseño formalizado:

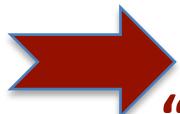
1. $A \rightarrow S : A, B, N_A$  **freshness**

2. $S \rightarrow A : \{N_A, B, K_{A,B}, \{K_{A,B}, A\}_{K_{B,S}}\}_{K_{A,S}}$

3. $A \rightarrow B : \{K_{A,B}, A\}_{K_{B,S}}$

4. $B \rightarrow A : \{N_B\}_{K_{A,B}}$

5. $A \rightarrow B : \{N_B - 1\}_{K_{A,B}}$

 **desafío-respuesta**
“challenge-response”

N_A : nonce/valor aleatorio

S: KDC

A: Alice

B: Bob

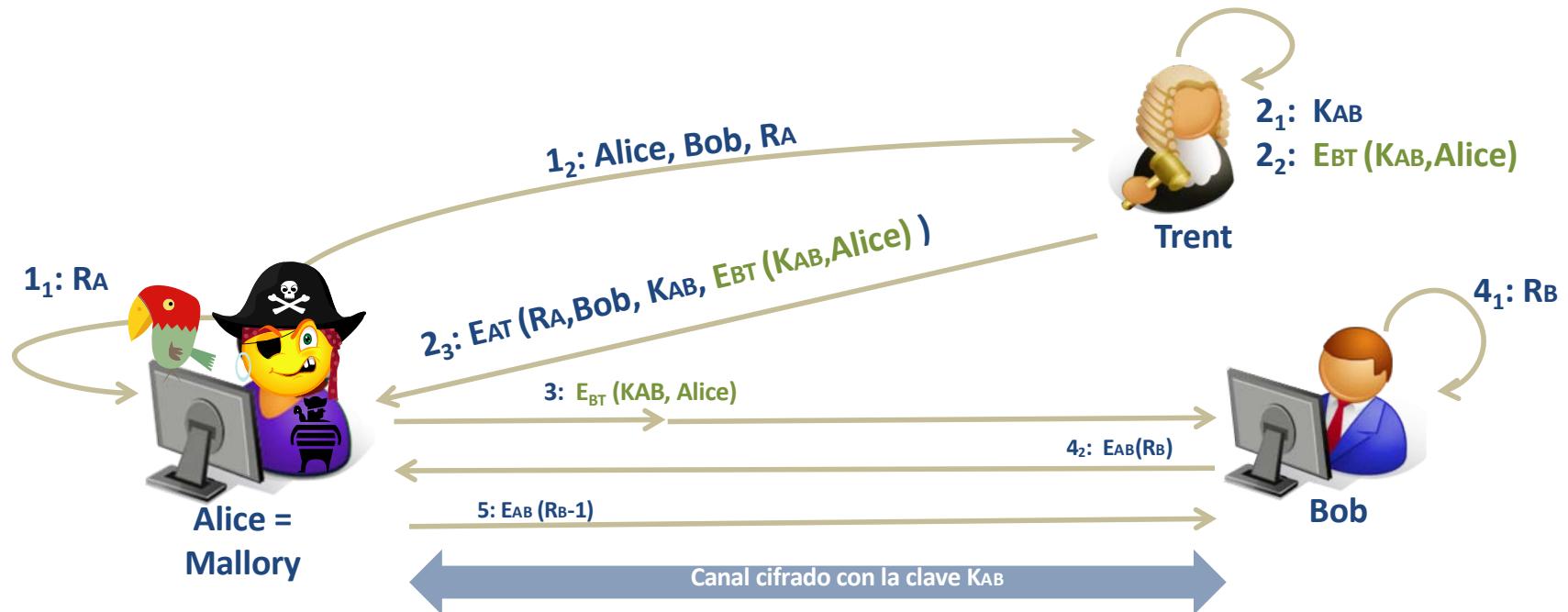
$K_{A,B}$: clave secreta compartida

- Este protocolo tiene varios **fallos de seguridad**, que fueron descubiertos años después de su funcionamiento



- Este protocolo tiene varios **fallos de seguridad**, que fueron descubiertos años después de su funcionamiento
 - **Ataque 1:** Mallory, el atacante, puede suplantar la identidad de Alice si éste consigue derivar la clave K_{AT}
 - **Ataque 2:** Mallory puede suplantar la identidad de Bob si éste consigue derivar la clave K_{BT}
 - **Ataque 3:** Mallory puede producir un ataque de DoS debido a un **ataque de repetición**, especialmente en las últimas fases del protocolo

- **Ataque 1**

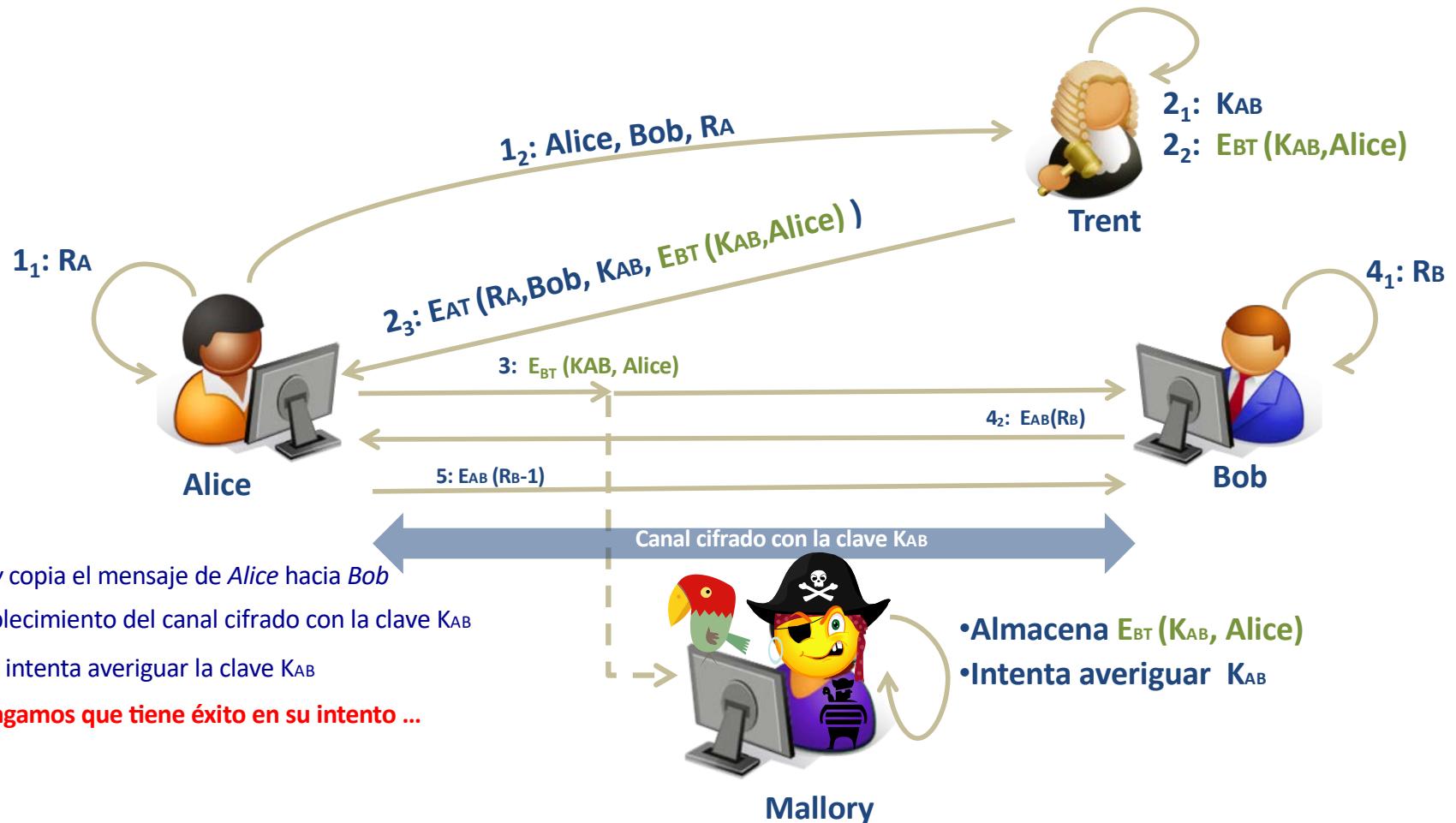


0: *Mallory copia cualquier mensaje de Alice hacia Trent en el pasado y deriva la clave K_{AT}. A partir de aquí, todos los mensajes quedan comprometidos*

supongamos que tiene éxito en su intento ...

- **Almacena E_{BT}(K_{AB}, Alice)**
- **Intenta averiguar K_{AB}**

- **Ataque 2**



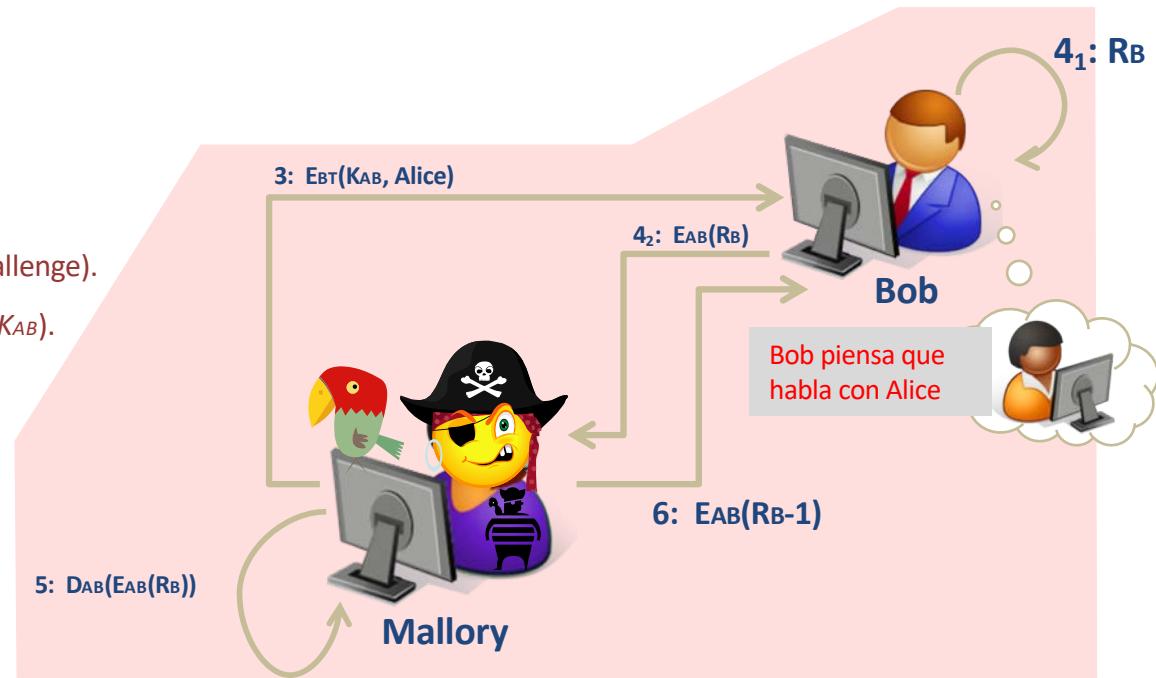
- continuación...

3: *Mallory* envía el mensaje $E_{BT}(K_{AB}, Alice)$ a *Bob*.

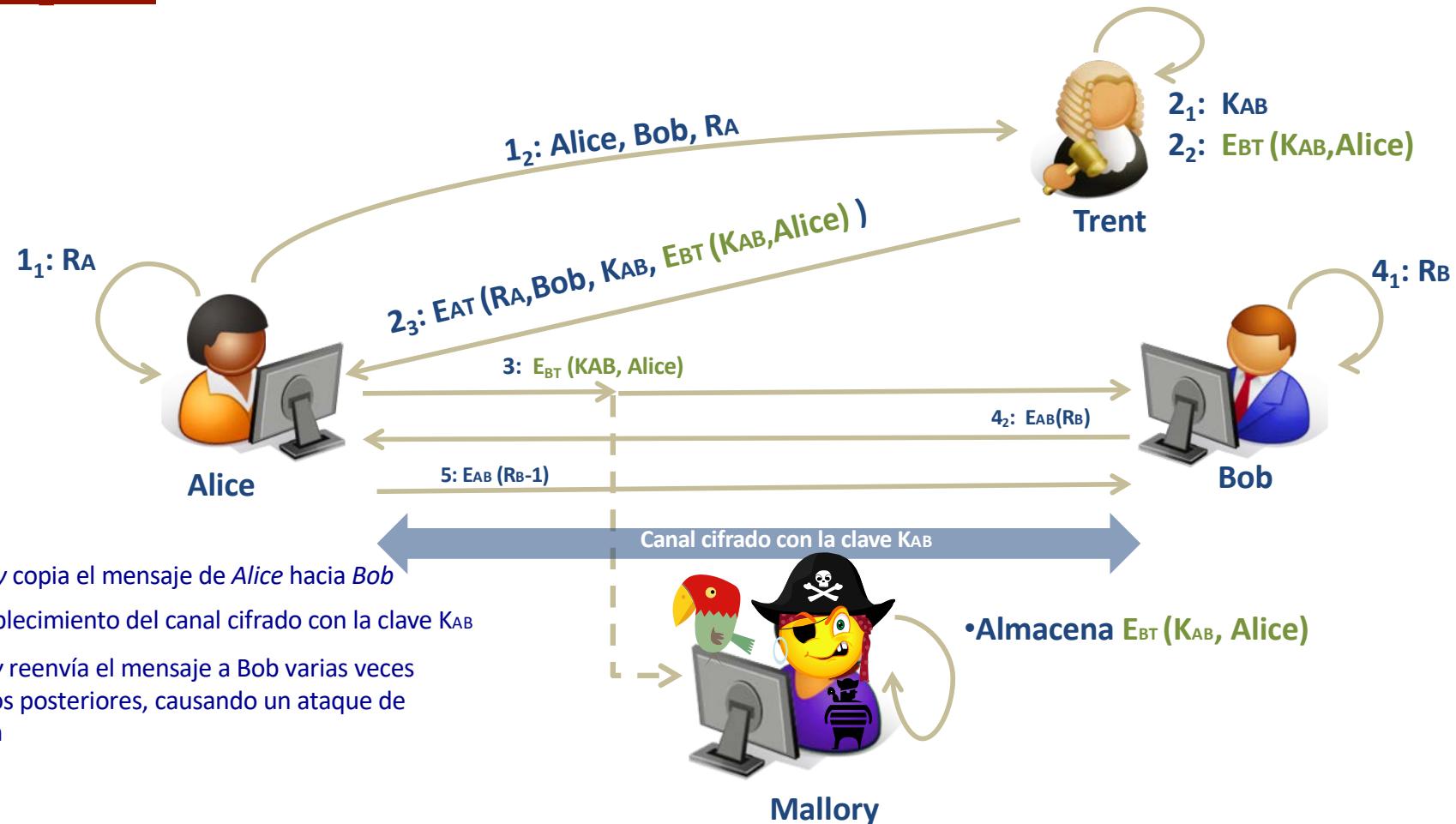
4: *Bob* responde a *Mallory* con un valor aleatorio (challenge).

5: *Mallory* descifra el valor aleatorio (porque conoce K_{AB}).

6: *Mallory* responde al challenge de *Bob*, y *Bob* piensa que habla con *Alice*.



- **Ataque 3**



- Metiendo en el mensaje 3 un nonce:

1. $A \rightarrow S : A, B, N_A$
2. $S \rightarrow A : \{N_A, B, K_{A,B}, \{K_{A,B}, A\}_{K_{B,S}}\}_{K_{A,S}}$

3. $A \rightarrow B : \{K_{A,B}, A\}_{K_{B,S}}$
4. $B \rightarrow A : \{N_B\}_{K_{A,B}}$
5. $A \rightarrow B : \{N_B - 1\}_{K_{A,B}}$

- Solución:

1. $A \rightarrow S : A, B, N_A$
2. $S \rightarrow A : \{N_A, B, K_{A,B}, \{K_{A,B}, A\}_{K_{B,S}}\}_{K_{A,S}}$
3. $A \rightarrow B : \{K_{A,B}, A\}_{K_{B,S}}$
4. $B \rightarrow A : \{N_B\}_{K_{A,B}}$
5. $A \rightarrow B : \{N_B - 1\}_{K_{A,B}}$

Problema: el freshness solo se encuentra en los mensajes 1 y 2, pero no en el resto de mensajes

Solución: extender el uso del nonce en el resto de transacciones

- Protocolo Amended Needham Schroeder protocol
 - soluciona el fallo del anterior Needham-Schroeder (en relación a los ataques de repetición)

$A \rightarrow B : A$

$B \rightarrow A : E_{KBT}\{A, N_{b0}\}$

$A \rightarrow T : A, B, N_a, E_{KBT}\{A, N_{b0}\}$

$T \rightarrow A : E_{KAT}\{A, N_a, K_{AB}, E_{KBT}\{A, K_{AB}, N_{b0}\}\}$

$A \rightarrow B : E_{KBT}\{A, K_{AB}, N_{b0}\}$

$B \rightarrow A : E_{KAB}\{N_b\}$

$A \rightarrow B : E_{KAB}\{N_{b-1}\}$

$3_1 : RA$

$3_2 : Alice, Bob, RA, E_{BT}(A, R_{B0})$

$4_3 : EAT(RA, Bob, K_{AB}, E_{BT}(K_{AB}, Alice, R_{B0}))$

$1 : A$

$2_2 : E_{BT}(A, R_{B0})$

$5 : E_{BT}(K_{AB}, Alice, R_{B0})$

$6_2 : E_{AB}(R_B)$

$5 : E_{AB}(R_{B-1})$

Alice

$4_1 : K_{AB}$

$4_2 : E_{BT}(K_{AB}, Alice, R_{B0})$



Trent

$2_1 : R_{B0}$

$6_1 : R_B$



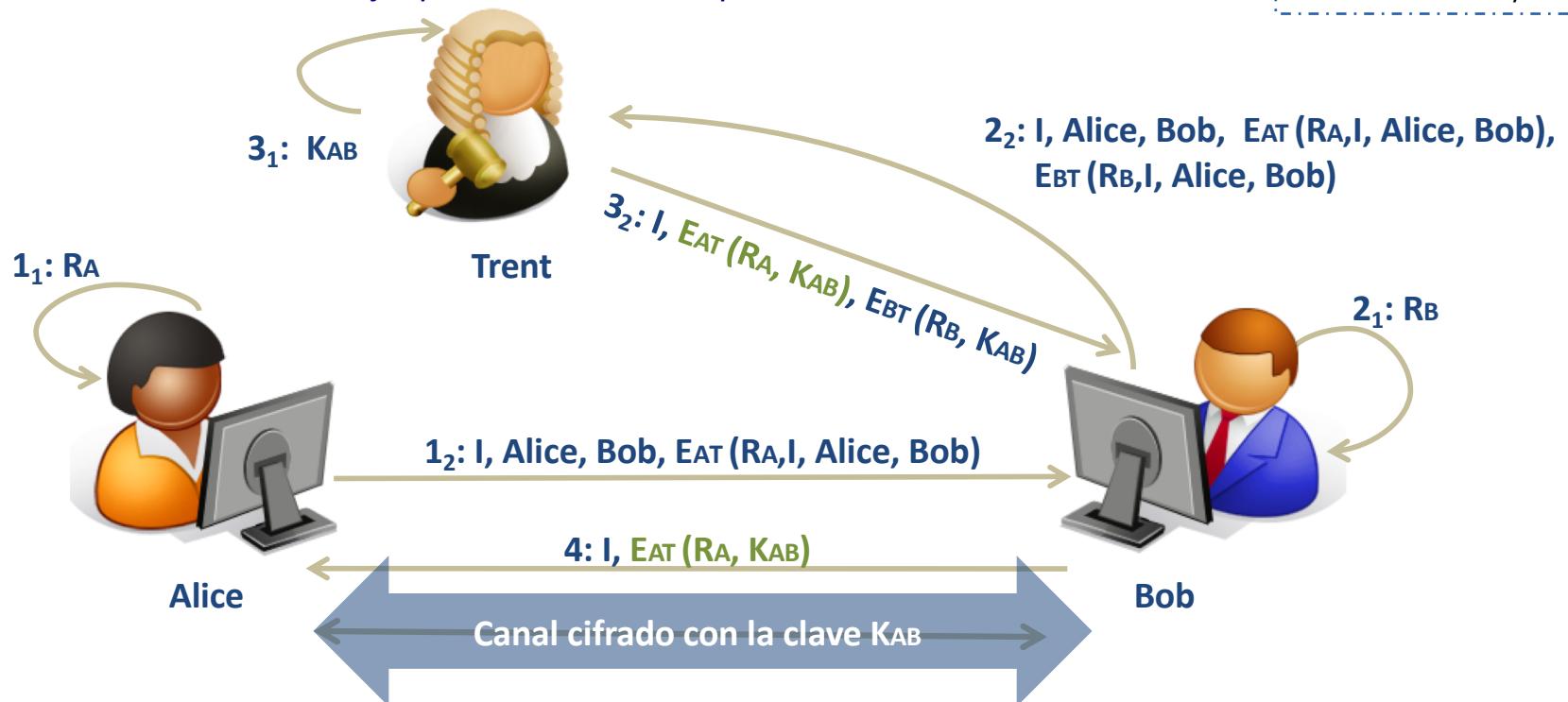
Bob



- **Protocolo Otway-Rees**
 - soluciona también el fallo del **Needham-Schroeder**, aunque con un diseño diferente

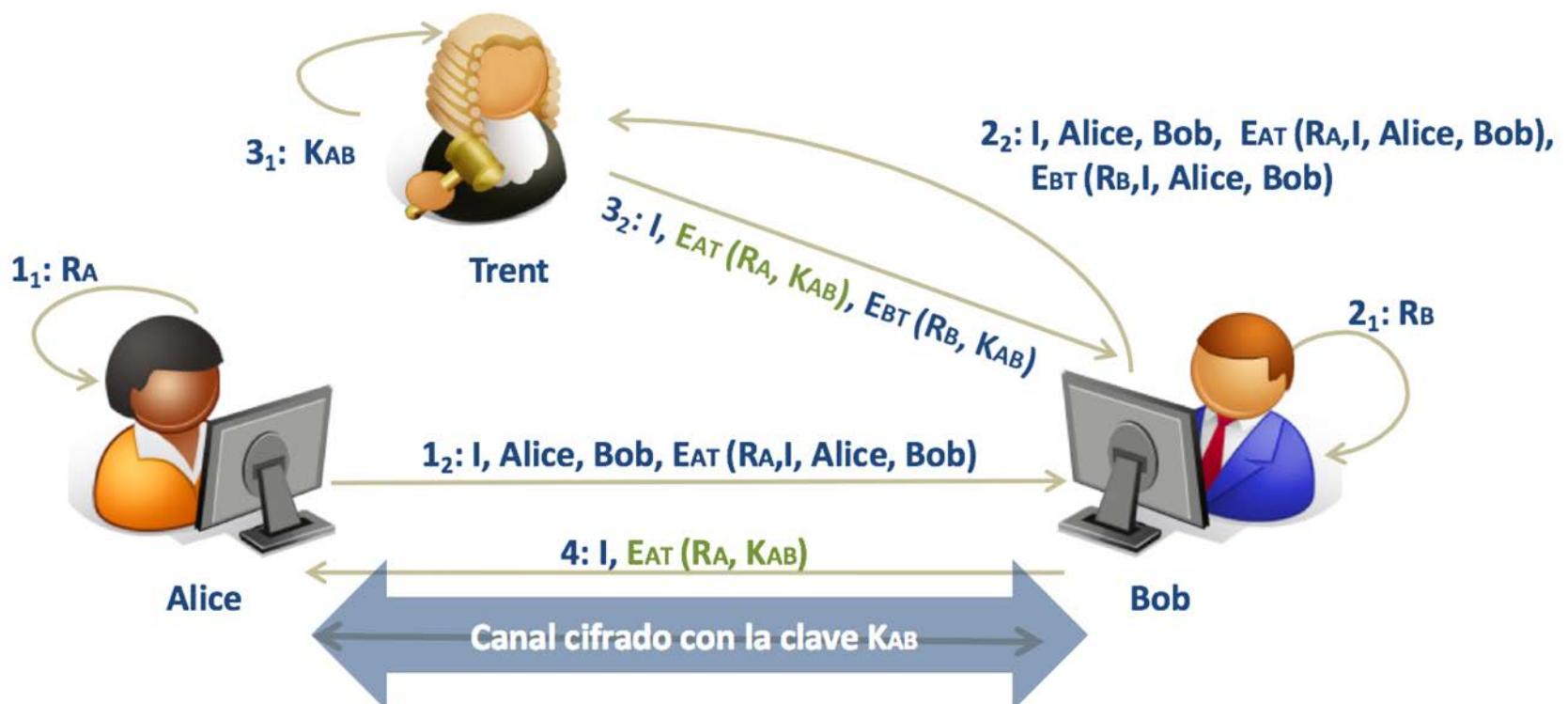
- 1:** Alice genera el valor aleatorio $R_A <1_1>$ y se lo envía hacia Bob, dentro de un mensaje cifrado con la clave que comparte con Trent $<1_2>$.
- 2:** Bob genera un valor aleatorio R_B y se lo envía a Trent usando la clave que comparte con éste $<2_2>$. También le envía el mensaje que recibió de Alice.
- 3:** Trent descifra el mensaje cifrado con la clave que comparte con Alice, genera la clave de sesión $K_{AB} <3_1>$ y se la envía a Bob cifrada, junto a un mensaje para Alice $<3_2>$.
- 4:** Bob envía a Alice el mensaje que recibió de Trent para ella.

I (Índice): I-ésima sesión establecida entre A y B.



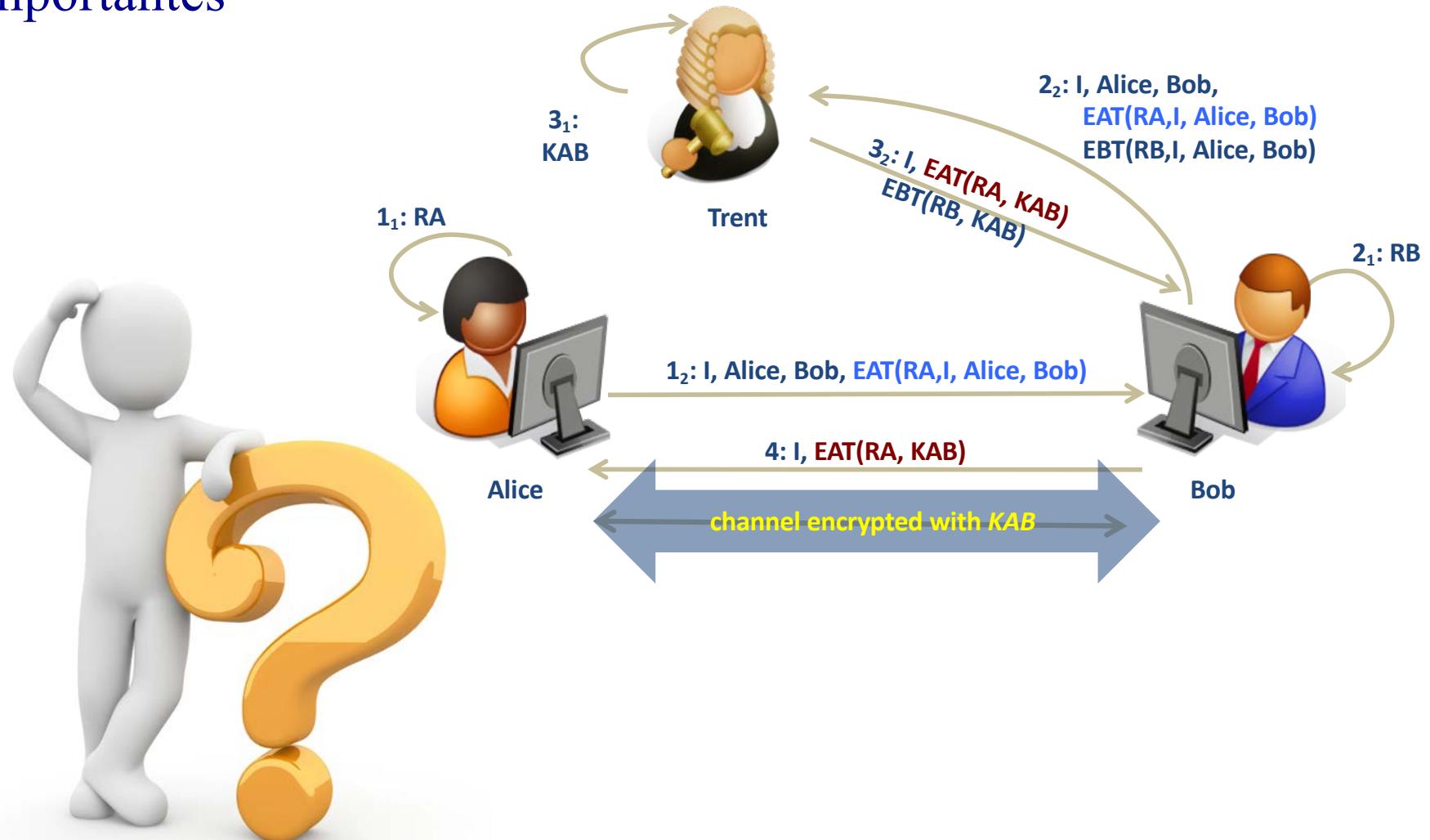
Otway-Rees

- Diseño formalizado:

$$A \rightarrow B : I, A, B, E_{KAT}\{N_a, I, A, B\}$$
$$B \rightarrow T : I, A, B, E_{KAT}\{N_a, I, A, B\}, E_{KBT}\{N_b, I, A, B\}$$
$$T \rightarrow B : I, E_{KAT}\{K_{AB}, N_a\}, E_{KBT}\{K_{AB}, N_b\}$$
$$B \rightarrow A : I, E_{KAT}\{K_{AB}, N_a\}$$


Otway-Rees

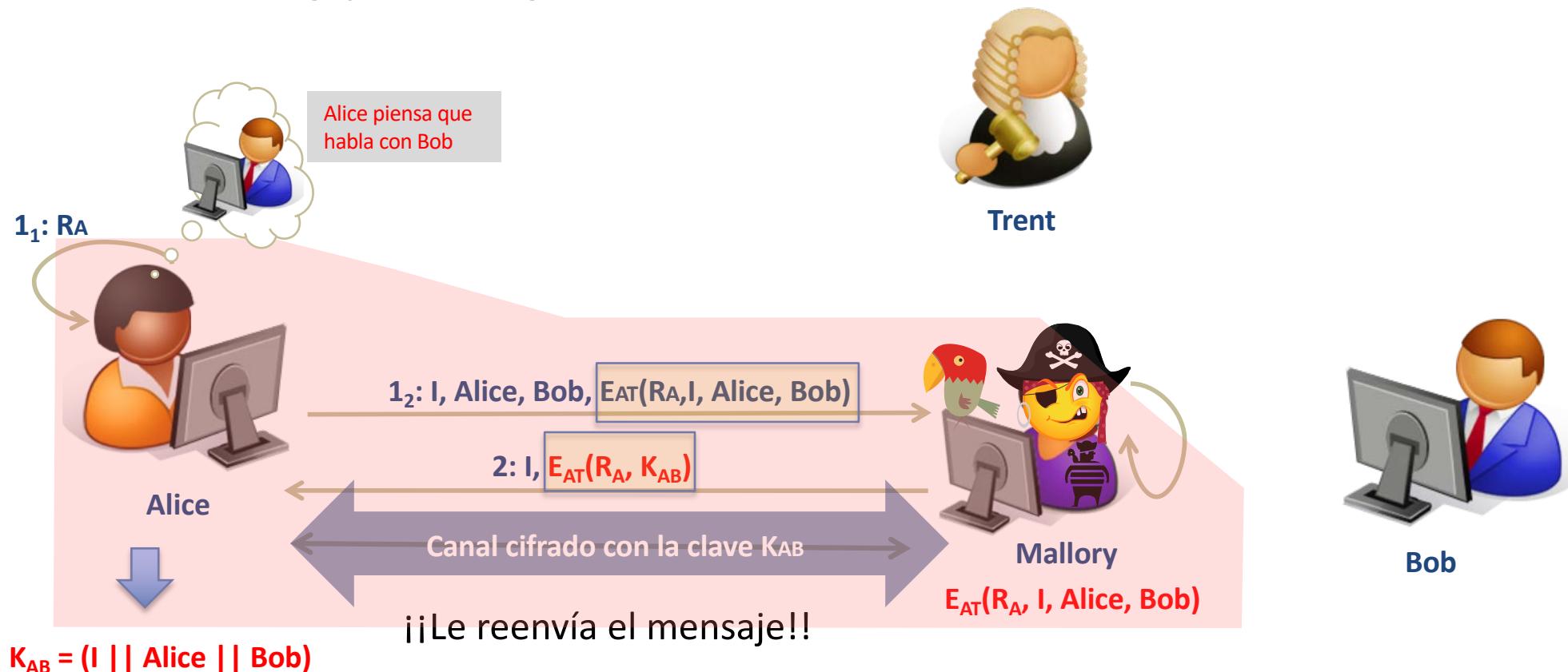
- Sin embargo, el protocolo presenta dos vulnerabilidades importantes



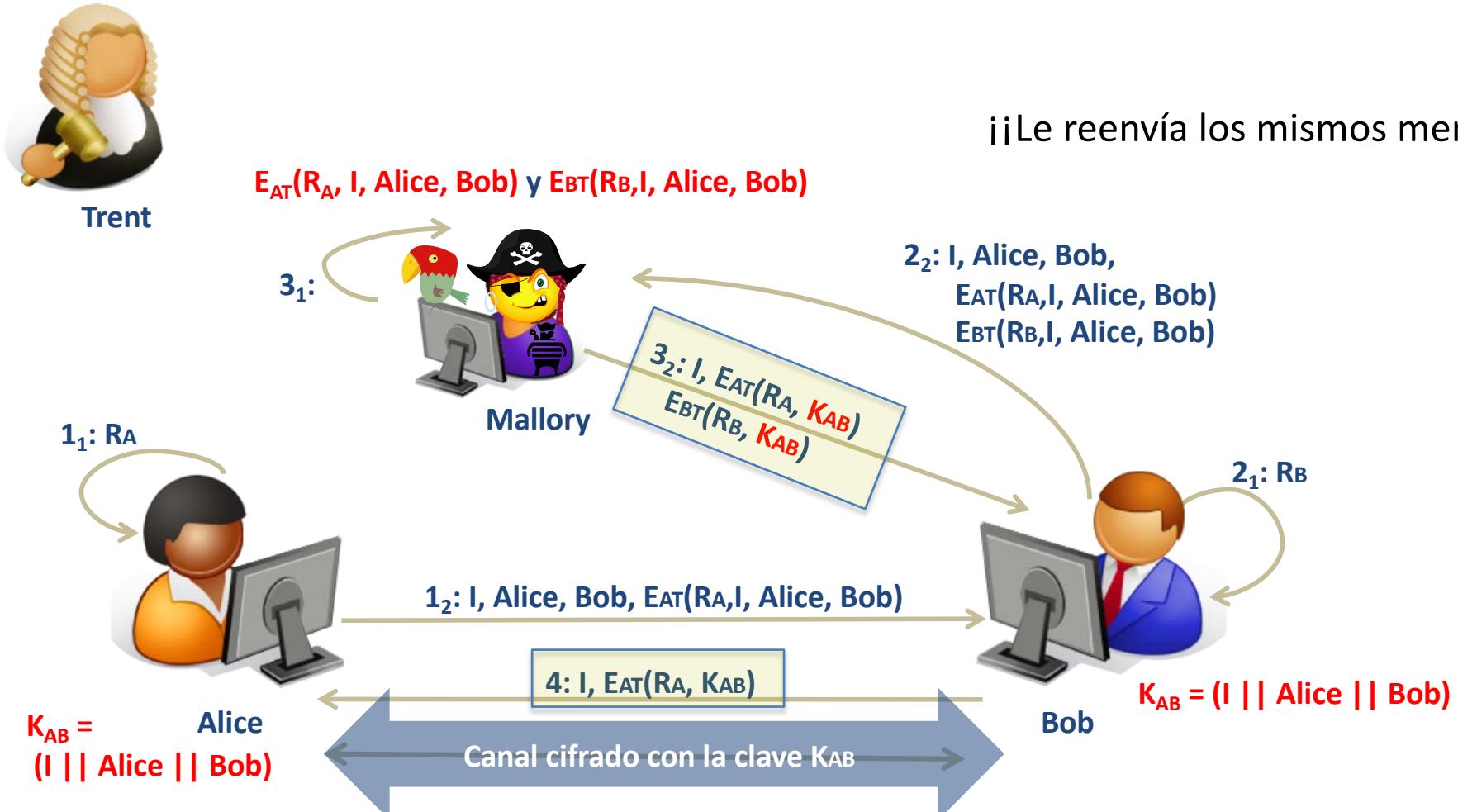
- Este protocolo también tiene un **fallo de seguridad**, y concretamente dos:
 - Primer agujero de seguridad:



- Otway-Rees también tiene **fallos de seguridad**, y concretamente dos:
 - Primer agujero de seguridad:



– Segundo agujero de seguridad:

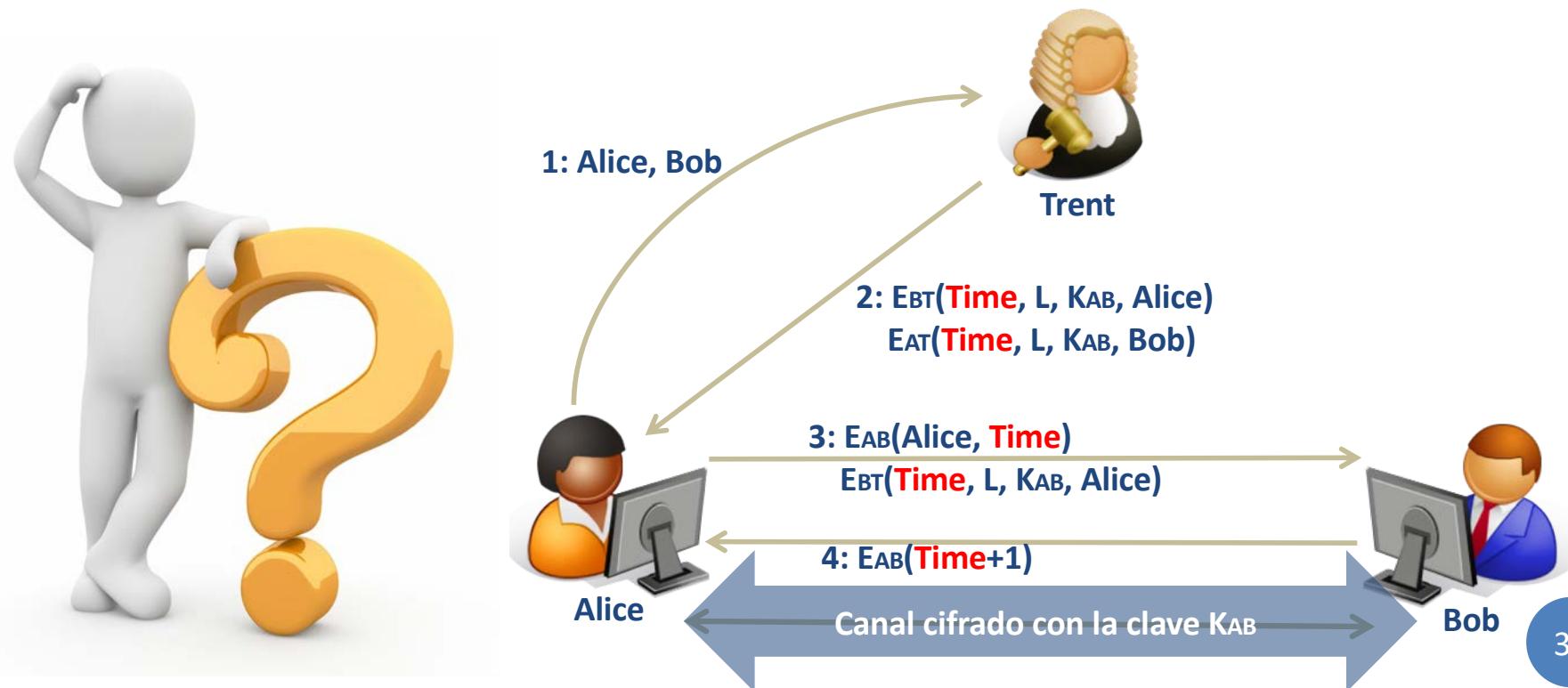


– Segundo agujero de seguridad:



• Protocolo Kerberos

- 1: Alice envía un mensaje a Trent con su identidad y la identidad de Bob.
- 2: Trent genera un mensaje con un *timestamp* (*Time*), un *tiempo de vida* (*L*), una clave de sesión aleatoria, y la identidad de Alice. Lo cifra con la clave compartida con Bob. Prepara un mensaje similar para Alice. Envía ambos mensajes cifrados a Alice.
- 3: Alice obtiene K_{AB} , genera un mensaje con su identidad y el timestamp, y lo cifra con K_{AB} para enviárselo a Bob. Alice también envía a Bob el mensaje cifrado que recibió de Trent.
- 4: Bob genera un mensaje que consta del timestamp más uno, lo cifra con K_{AB} y se lo envía a Alice.



• Protocolo Kerberos

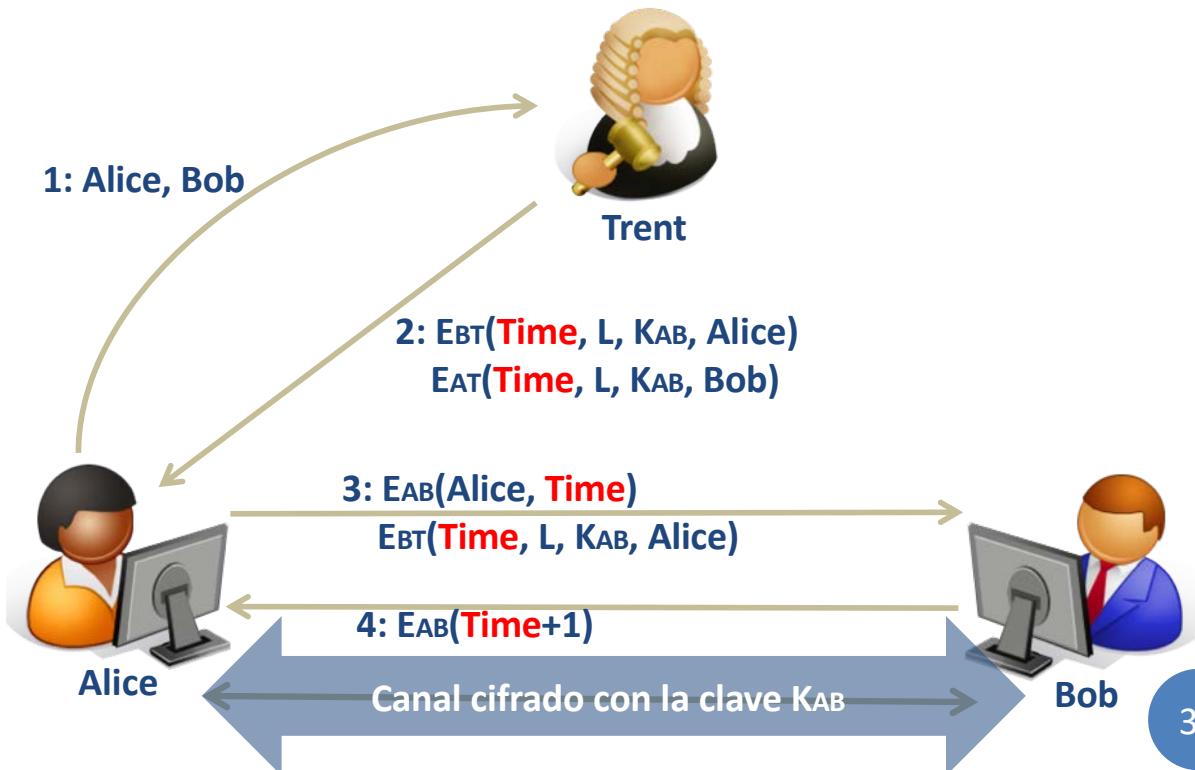
- 1: Alice envía un mensaje a Trent con su identidad y la identidad de Bob.
- 2: Trent genera un mensaje con un *timestamp* (*Time*), un *tiempo de vida* (*L*), una clave de sesión aleatoria, y la identidad de Alice. Lo cifra con la clave compartida con Bob. Prepara un mensaje similar para Alice. Envía ambos mensajes cifrados a Alice.
- 3: Alice obtiene K_{AB} , genera un mensaje con su identidad y el timestamp, y lo cifra con K_{AB} para enviárselo a Bob. Alice también envía a Bob el mensaje cifrado que recibió de Trent.
- 4: Bob genera un mensaje que consta del timestamp más uno, lo cifra con K_{AB} y se lo envía a Alice.

- Asume que los relojes de todos los sistemas están sincronizados con el reloj de Trent.

En la práctica se sincronizan en el rango de unos pocos minutos.

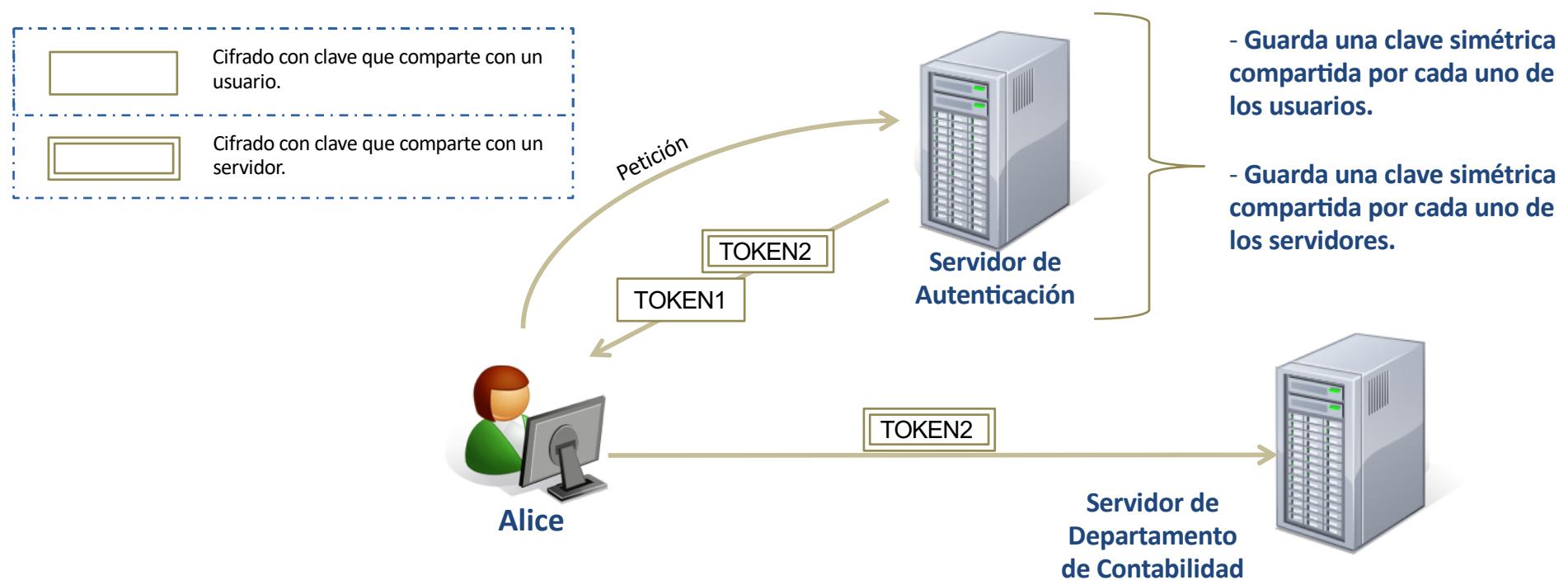
- Por fallos del sistema o por sabotaje, los relojes pueden desincronizarse (→ ataque)

¿Dónde podría usarse?...

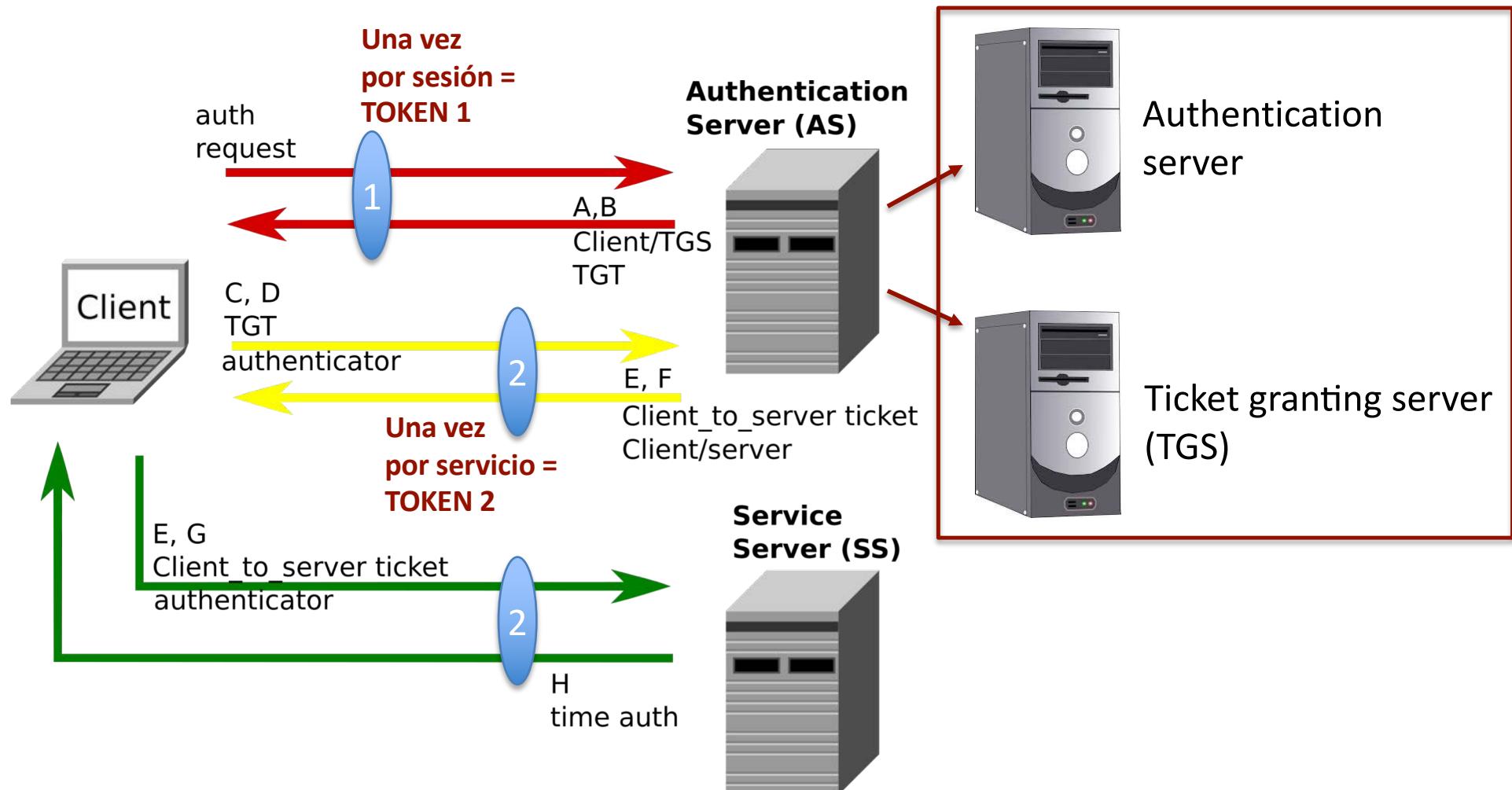


Kerberos

- Ejemplo de escenario de uso de Kerberos
 - Campus universitario, empresa, ...
 - Se usa Kerberos para evitar que cada usuario tenga una cuenta en cada servidor con el que va a contactar
 - En el escenario de abajo *Alice* accede al Servidor del Departamento de Contabilidad sin tener una cuenta en ese servidor



Kerberos



- Aparte de estos protocolos, hay muchos otros que combinan múltiples tipos de estrategias (a nivel de modelos como de mecanismos de seguridad):

- **Yahalom**

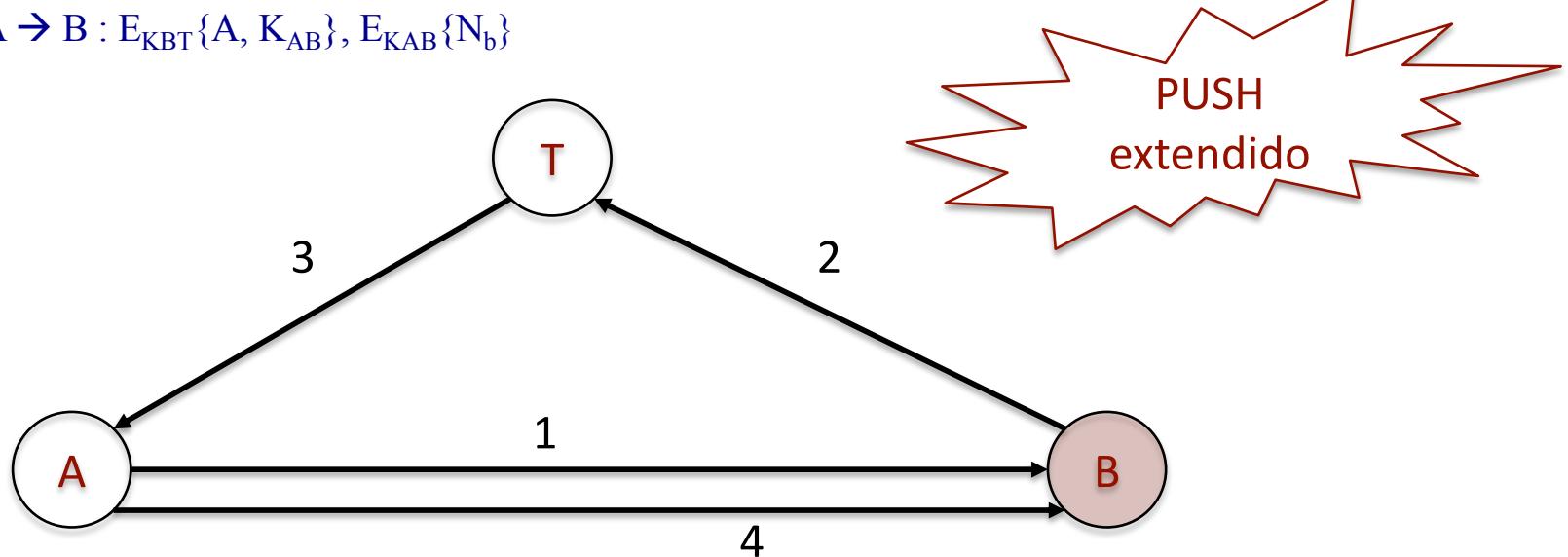
- Objetivo: consiste que Trent genere la clave de sesión K_{AB} y lo envía directamente a Alice e indirectamente a Bob

$A \rightarrow B : A, N_a$

$B \rightarrow T : B, E_{KBT}\{A, N_a, N_b\}$

$T \rightarrow A : E_{KAT}\{B, K_{AB}, N_a, N_b\}, E_{KBT}\{A, K_{AB}\}$

$A \rightarrow B : E_{KBT}\{A, K_{AB}\}, E_{KAB}\{N_b\}$



– Neuman Stubblebine

- Objetivo: consiste en combinar múltiples formas de verificar la autenticidad de las transacciones – time-stamps, nonces

$A \rightarrow B : A, N_a$

$B \rightarrow T : B, E_{KBT}\{A, N_a, \text{time-stamp}_b\}, N_b$

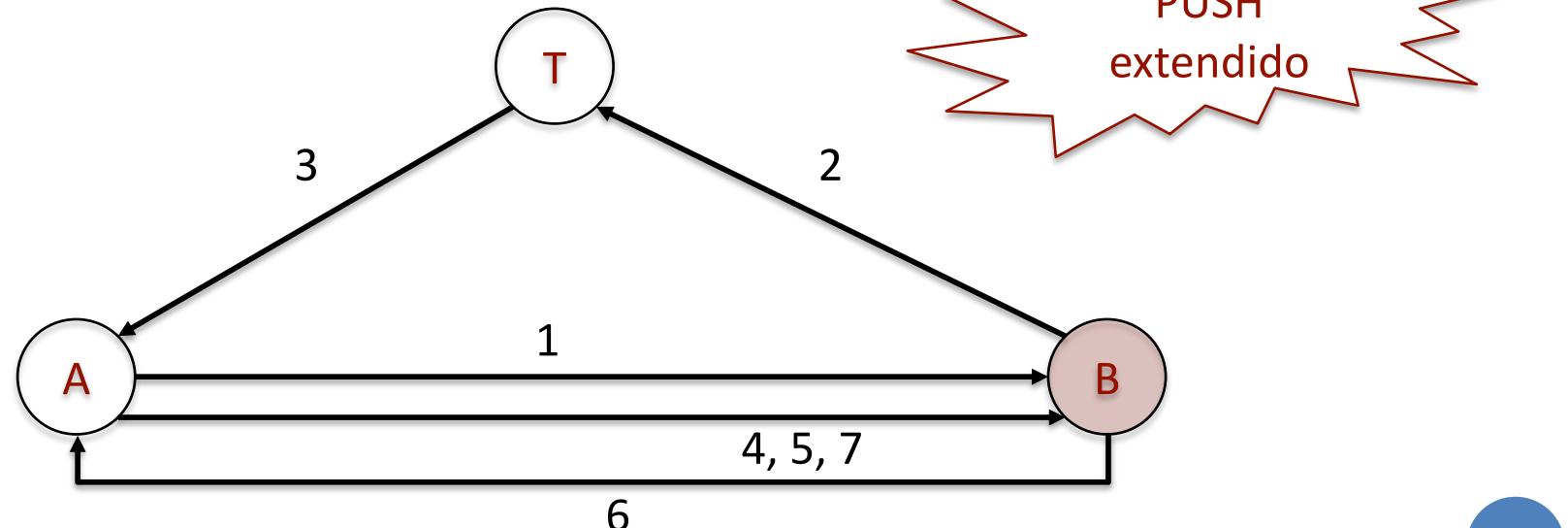
$T \rightarrow A : E_{KAT}\{B, K_{AB}, N_a, \text{time-stamp}_b\}, E_{KBT}\{A, K_{AB}, \text{time-stamp}_b\}, N_b$

$A \rightarrow B : E_{KBT}\{A, K_{AB}, \text{time-stamp}_b\}, E_{KAB}\{N_b\}$

$A \rightarrow B : N'_a, E_{KBT}\{A, K_{AB}, \text{time-stamp}_b\}$

$B \rightarrow A : N'_b, E_{KAB}\{N'_a\}$

$A \rightarrow B : E_{KAB}\{N'_b\}$



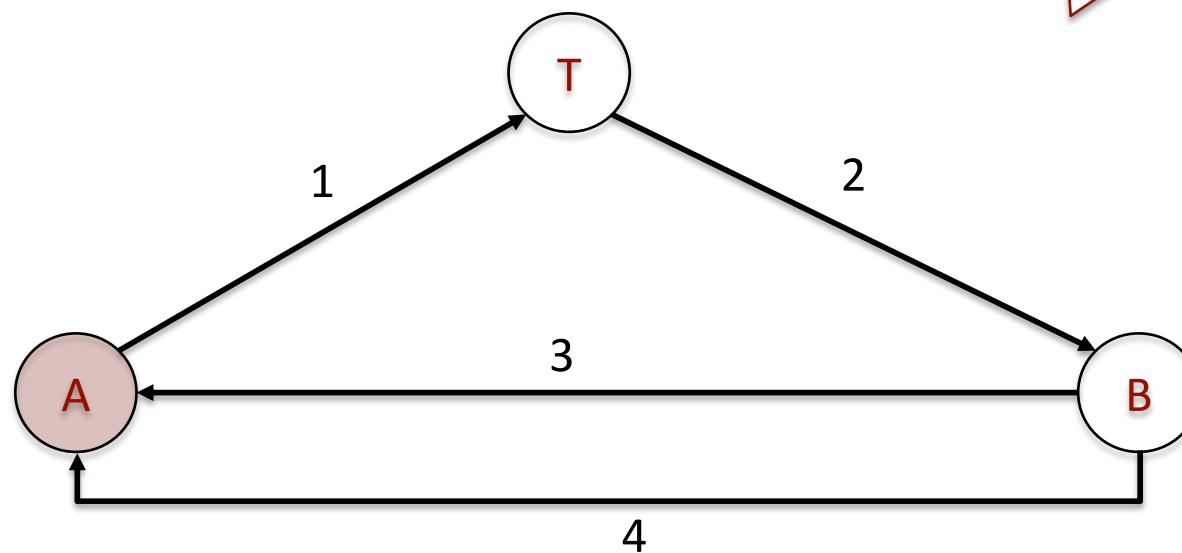
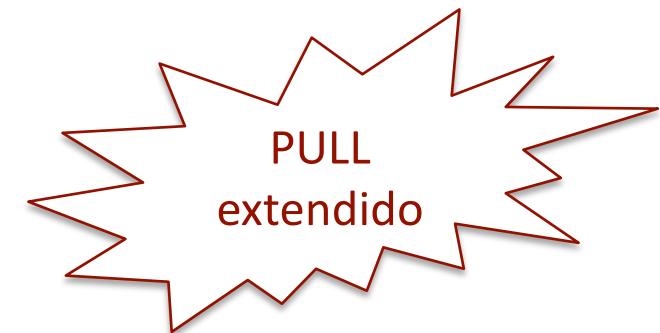
– Kao Chow

$A \rightarrow T : A, B, N_a$

$T \rightarrow B : E_{KAT}\{N_a, K_{AB}, A, B\}, E_{KBT}\{N_a, K_{AB}, A, B\}$

$B \rightarrow A : E_{KAT}\{N_a, K_{AB}, A, B\}, E_{KAB}\{N_a\}, N_b$

$A \rightarrow B : E_{KAB}\{N_b\}$



- Hemos visto que existen distintos protocolos para solucionar el mismo tipo de problema de administración/intercambio de claves
- El protocolo a elegir depende bastante de la **arquitectura de comunicaciones subyacente**, o lo que es lo mismo, de las respuestas a preguntas como:
 - ¿Es necesario minimizar el tamaño de los mensajes?
 - ¿Es necesario minimizar el número de mensajes?
 - ¿Quién ha de contactar primero con el KDC: Alice o Bob?
 - ¿Debe el KDC contactar directamente con ambos o es suficiente que lo haga con sólo uno de ellos?
 - etc.

- Usar un KDC no está exento de potenciales problemas:
 1. el KDC posee suficiente **información para suplantar** a cualquier usuario
 - y si un intruso llega hasta él todos los documentos cifrados que circulan por la red se hacen vulnerables;
 1. el KDC representa un **único punto de fallos** (o ataques)
 - si queda inutilizado nadie puede establecer comunicaciones seguras dentro de la red;
 2. el **rendimiento** de todo el sistema **puede bajar** cuando el KDC se convierte en un cuello de botella
 - lo cual no es difícil ya que todos los usuarios necesitan comunicar con él de forma frecuente con objeto de obtener las claves.

- Caso real de uso por desarrolladores en Information Security Stack Exchange Q&A:

- Question: Today I was reading notes about cryptography and I came across a problem that exists in Symmetric Key encryption: How to share the secret key across the network
 - 1st Method: Make use of the trusted KDC (Key Distribution Center)
 - 2nd Method: Encrypt the key using Public Key technique and exchange the key (e.g., SSL)

My question is:

- which technique is prove to be more effective?
- Why has SSL implemented method 2? Is it because it's more secure?

Which technique is prove to be more effective?

- Answer 1:

KDC is suitable for smaller infrastructures where you place explicit trust into each person or node doing encryption. Each time Alice wants to encrypt a message to Bob, she has to ask KDC for a temporary key to use for encryption. The KDC will also need to provide that temporary key to Bob so he can decrypt the message. You don't want to just give Alice Bob's encryption key, as then she would be able to read all encrypted messages sent to Bob (the nature of symmetric encryption is such that the same key both encrypts and decrypts the message).

As you can imagine, this does not scale beyond local infrastructures, nor is very fault-tolerant. If the Internet relied on KDCs in order to do encryption, it would be easy for attackers to DoS them and kill all commerce on the web -- or at least make it unreliable enough not to bother. This is why browsers went with Public Key cryptography and a framework of mutually trusted certificate authorities (CAs). This has its own set of trade-offs. One, you have to trust that the CAs know what they are doing -- a trust they have repeatedly violated. Two, PKI relies on the hope that we'll never find a fast way to factor the product of two very large prime numbers -- a problem not present in symmetric cryptography. If the likes of Grigory Perelman one day find a way to quickly solve that problem, anyone will be able to obtain the private decryption key from the public encryption key. If that happens, we'll be all in big trouble and will have to come up with some other way for two untrusted parties to exchange encryption keys.

So, to answer your question:

- cryptographically, KDCs and symmetric keys are stronger than asymmetric public-private keys, as there is no danger that one day some crazy mathematician will find a way to quickly factor products of large primes.
- On the other hand, KDCs have inherent problems with key distribution, reliability and ongoing trust that can't be easily solved and therefore KDCs are not suitable beyond local installations where such trust is easy to assure.

Why has SSL implemented method 2? Is it because it's more secure?

- Answer 2:

A key distribution centre is a central system which distributes the keys to the user. Its **central nature** implies that:

- Everybody talks to the KDC, so the KDC is easily overwhelmed in big networks.
- The KDC has the power to betray everybody in the system.

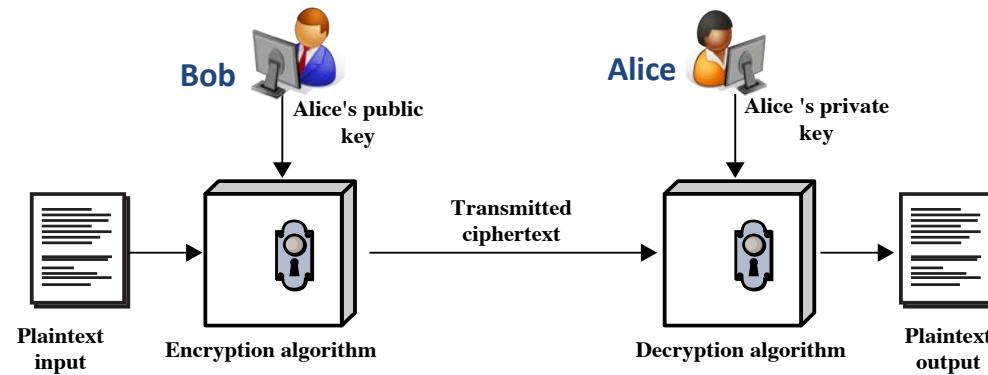
As such, a KDC does not scale well, and requires an existing, **online infrastructure**.

On the other hand, **SSL uses asymmetric cryptography and public key distribution with certificates**; even there, there *are* entities who can betray many people (*the certification authorities*) but at least they can operate offline, which avoids scalability issues and makes them easier to protect against external subversion.

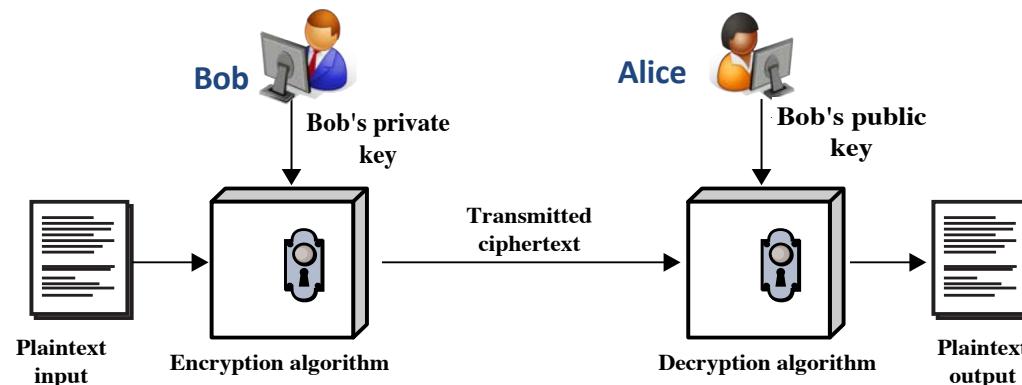
Thus, **SSL uses asymmetric cryptography because it is easier to apply generically (especially worldwide), and also easier to keep secure.**

Mecanismos e Infraestructuras de administración de claves públicas. El caso del DNI-e

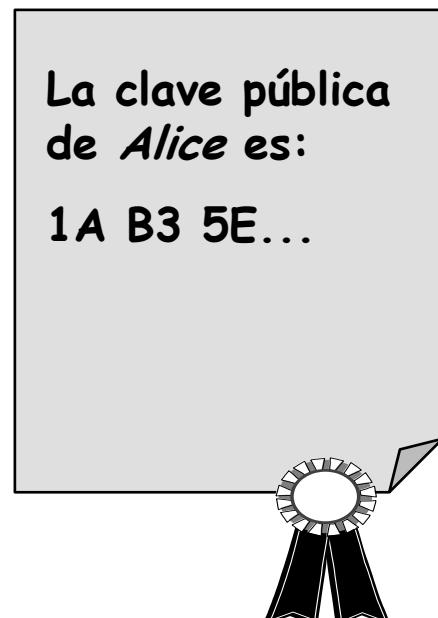
- ¿Cómo sabe *Bob* en este escenario si la clave pública de *Alice* es genuina?



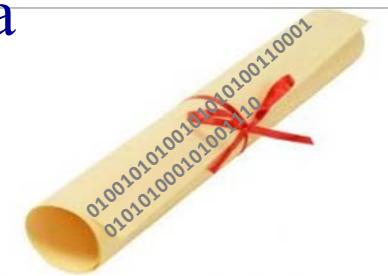
- ¿Cómo sabe *Alice* en este escenario si la clave pública de *Bob* es genuina?



- Las preguntas anteriores equivalen a plantearse ¿cómo garantizar que **las claves públicas** de *Alice* y *Bob* **son auténticas**?
- Veamos, como ejemplo, el caso en el que *Bob* necesita la clave pública de *Alice*
 - *Bob* necesitaría algún documento digital con algún “**sello de garantía**”, o sea, algo equivalente a lo que en papel sería:



- En realidad, ¿qué información relevante habría de contener ese documento digital para optimizar su utilidad?
 - La **identidad del usuario** al respecto del cual se ofrece información (*Alice* en la figura anterior)
 - El valor de la **clave pública** de Alice (o sea, 1A B3 5E ...)
 - Algo que identifique únicamente a ese documento entre otros muchos (por ejemplo, un **número de serie**)
 - ¿sirve la identidad del usuario para este menester?
 - No, porque un usuario puede tener más de un par <clave pública, clave privada>
 - Algo que indique “desde” cuándo y “hasta” cuándo es válido el documento digital (por ejemplo, una fecha de **emisión** y una de **expiración**)
 - La identidad de **quien emite** el documento
 - La **firma digital** de quien emite el documento

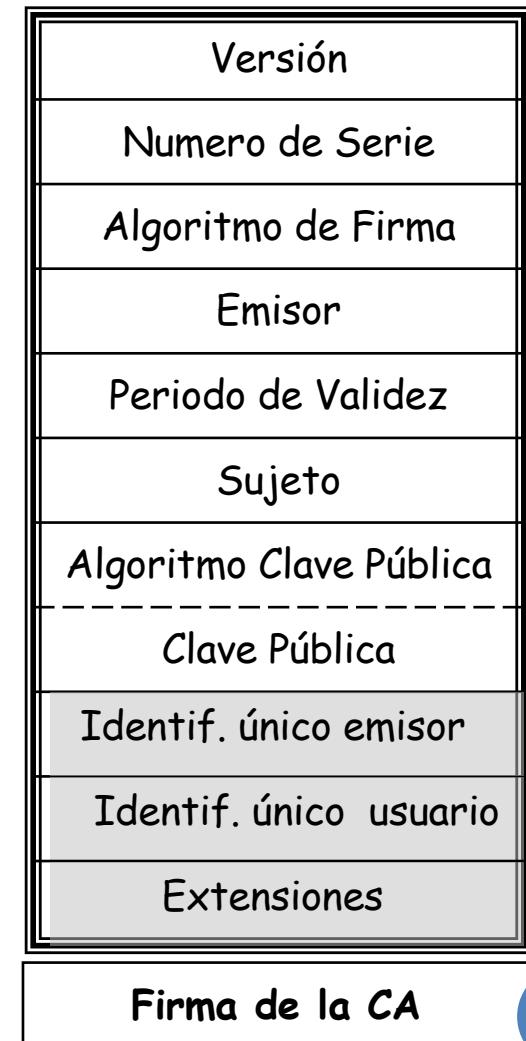


- El documento digital con esa información se denomina **certificado digital** o **certificado de clave pública**
 - Es la firma digital de un documento (que contiene la información antes mencionada) la que garantiza que cierta clave pública pertenece a un determinado usuario
- Se denomina **Autoridad de Certificación** a la *tercera parte confiable* (TTP) que emite y administra los certificados digitales de los usuarios de un sistema
 - Garantiza que una clave pública pertenece a cierto usuario inequívocamente identificado

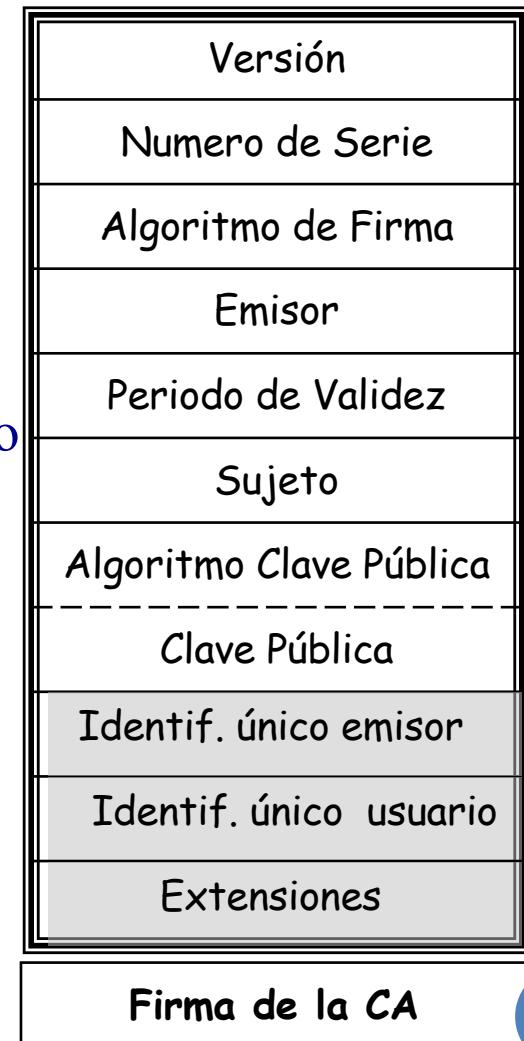


- La ITU-T ha definido una estructura estándar de certificado digital que ha sido adoptada internacionalmente: **certificado X.509**

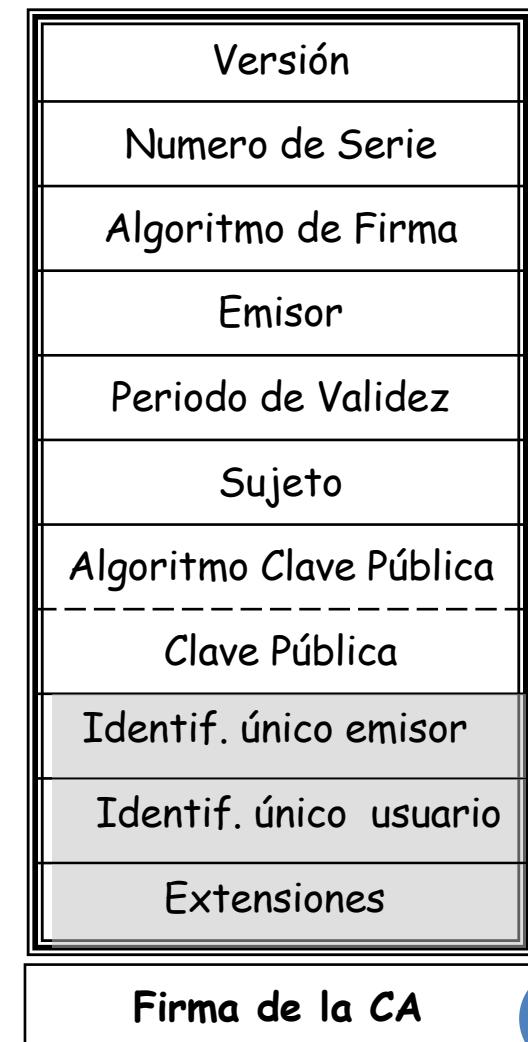
- **Versión**: Indica el número de versión de X.509 (o sea, 1, 2 ó 3)
- **Número de Serie**: Número de identificación único para este certificado digital, asignado por la CA
- **Algoritmo de Firma**: Identificador del algoritmo de firma digital usado por la CA para firmar el certificado
- **Emisor**: Nombre X.500 de la **CA emisora**
- **Periodo de Validez**: Fecha desde el que el certificado comienza a ser válido, y día y hora de expiración



- La ITU-T ha definido una estructura estándar de certificado digital que ha sido adoptada internacionalmente: **certificado X.509**
 - **Sujeto:** Nombre en formato X.500 del usuario cuya clave pública se está certificando
 - **Algoritmo Clave Pública:** Identificador del algoritmo de clave pública con el que se ha de utilizar la clave pública
 - **Clave Pública:** Valor de la clave pública del usuario
 - **Identificador único de emisor:** Cadena opcional para que el nombre de la **CA** no sea ambiguo, en caso de que esto pudiera ocurrir (versión 2)
 - **Identificador único de usuario:** Cadena opcional para que el nombre del **usuario** no sea ambiguo, en caso de que pudiera ocurrir (versión 2)



- La ITU-T ha definido una estructura estándar de certificado digital que ha sido adoptada internacionalmente: **certificado X.509**
 - *Extensiones*: Campo opcional para almacenar información de distinto tipo (versión 3)
 - **Firma**: firma digital de la CA sobre el valor hash del conjunto de los demás campos del certificado





Cristina

Entidad de certificación raíz

Caduca: jueves, 19 de marzo de 2026, 20:26:58 (hora estándar de Europa central)

✗ Este certificado raíz no es fiable

► Confiar

▼ Detalles

Nombre del sujeto

País SP

Estado/Provincia Malaga

Localidad Malaga

Empresa UMA

Unidad organizativa UMA

Nombre común Cristina

Dirección de correo ab@lcc.uma.es

Nombre del emisor

País SP

Estado/Provincia Malaga

Localidad Malaga

Empresa UMA

Unidad organizativa UMA

Nombre común Cristina

Dirección de correo ab@lcc.uma.es

Número de serie 00 D9 AE F5 9B 24 2D 04 FE

Versión 3

Algoritmo de firma SHA-1 con encriptación RSA (1.2.840.113549.1.1.5)

Parámetros ninguno/a

No válido antes de lunes, 21 de marzo de 2016, 20:26:58 (hora estándar de Europa central)

No válido después de jueves, 19 de marzo de 2026, 20:26:58 (hora estándar de Europa central)



Versión

Número de Serie

Algoritmo de Firma

Emisor

Periodo de Validez

Sujeto

Algoritmo Clave Pública

Clave Pública

Identif. único emisor

Identif. único usuario

Extensiones

Firma de la CA



Cristina

Entidad de certificación raíz

Caduca: jueves, 19 de marzo de 2026, 20:26:58 (hora estándar de Europa central)

✗ Este certificado raíz no es fiable

► Confiar

▼ Detalles

Nombre del sujeto

País SP
Estado/Provincia Málaga
Localidad Málaga
Empresa UMA
Unidad organizativa UMA
Nombre común Cristina
Dirección de correo ab@lcc.uma.es

Nombre del emisor

País SP
Estado/Provincia Málaga
Localidad Málaga
Empresa UMA
Unidad organizativa UMA
Nombre común Cristina
Dirección de correo ab@lcc.uma.es

Número de serie 00 D9 AE F5 9B 24 2D 04 FE

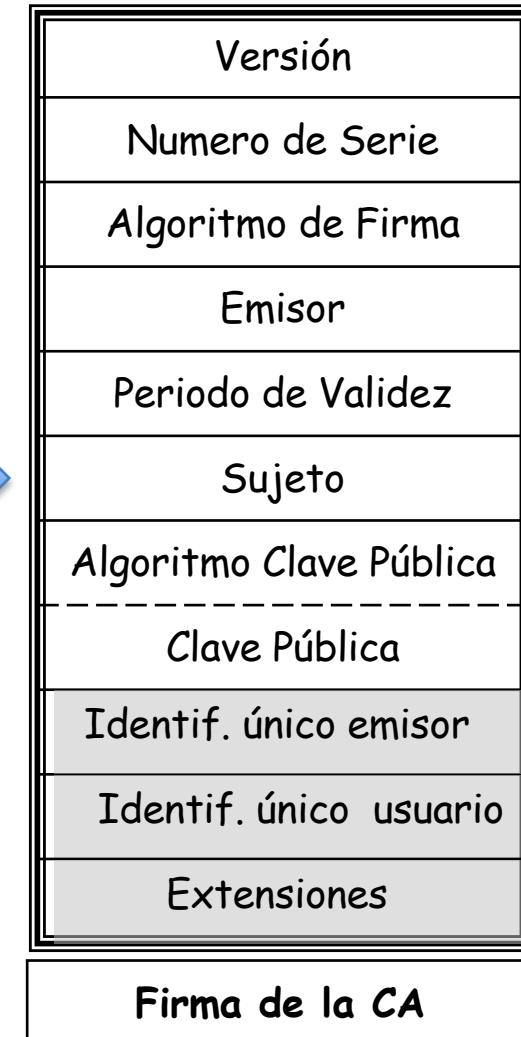
Versión 3

Algoritmo de firma SHA-1 con encriptación RSA (1.2.840.113549.1.1.5)

Parámetros ninguno/a

No válido antes de lunes, 21 de marzo de 2016, 20:26:58 (hora estándar de Europa central)

No válido después de jueves, 19 de marzo de 2026, 20:26:58 (hora estándar de Europa central)



Información de la clave pública

Algoritmo Encriptación RSA (1.2.840.113549.1.1.1)
Parámetros ninguno/a
Clave pública 128 bytes: BF F9 49 27 D5 E6 29 D5 ...
Exponente 65537
Tamaño de la clave 1024 bits
Uso de la clave Cualquiera

Firma 128 bytes: 98 43 60 F8 B4 C4 D7 E1 ...

Extensión Restricciones básicas (2.5.29.19)
Crítico NO
Entidad de certificación Sí

Extensión Identificador de clave del sujeto (2.5.29.14)
Crítico NO
Nombre de la clave 24 85 28 DC 50 C1 BD 39 5D D0 D5 72 A5 09 1B F4 73 9D AF 7F

Extensión Identificador de clave de entidad emisora (2.5.29.35)
Crítico NO
Nombre de la clave 24 85 28 DC 50 C1 BD 39 5D D0 D5 72 A5 09 1B F4 73 9D AF 7F

Nombre de directorio

País SP
Estado/Provincia Málaga

Localidad Málaga

Empresa UMA

Unidad organizativa UMA

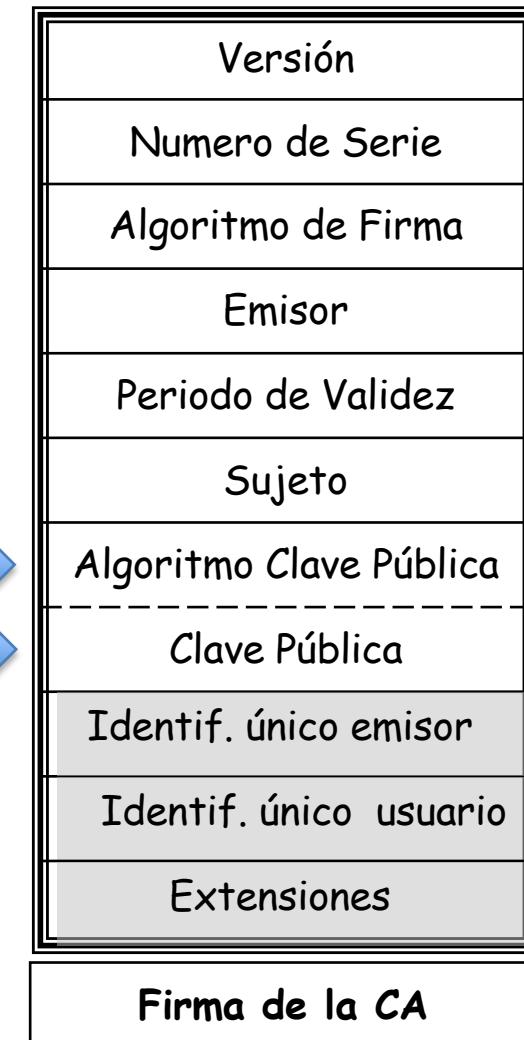
Nombre común Cristina

Dirección de correo ab@lcc.uma.es

Número de serie 00 D9 AE F5 9B 24 2D 04 FE

Huellas digitales

SHA1 8B 5F 16 E7 64 66 15 9C 89 F3 C1 13 44 94 44 A0 69 75 8F 90
MD5 D6 DB ED 1D 4B DC B3 42 17 31 78 D7 70 8E 0A 96



Información de la clave pública

Algoritmo Encriptación RSA (1.2.840.113549.1.1.1)
Parámetros ninguno/a
Clave pública 128 bytes: BF F9 49 27 D5 E6 29 D5 ...
Exponente 65537
Tamaño de la clave 1024 bits
Uso de la clave Cualquiera

Firma 128 bytes: 98 43 60 F8 B4 C4 D7 E1 ...

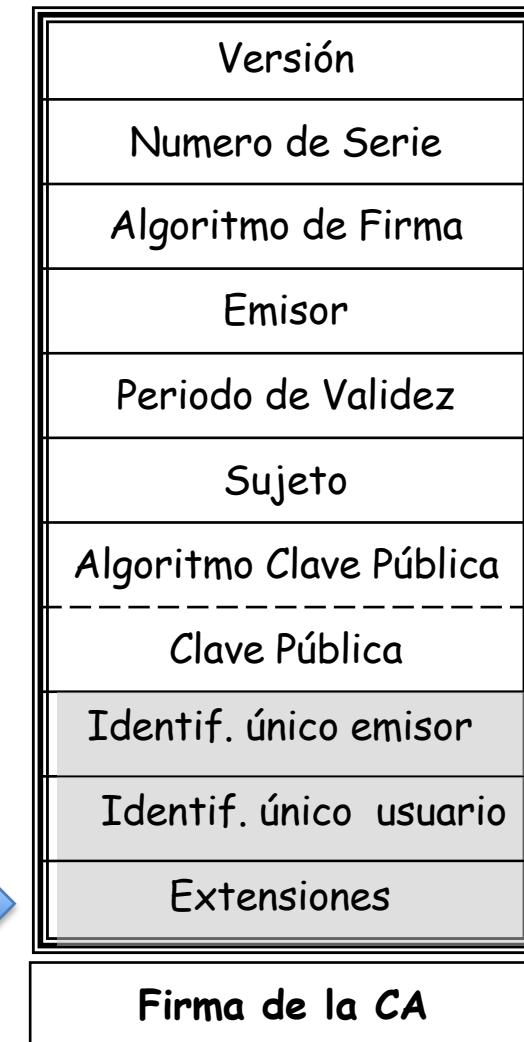
Extensión Restricciones básicas (2.5.29.19)
Crítico NO
Entidad de certificación Sí

Extensión Identificador de clave del sujeto (2.5.29.14)
Crítico NO
Nombre de la clave 24 85 28 DC 50 C1 BD 39 5D D0 D5 72 A5 09 1B F4 73 9D AF 7F

Extensión Identificador de clave de entidad emisora (2.5.29.35)
Crítico NO
Nombre de la clave 24 85 28 DC 50 C1 BD 39 5D D0 D5 72 A5 09 1B F4 73 9D AF 7F
Nombre de directorio
País SP
Estado/Provincia Málaga
Localidad Málaga
Empresa UMA
Unidad organizativa UMA
Nombre común Cristina
Dirección de correo ab@lcc.uma.es
Número de serie 00 D9 AE F5 9B 24 2D 04 FE

Huellas digitales

SHA1 8B 5F 16 E7 64 66 15 9C 89 F3 C1 13 44 94 44 A0 69 75 8F 90
MD5 D6 DB ED 1D 4B DC B3 42 17 31 78 D7 70 8E 0A 96



Información de la clave pública

Algoritmo Encriptación RSA (1.2.840.113549.1.1.1)
Parámetros ninguno/a
Clave pública 128 bytes: BF F9 49 27 D5 E6 29 D5 ...
Exponente 65537
Tamaño de la clave 1024 bits
Uso de la clave Cualquiera

Firma 128 bytes: 98 43 60 F8 B4 C4 D7 E1 ...

Extensión Restricciones básicas (2.5.29.19)
Crítico NO
Entidad de certificación Sí

Extensión Identificador de clave del sujeto (2.5.29.14)
Crítico NO
Nombre de la clave 24 85 28 DC 50 C1 BD 39 5D D0 D5 72 A5 09 1B F4 73 9D AF 7F

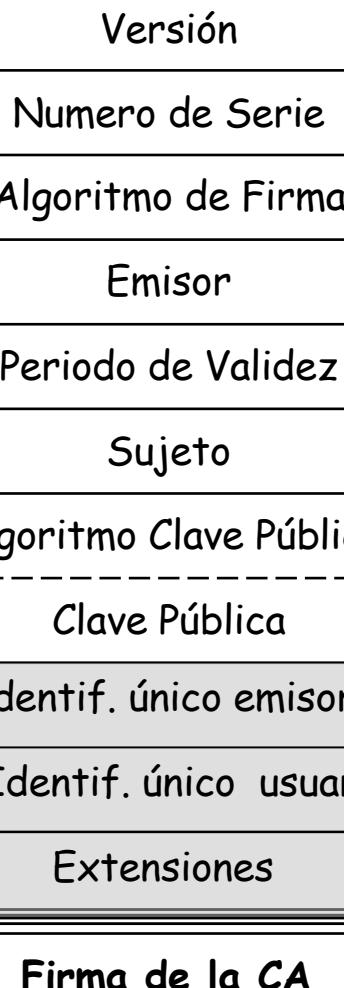
Extensión Identificador de clave de entidad emisora (2.5.29.35)
Crítico NO
Nombre de la clave 24 85 28 DC 50 C1 BD 39 5D D0 D5 72 A5 09 1B F4 73 9D AF 7F

Nombre de directorio
País SP
Estado/Provincia Malaga
Localidad Malaga
Empresa UMA
Unidad organizativa UMA
Nombre común Cristina
Dirección de correo ab@lcc.uma.es

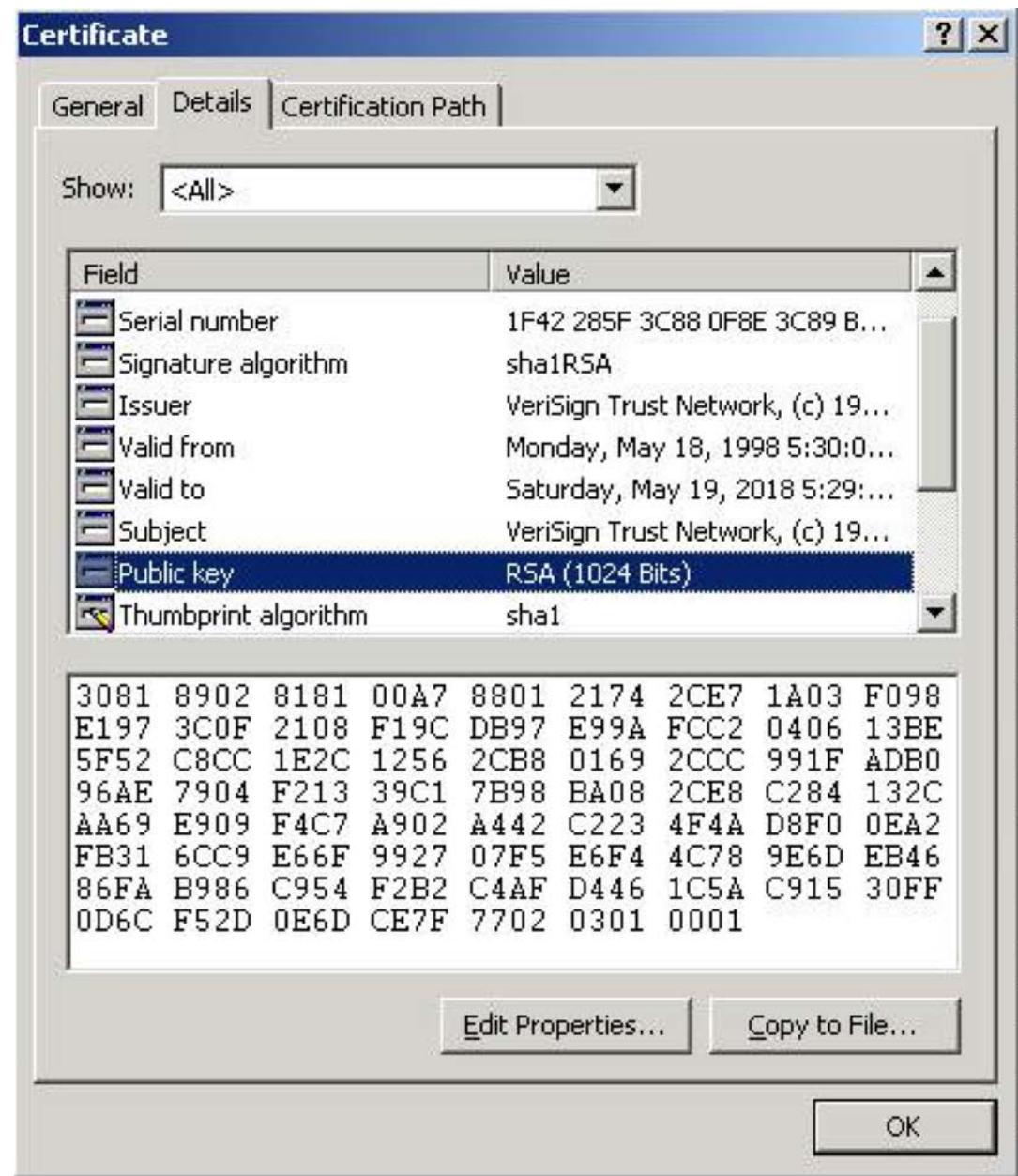
Número de serie 00 D9 AE F5 9B 24 2D 04 FE

Huellas digitales

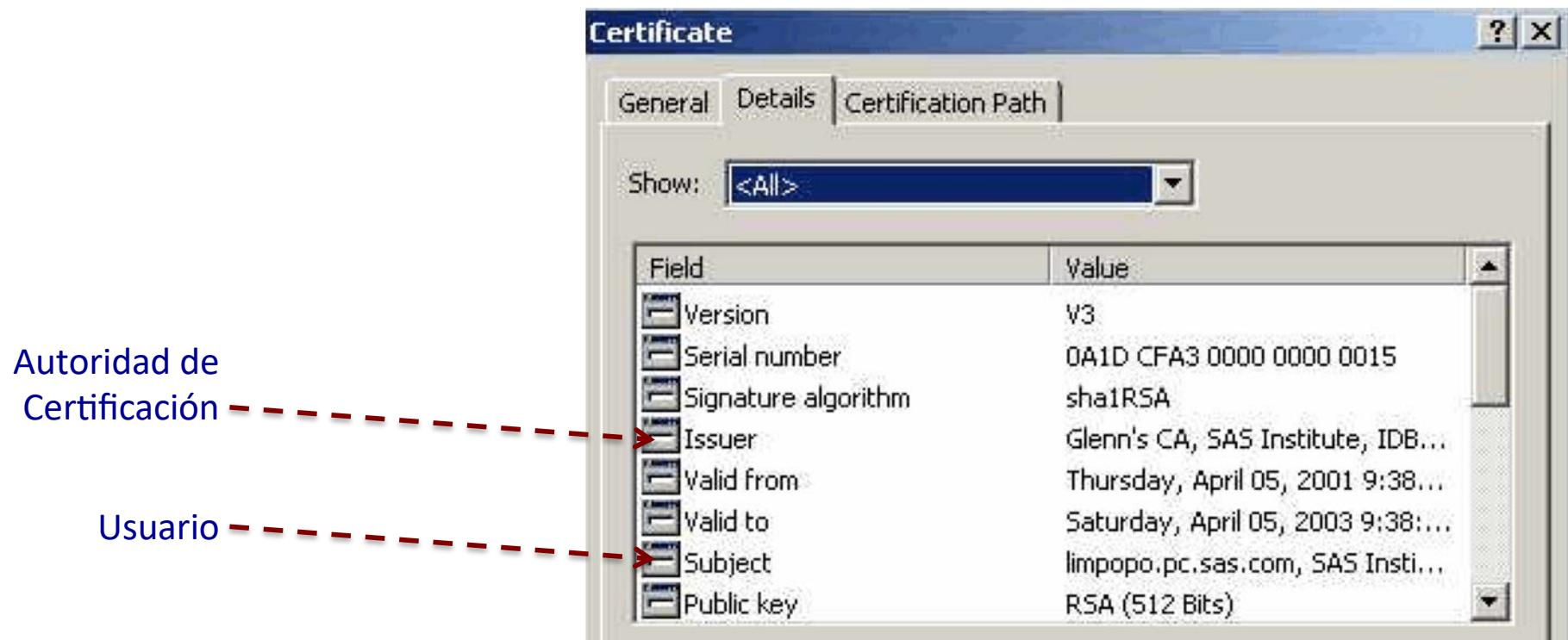
SHA1 8B 5F 16 E7 64 66 15 9C 89 F3 C1 13 44 94 44 A0 69 75 8F 90
MD5 D6 DB ED 1D 4B DC B3 42 17 31 78 D7 70 8E 0A 96



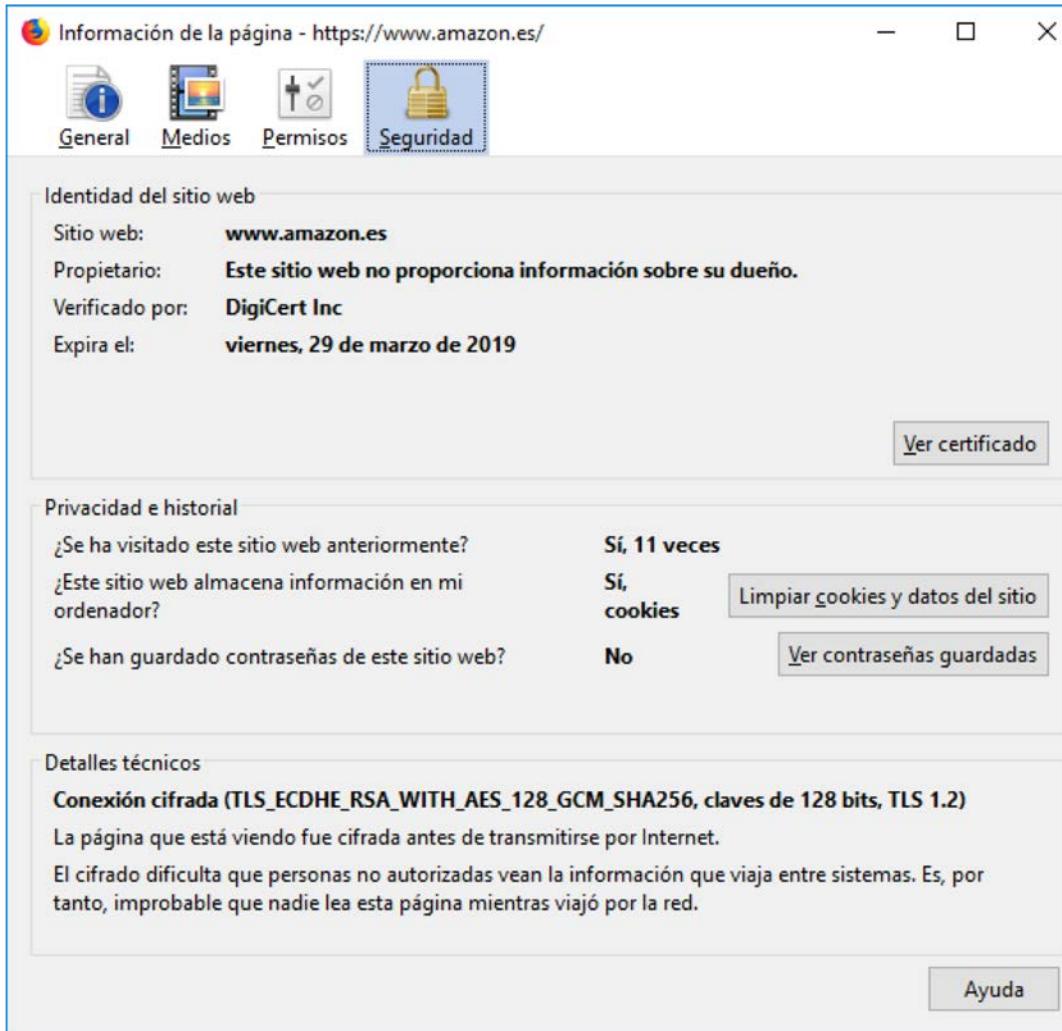
- Ejemplo de certificado X.509 instalado en un navegador



- Ejemplo de certificado X.509 instalado en un navegador:



- Ejemplo de certificado X.509 de una página HTTPS:

 Información de la página - <https://www.amazon.es/>

General Medios Permisos Seguridad

Identidad del sitio web

Sitio web: www.amazon.es
Propietario: Este sitio web no proporciona información sobre su dueño.
Verificado por: DigiCert Inc
Expira el: viernes, 29 de marzo de 2019

Ver certificado

Privacidad e historial

¿Se ha visitado este sitio web anteriormente? Sí, 11 veces
Sí, cookies Limpiar cookies y datos del sitio

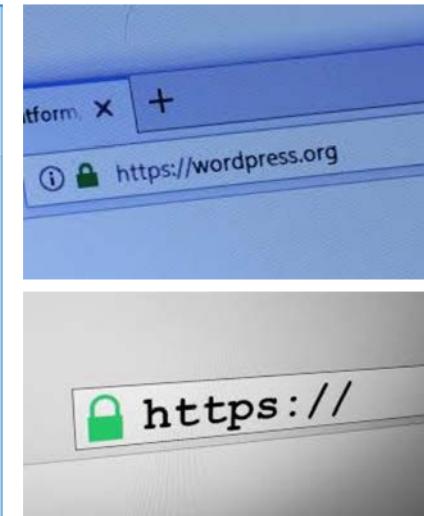
¿Este sitio web almacena información en mi ordenador?

¿Se han guardado contraseñas de este sitio web? No Ver contraseñas guardadas

Detalles técnicos

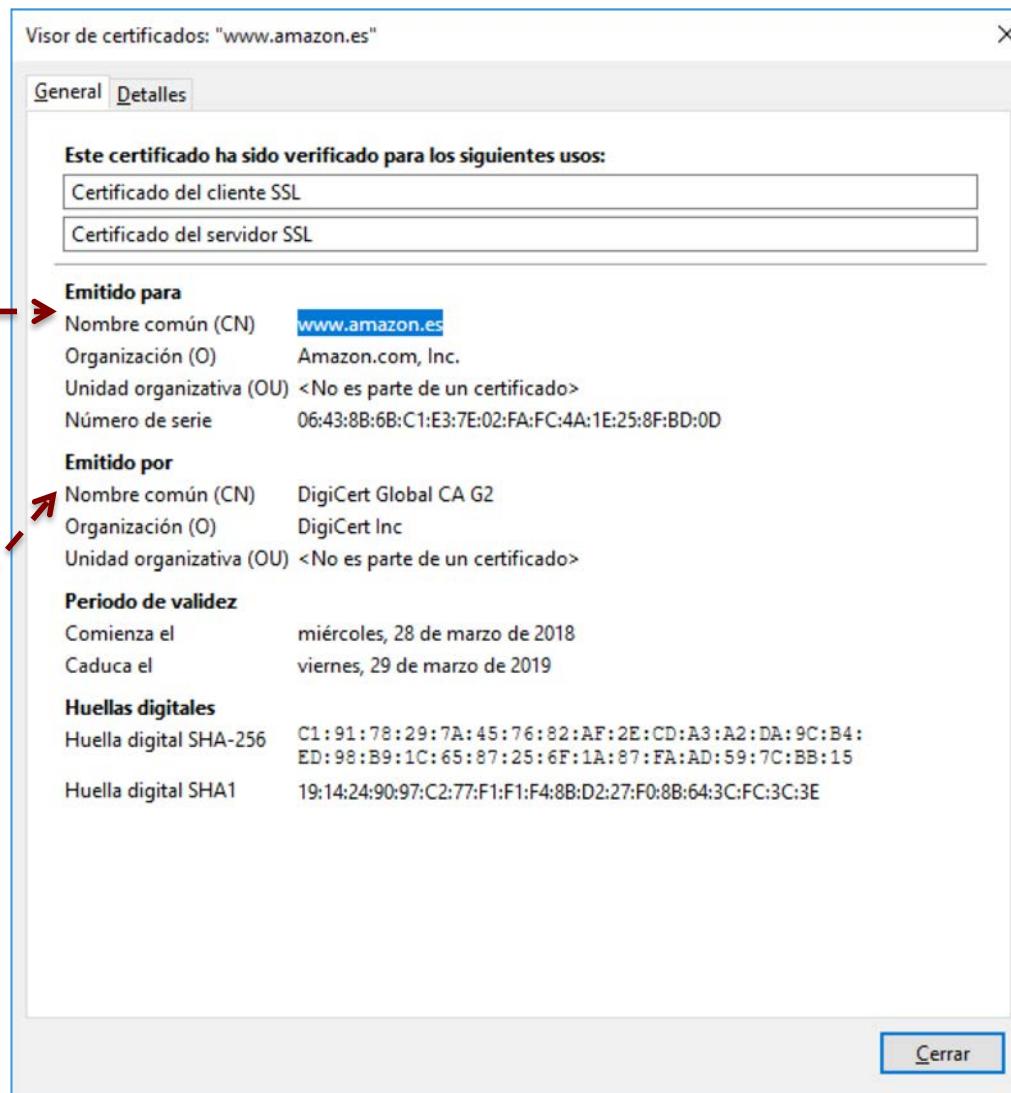
Conexión cifrada (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, claves de 128 bits, TLS 1.2)
La página que está viendo fue cifrada antes de transmitirse por Internet.
El cifrado dificulta que personas no autorizadas vean la información que viaja entre sistemas. Es, por tanto, improbable que nadie lea esta página mientras viajó por la red.

Ayuda



- Ejemplo de certificado X.509 de una página HTTPS:

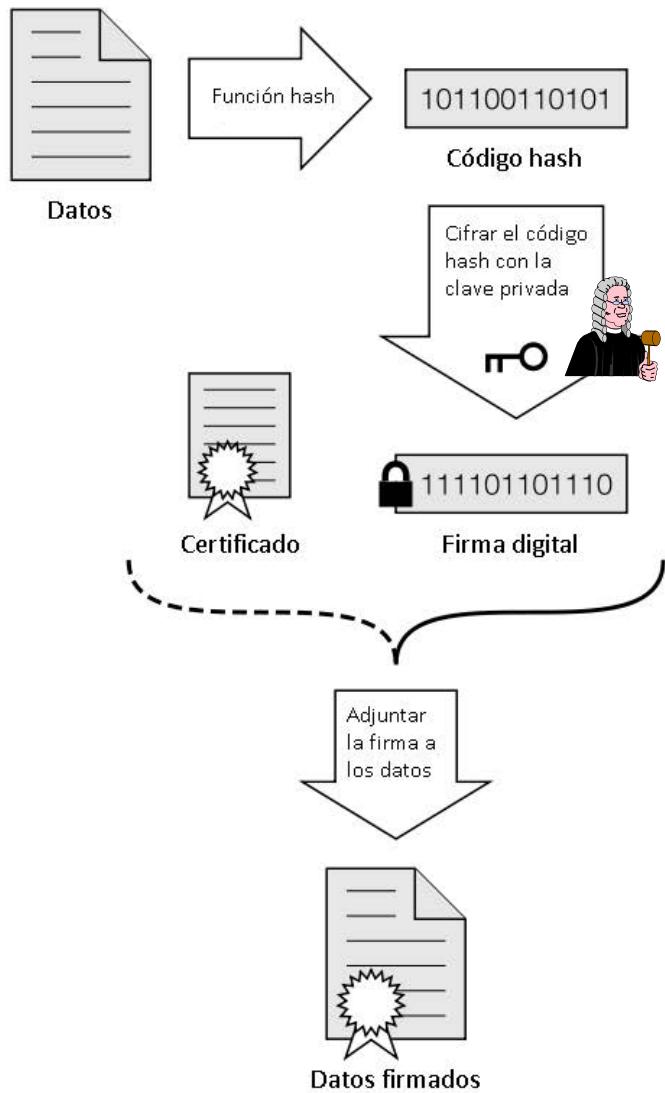
Usuario - - ->
Autoridad de
Certificación



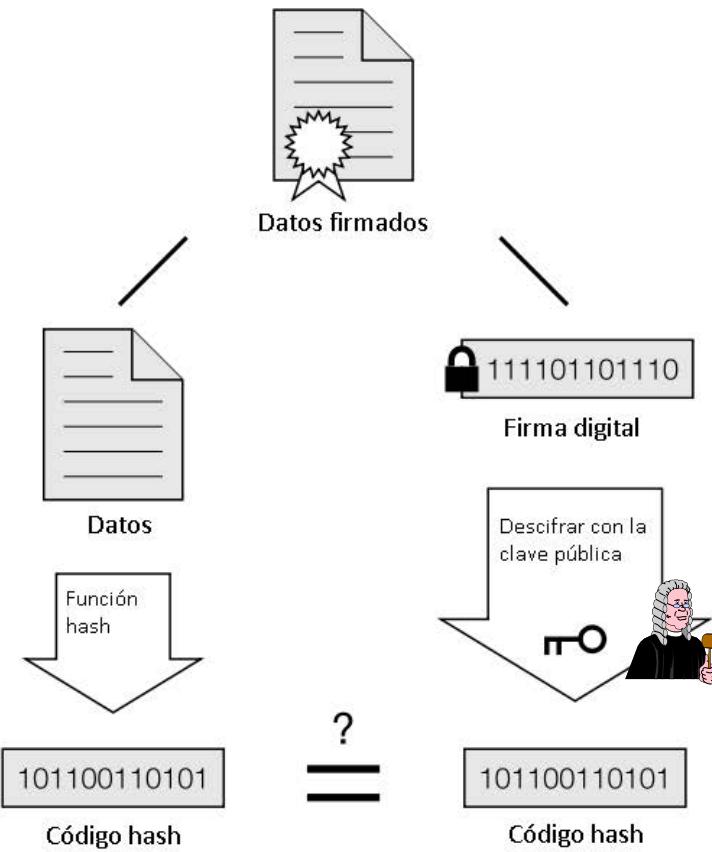




Firma Digital



Comprobación de una Firma



Si los códigos hash coinciden, la firma es válida

Certificados digitales



<http://www.cacert.org>

¿Nuevo en CACert?

CAcert.org es una autoridad certificadora dirigida por la comunidad que emite certificados gratuitos al público.

El objetivo de CACert es promover el conocimiento y la educación sobre la seguridad informática a través del cifrado, ofreciendo específicamente certificados criptográficos. Estos certificados se pueden utilizar para firmar digitalmente y cifrar mensajes de correo electrónico, autenticar y autorizar usuarios que se conectan a sitios web y asegurar la transmisión de datos a través de Internet. Cualquier aplicación que tenga soporte del protocolo Secure Socket Layer (SSL o TLS) puede hacer uso de los certificados firmados por CACert, así como cualquier aplicación que utilice certificados X.509, por ejemplo para el cifrado o firmado de código y las firmas digitales en documentos.

Si desea obtener certificados gratuito emitidos en su nombre, [úñase a la Comunidad CACert](#).

Si desea utilizar los certificados emitidos por CACert, lea la CACert [Root Distribution License](#). Esta licencia se aplica cuando se utilizan [claves raíz](#) de CACert.

ÚLTIMAS NOTICIAS

Accréditation à / Assurance in Paris

Le prochain rendez-vous mensuel à Paris à lieu le mardi 21 mars 2017 entre 19:00 heures et 20:00 heures. Nous vous proposons une rencontre pour toutes personnes intéressées par CACert. Validation, certification, accréditation de vos identités et informations sur CACert. Bar de l'Hôtel Novotel Les Halles 8, place Marguerite de Navarre Paris 1er, Mo Châtelet Pour [...]

CAcert 2017

February brought the start of the exhibition season for CACert with our presence at FOSDEM – one of the biggest Europe-wide developer conferences in Brussels, Belgium. Of course we performed our well-known assurances, which is very popular at such events, with which CACert safeguards its certificates by checking users' ID documents. This allows us to [...]

CAcert and secure-u e.V. present at FOSDEM 2017

CAcert and secure-u e.V. will be present at FOSDEM 2017, the Free and Open Source Software Developers' European Meeting in Brussels, on February 4th and February 5th. Booth (Sat + Sun) Keysigning Party If you want to help at our booth, register yourself on our events wiki page for FOSDEM 2017 planning. CU at FOSDEM [...]

[[MÁS Noticias](#)]

Dirigido a los miembros de la comunidad CACert

¿Ha superado ya la [Prueba de Notario](#) de CACert?

¿Ha leído ya el [Acuerdo de Comunidad](#) de CACert?

Para encontrar la documentación general y ayuda visite el [sitio de Documentación Wiki](#) de CACert. Para leer acerca de directrices específica, lea la [página de Directrices Aprobadas](#) de CACert.

[Alta en CACert.org](#)
[Darse de alta](#)
[Acuerdo de la comunidad](#)
[Certificado raíz](#)

[Mi cuenta](#)
[Iniciar sesión con contraseña](#)
[Contraseña olvidada](#)
[Iniciar sesión con Net Cafe](#)
[Iniciar sesión con certificado](#)

[+ Acerca de CACert.org](#)

[+ Traducciones](#)

[Publicidad](#)

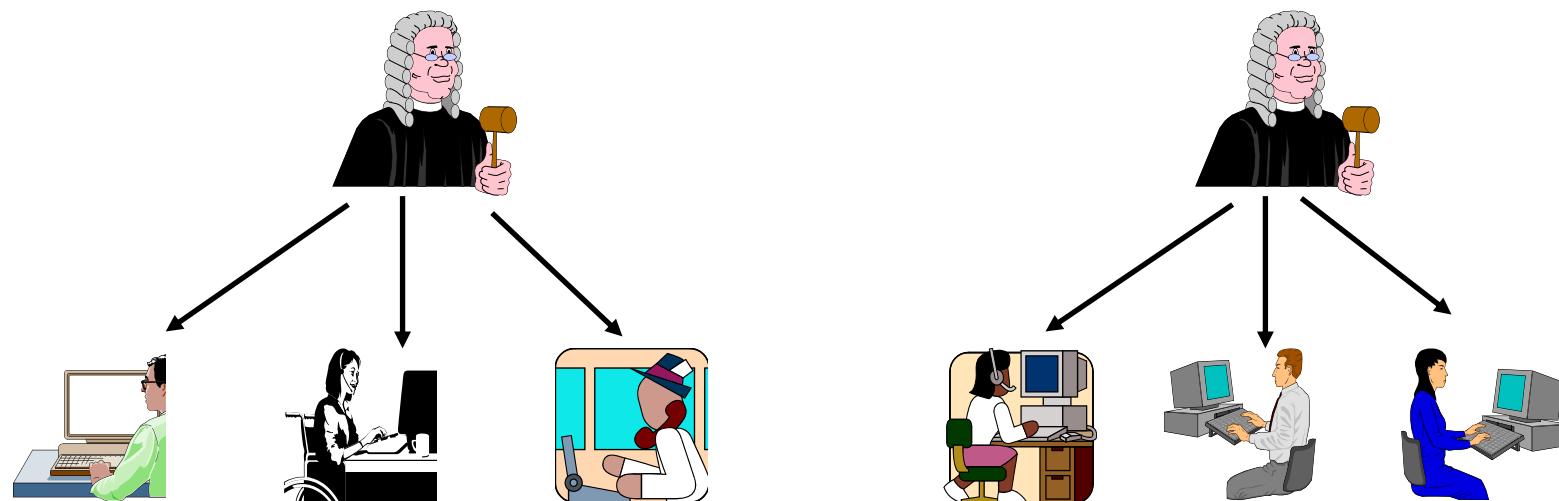
Certificados digitales



<https://letsencrypt.org/>

The screenshot shows the official website for Let's Encrypt. At the top left is the Let's Encrypt logo, which consists of a stylized sun icon followed by the text "Let's Encrypt". To the right of the logo is a navigation bar with links for "Documentation", "Get Help", "Donate", "About Us", and "Languages". Above the navigation bar, there is a small text "LINUX FOUNDATION COLLABORATIVE PROJECTS". The main content area features a large, semi-transparent white box containing the text: "Let's Encrypt is a **free**, **automated**, and **open** Certificate Authority. This text is overlaid on a background of abstract geometric shapes in shades of blue, green, and orange. At the bottom of the white box are two blue-outlined buttons: "Get Started" on the left and "Sponsor" on the right.

- La situación ideal sería que una única CA pudiera certificar a todos los usuarios de Internet
- Sin embargo, la situación real es bien distinta, dado que existe una gran multiplicidad de grupos de usuarios en Internet, y distribuidos geográficamente, lo que implica la necesidad de **múltiples CAs**



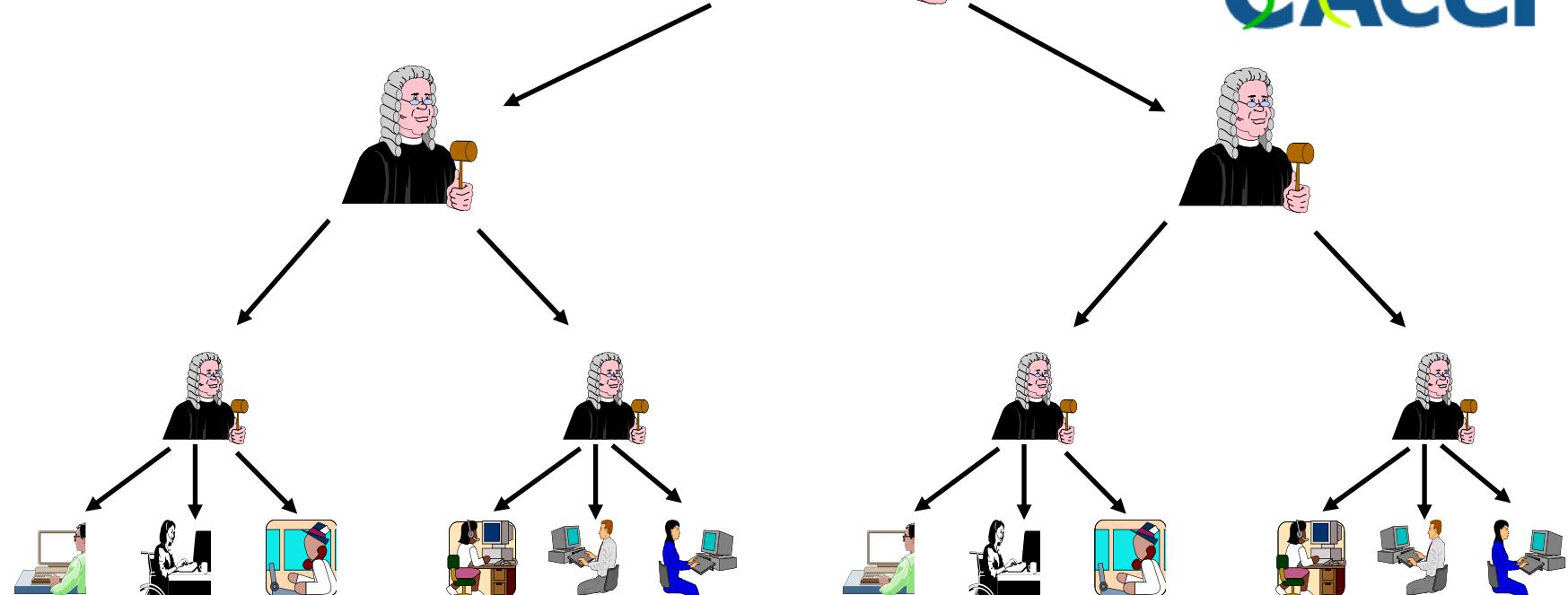
Certificados digitales



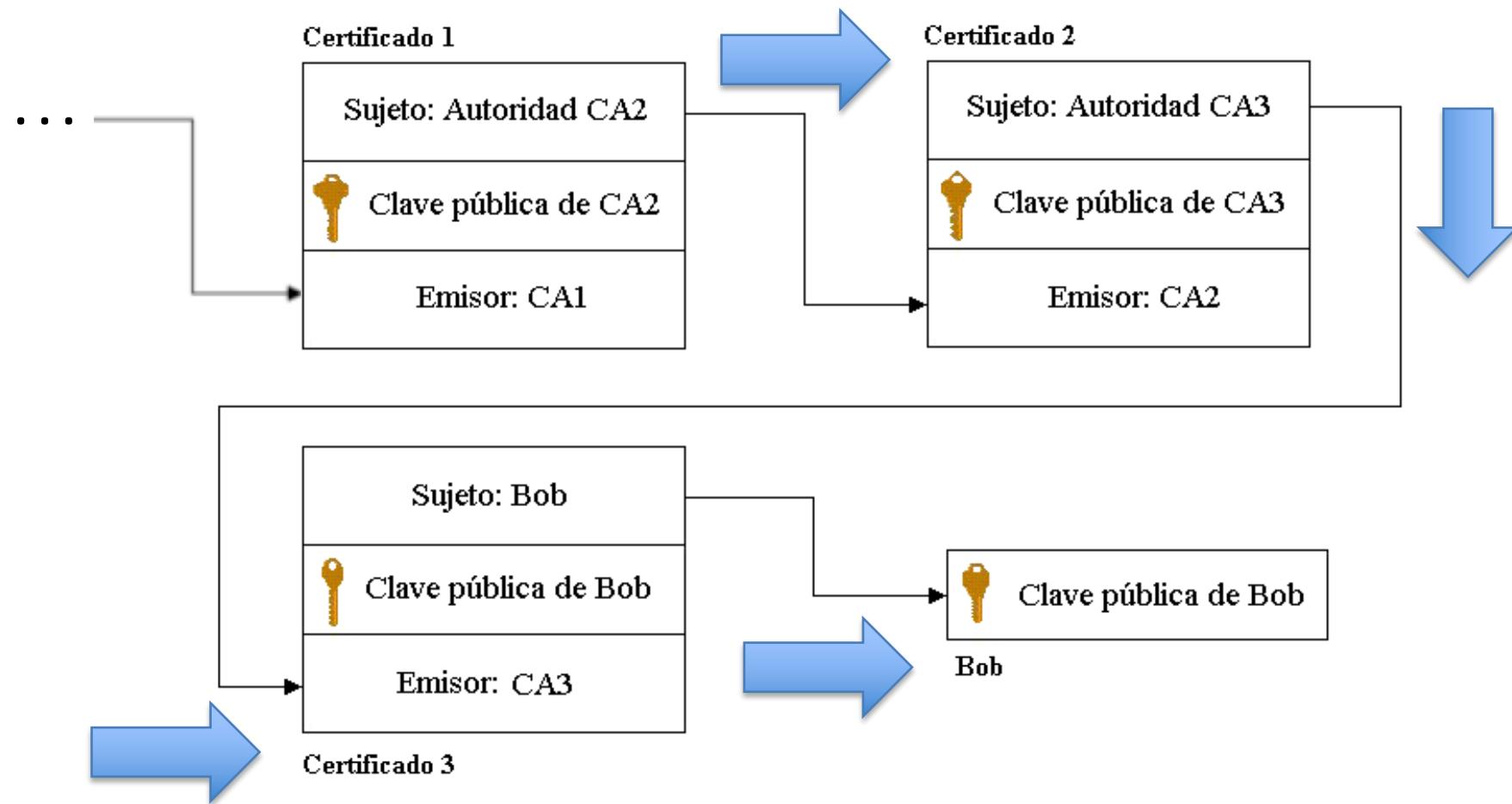
✓eriSign®



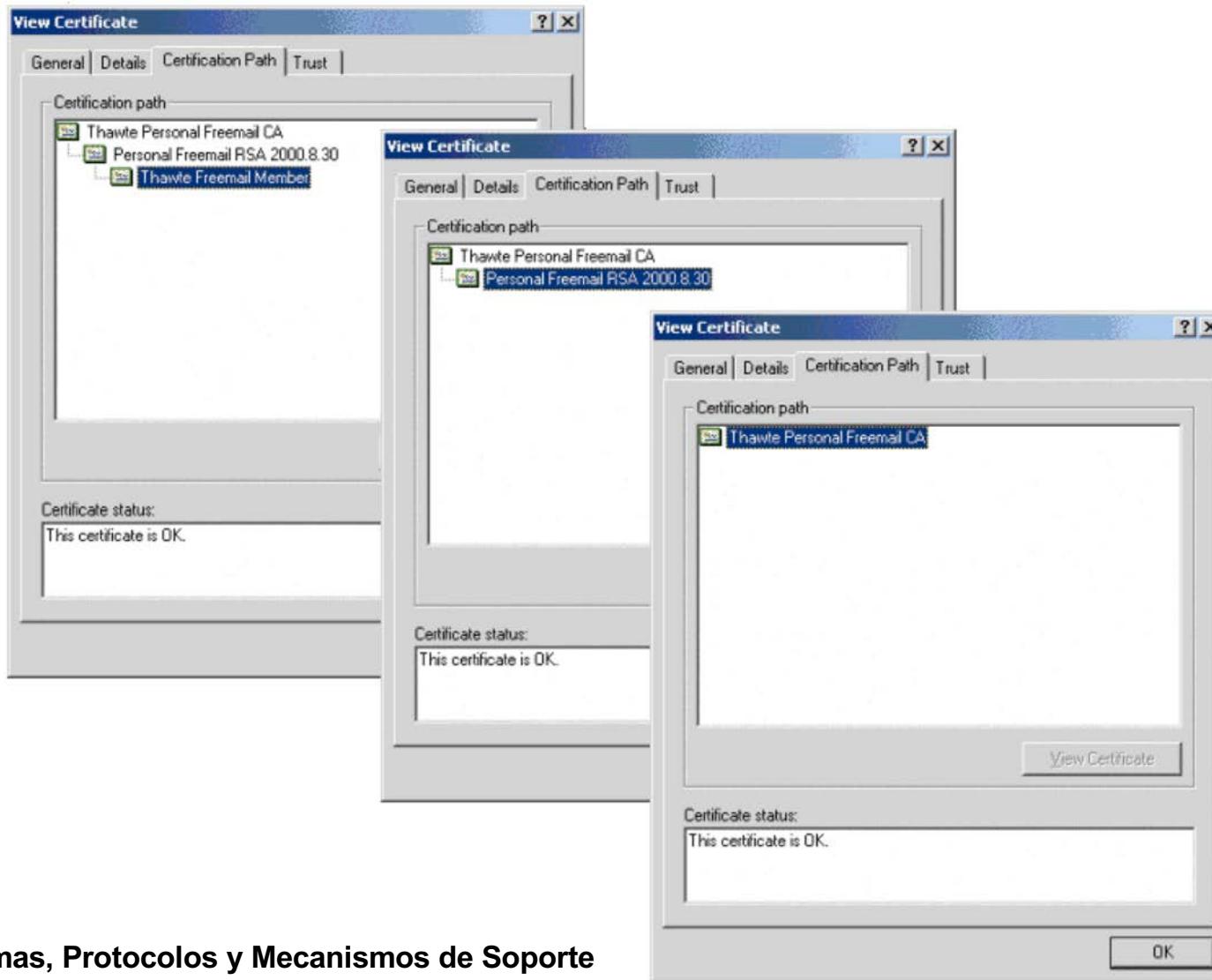
CAcert



- Entre las CAs se utiliza de forma recursiva el esquema de certificación, creándose las **cadenas de confianza** (o **caminos de certificación**)



- Ejemplo de camino de certificación en un navegador:

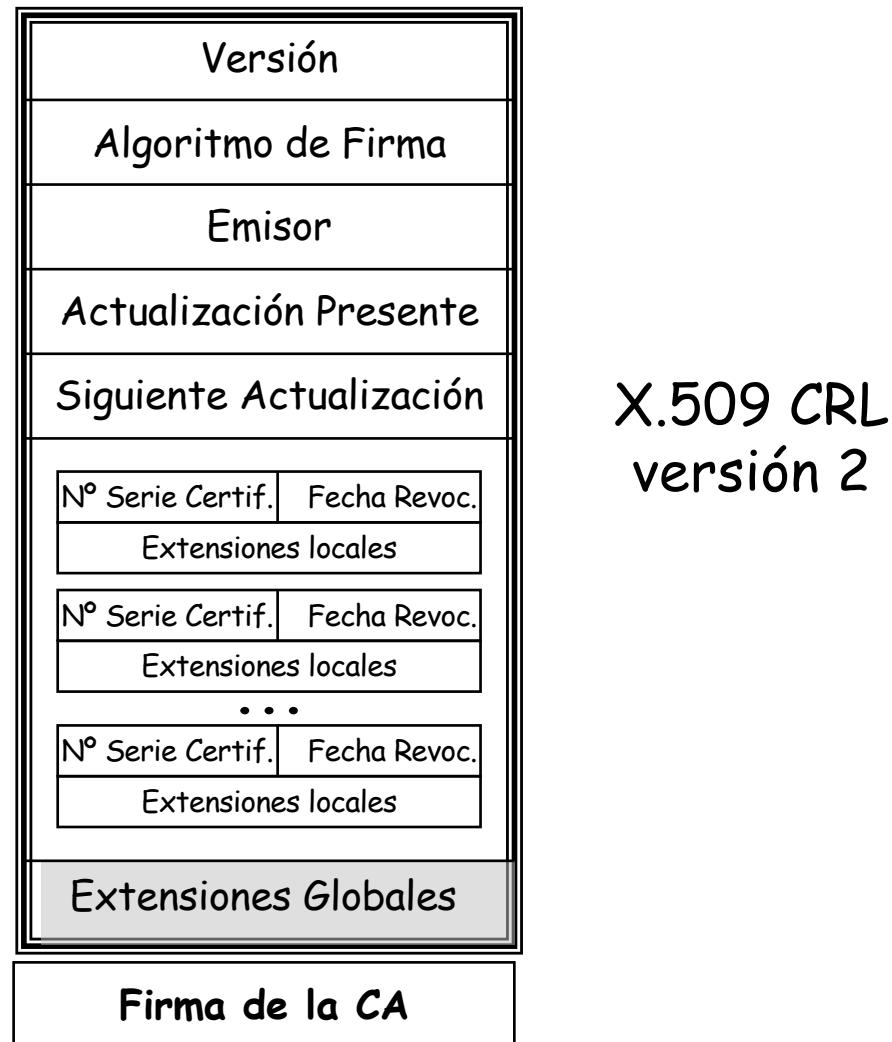


- Esas cadenas de confianza se forman gracias a la infraestructura de CAs, denominada **Infraestructura de Clave Pública (PKI – Public Key Infrastructure)**
 - Una PKI proporciona el marco subyacente que permite la implantación de la tecnología de clave pública
- Servicios ofrecidos por una PKI:
 1. Emisión de Certificados
 2. Distribución de Certificados
 3. Obtención de Certificados
 4. Certificación Cruzada
 5. Generación de Claves
 6. Actualización de Claves
 7. Salvaguarda y Recuperación de Claves
 8. Revocación y Suspensión de Certificados

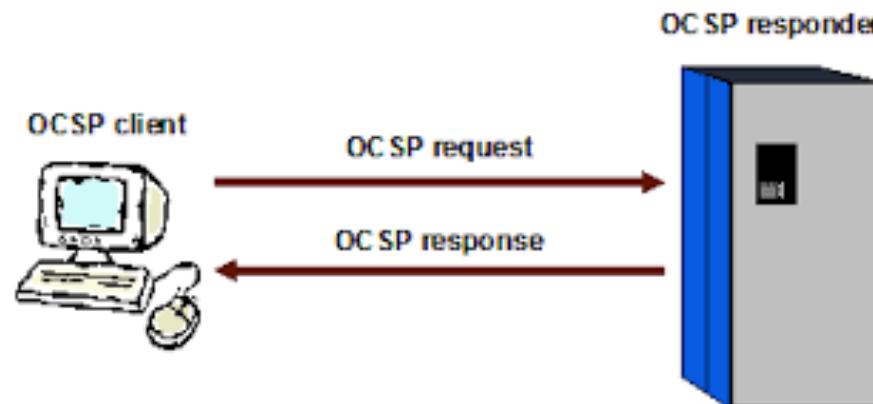
- Revocación de certificados
 - Puede ser recomendable **invalidar** (revocar) **un certificado antes de la fecha de expiración** cuando:
 - la clave pública deja de ser válida
 - el usuario identificado en el certificado no se considera por más tiempo un usuario con potestad sobre la clave privada correspondiente
 - varía la información dentro del certificado
 - La **CA se encarga de realizar la revocación**, bajo petición del usuario
 - ha de **publicar** esa información acerca del estado del certificado para que el resto de usuarios puedan realizar la comprobación antes de usarlo
 - La comunidad Internet y la ITU-T han desarrollado el concepto de **Lista de Revocación de Certificados, CRL**, como mecanismo de revocación
 - Una CRL es una lista (con timestamping) de certificados revocados, **firmada por la autoridad que emitió los certificados**

- Escenario típico de uso:
 - Para que *Bob* verifique la firma de *Alice* sobre un documento digital, no sólo ha de verificar el certificado de *Alice* y su validez; además, ha de comprobar que ese certificado no está en la CRL
 - o sea, ha de adquirir la versión más reciente de la CRL y confirmar que el número de serie del certificado de *Alice* no está en tal CRL
- Una CA emite CRLs regularmente (cada hora, día, semana,...) con independencia de que se hayan producido nuevas revocaciones
- El intervalo de emisión de CRLs depende de la política de certificación de la CA
- El certificado **se borra de la CRL cuando alcanza la fecha de expiración** (o sea, cuando se hubiese producido su caducidad natural)

- Estructura de una CRL, según el estándar X.509:



- El protocolo ***Online Certificate Status Protocol (OCSP)*** es otra solución de revocación (RFC 6960)
 - Define un formato estándar para mensajes de **peticiones y respuestas**
 - Su funcionamiento se basa en que un **usuario puede confirmar on-line el status de un certificado** mediante la ejecución de una transacción con un servidor (Responder) OCSP asociado a la CA
 - La CA debe poner a disposición de todos los usuarios potenciales un **servicio online de alta disponibilidad**, y además, el servicio ha de proporcionarse dentro de un entorno seguro
 - El Respondedor OCSP puede ser o bien la misma CA o alguna entidad autorizada por ella



El caso del DNI-e

- Una **tarjeta inteligente** o smartcard es una tarjeta que incluye un chip cuya función puede ser variada:
 - desde **simplemente almacenar** cierta información en su memoria interna (con o sin medidas de protección) ...
 - ... hasta realizar **complejos cálculos criptográficos** y encargarse de proteger el acceso a las claves que almacena
- Su uso se extiende hoy a muchos sectores:
 - tarjetas de fidelización
 - tarjetas bancarias
 - tarjetas de parking
 - documentos de identificación (DNI electrónico o pasaporte electrónico)
 - etc.



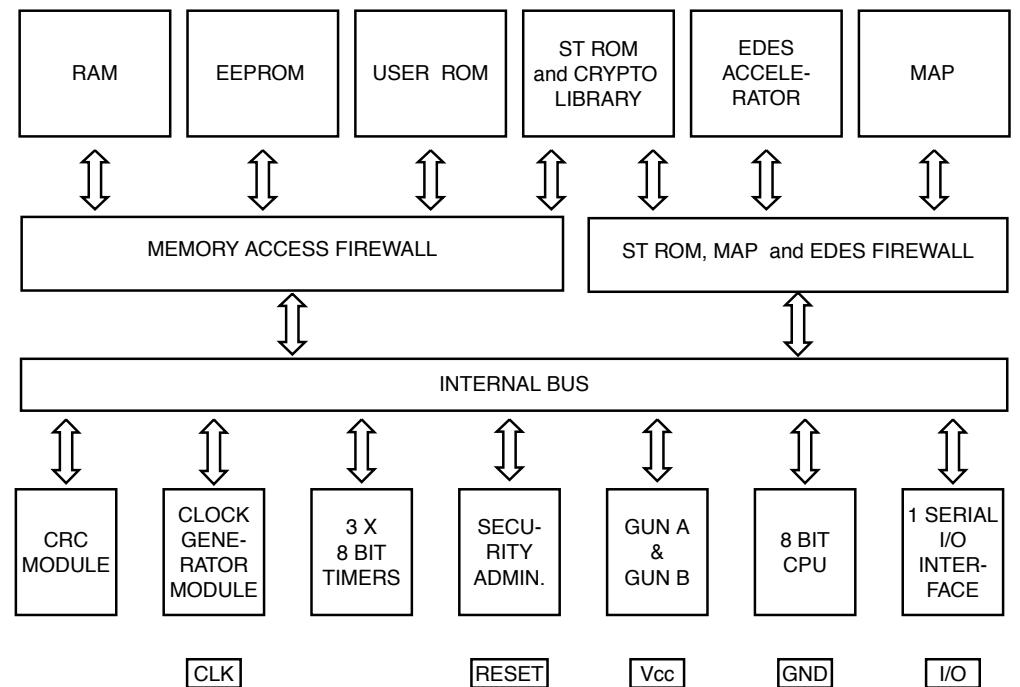
- Si atendemos al **método de comunicación o interfaz** con el circuito integrado, las smartcards se clasifican en:
 - **Tarjetas de contacto**
 - **Tarjetas sin contacto**
- Si atendemos a las **capacidades del chip**, se clasifican en:
 - **Tarjetas de memoria:** sólo contienen datos, y no albergan aplicaciones
 - Uso: identificación y control de acceso sin altos requisitos de seguridad
 - **Tarjetas microprocesadas:** albergan datos y aplicaciones
 - Uso: pago con monederos electrónicos
 - **Tarjetas criptográficas:** tarjetas microprocesadas avanzadas que incluyen módulos hardware para la ejecución de cifrados y firmas digitales
 - Uso: puede almacenar de forma segura un certificado digital (y su clave privada), así como firmar documentos o autenticarse
 - El procesador de la tarjeta realiza la firma



- El DNI electrónico, a través de las capacidades criptográficas que aporta, permite:
 - **identificación en medios telemáticos**
 - **firmar electrónicamente**
- El DNI-e está dotado con el chip ST19WL34 (STMicroelectronics), compuesto por:
 - microprocesador securizado de 8 bits
 - 6 Kb de memoria RAM
 - 224 KB de memoria ROM para el almacenamiento del sistema operativo y código de programas
 - 34 KB de memoria EEPROM para el almacenamiento de datos personales con tecnología de almacenamiento fiable y código de corrección de errores
- El chip ofrece una retención de datos de al menos 10 años, y una resistencia de 500.000 ciclos de borrado y escritura



- Este chip se caracteriza por incorporar también:
 - procesador aritmético modular (MAP) de 1088 bits para **criptografía de clave pública**
 - motor de aceleración por hardware de los **algoritmos DES y triple-DES**
 - módulo para el cálculo de **funciones CRC**
 - interfaz de entrada/salida serie
 - generador de números aleatorios
 - bus de interconexión interno
 - 3 timers de 8 bits
 - reloj interno



- La tabla muestra algunas medidas de tiempo de la ejecución de operaciones criptográficas

1. Typical values, independent from external clock frequency and supply voltage.
2. CRT: Chinese Remainder Theorem.

Function	Speed ⁽¹⁾
RSA 1024 bits signature with CRT ⁽²⁾	85 ms
RSA 1024 bits signature without CRT ⁽²⁾	282 ms
RSA 1024 bits verification ($e='\$10001'$)	5.5 ms
RSA 1024 bits key generation	2.5 s
RSA 2048 bits signature with CRT ⁽²⁾	570 ms
RSA 2048 bits verification ($e='\$10001'$)	91 ms
Triple DES (with enhanced security)	58.0 μ s
Single DES (with enhanced security)	43.0 μ s

- El sistema operativo que gestiona el chip se denomina DNIe v1.1, desarrollado por la FNMT a partir de las especificaciones funcionales de la Dirección General de Policía
 - este sistema operativo ha sido sometido con posterioridad a los perfiles de protección de la certificación *Common Criteria*

- El DNI-e contiene dos certificados digitales asociados al titular:
 - **certificado de autenticación**: asegura que la comunicación electrónica se realiza con el titular del DNI, pero no demuestra voluntad de firma
 - restringido a operaciones para confirmar la identidad y acceso seguro a sistemas remotos
 - **certificado de firma**: para la firma de documentos, garantizando la integridad del documento y el no repudio de origen
- Cuenta también con un **certificado de componente**, emitido para autenticar al propio chip y cifrar la comunicación con él
 - de forma similar a como se utiliza un certificado SSL en un servidor Web
- El generador interno de números aleatorios origina el par de claves de cada certificado, en presencia del ciudadano:
 - se garantiza que sólo existirá una copia de cada clave privada, y que ésta residirá siempre en el interior del chip

- La información de la **memoria EEPROM** del chip está distribuida en tres zonas, con diferentes niveles y condiciones de acceso
- Las tres son sólo accesibles para realizar **operaciones de lectura**, no siendo posible para el ciudadano escribir o grabar datos
 - **zona pública**: accesible sin restricciones
 - certificado CA emisora
 - claves Diffie-Hellman
 - certificado X.509 de componente
 - **zona privada**: accesible por el ciudadano mediante la utilización de su PIN
 - certificado de autenticación (identificación)
 - certificado de firma
 - **zona de seguridad**: accesible por el ciudadano de forma exclusiva en los puntos de actualización del DNI-e (en las comisarías)
 - datos de filiación del ciudadano
 - fotografía del titular
 - imagen de la firma manuscrita

Smart Card -- el caso del DNI-e

The screenshot shows the official website for the Spanish National Police's electronic services. The header features the police crest and the text "CUERPO NACIONAL DE POLICÍA". The main banner is titled "DNI y Pasaporte" and "Cuerpo Nacional de Policía". The left sidebar has a vertical navigation menu with sections like "DNI electrónico", "Certificados Electrónicos", "Marco legal", and "PASAPORTE". The main content area shows a breadcrumb trail: "Inicio / Certificados Electrónicos / Qué son los Certificados Electrónicos". Below this, a section titled "Renovación Certificados" discusses the renewal of electronic certificates. A red box highlights the following text: "El titular puede proceder a renovar los certificados, si el estado de los mismos es uno de los siguientes:" followed by a list of three conditions. At the bottom, another red box contains the text: "EL PROCESO DE RENOVACIÓN DE CERTIFICADOS EN EL PUNTO DE ACTUALIZACIÓN DEL DNI 3.0 ES EL SIGUIENTE:" followed by a detailed description of the process.

DNI y Pasaporte

Cuerpo Nacional de Policía

DNI electrónico

Obtención del DNI

Cómo utilizar el DNI

Guía de referencia básica

Certificados Electrónicos

- »» Qué son los certificados electrónicos
- »» Renovación de Certificados
- »» Aceptación de los Certificados
- »» Autoridades de validación
- »» Política de certificación

Marco legal

Glosario

Atención al Ciudadano

Preguntas más frecuentes

Recursos

PASAPORTE

POLICÍA NACIONAL

sede electrónica

Idiomas | Inicio | Mapa web | Contacto

Inicio / Certificados Electrónicos / Qué son los Certificados Electrónicos

Renovación Certificados

Renovación de claves sin renovación del soporte físico (tarjeta):

La renovación de las claves es voluntaria, gratuita y por iniciativa del ciudadano.

En fechas próximas a la caducidad de sus certificados, recibirá, en la cuenta de correo electrónico que usted haya proporcionado en el momento de la expedición de su DNI, un aviso procedente de la dirección oficial notificaciones@policia.es, en el que le advierten de la próxima caducidad de sus certificados electrónicos.

El titular puede proceder a renovar los certificados, si el estado de los mismos es uno de los siguientes:

- Si fueron revocados a petición del ciudadano (solo podrá revocarse el certificado de firma digital).
- Por caducidad. Los certificados caducan pasados 60 meses desde la emisión de los mismos o si la fecha de caducidad del documento es inferior a esos 60 meses, limitación a la fecha de caducidad del mismo (mejora de notoriedad importante, puesto que la anterior regulación los limitaba a 30 meses y solo se podían renovar una vez caducados o dentro de los 30 días de la fecha de caducidad).
- Para proceder a la renovación deberá mediar la presencia física del titular en una Oficina de expedición. El ciudadano, haciendo uso de los Puntos de Actualización del DNI 3.0 habilitados en dichas oficinas y previa autenticación mediante la tarjeta y las plantillas biométricas (impresiones dactilares) capturadas durante la expedición de la Tarjeta, podrá desencadenar de forma desatendida el proceso de renovación de sus certificados.

EL PROCESO DE RENOVACIÓN DE CERTIFICADOS EN EL PUNTO DE ACTUALIZACIÓN DEL DNI 3.0 ES EL SIGUIENTE:

- El titular tras introducir correctamente el PIN, accede a la pantalla de "información sobre el contenido de su DNI 3.0", en la parte inferior puede visualizar el estado de sus certificados. En su caso, en la parte izquierda aparece una casilla "renovar certificados". Si se selecciona "renovar certificados" solicita nuevamente el PIN y posteriormente la presentación de la huella dactilar. Si el resultado es positivo se procede a la renovación de los certificados; este proceso dura aproximadamente 3 minutos. Es importante, no retirar el documento del lector de tarjetas hasta la finalización del proceso, porque el DNIe 3.0 podría quedar inservible. Si no fuere posible obtener la impresión dactilar de alguno de los dedos, el ciudadano deberá solicitar la renovación en un puesto de expedición atendido por un funcionario.

https://www.dnielectronico.es/PortalDNIe/PRF1_Cons02.action?pag=REF_1028&id_menu=%5B37%5D

Smart Card -- el caso del DNI-e

CUERPO NACIONAL DE POLICÍA
GOBIERNO MINISTERIO DE ESPAÑA DEL INTERIOR
DIRECCIÓN GENERAL DE LA POLICIA

DNI y Pasaporte Cuerpo Nacional de Policia

Policía Nacional

DNI electrónico

- Obtención del DNI
- Cómo utilizar el DNI
- Guía de referencia básica
- Certificados Electrónicos**
 - » Qué son los certificados electrónicos
 - » Renovación de Certificados
 - » Aceptación de los Certificados
 - » Autoridades de validación
 - » Política de certificación
- Marco legal
- Glosario
- Atención al Ciudadano
- Preguntas más frecuentes
- Recursos
- PASAPORTE**

POLICÍA NACIONAL

sede electrónica

Renovación Certificados

Renovación de claves sin renovación del soporte físico (tarjeta):

La renovación de las claves es voluntaria, gratuita y por iniciativa del ciudadano.

En fechas próximas a la caducidad de sus certificados, recibirá, en la cuenta de correo electrónico que figura en su DNI, un aviso procedente de la dirección oficial notificaciones@policiaciberseguridad.es con la información sobre la renovación de los certificados y la fecha de caducidad de sus certificados electrónicos.

El titular puede proceder a renovar los certificados, si el estado de los mismos es uno de los siguientes:

- Si fueron revocados a petición del ciudadano (solo podrá revocarse el certificado de firma digital)
- Por caducidad. Los certificados caducan pasados 60 meses desde la emisión de los mismos o si es inferior a esos 60 meses, limitación a la fecha de caducidad del mismo (mejora de notoriedad impidió que los limitaba a 30 meses y solo se podían renovar una vez caducados o dentro de los 30 días de la fecha de caducidad).
- Para proceder a la renovación deberá mediar la presencia física del titular en una Oficina de expedición. El ciudadano, haciendo uso de los Puntos de Actualización del DNIe 3.0 habilitados en dichas oficinas y previa autenticación mediante la tarjeta y las plantillas biométricas (impresiones dactilares) capturadas durante la expedición de la Tarjeta, podrá desencadenar de forma desatendida el proceso de renovación de sus certificados.

EL PROCESO DE RENOVACIÓN DE CERTIFICADOS EN EL PUNTO DE ACTUALIZACIÓN DEL DNI 3.0 ES EL SIGUIENTE:

- El titular tras introducir correctamente el PIN, accede a la pantalla de "información sobre el contenido de su DNI 3.0", en la parte inferior puede visualizar el estado de sus certificados. En su caso, en la parte izquierda aparece una casilla "renovar certificados". Si se selecciona "renovar certificados" solicita nuevamente el PIN y posteriormente la presentación de la huella dactilar. Si el resultado es positivo se procede a la renovación de los certificados; este proceso dura aproximadamente 3 minutos. Es importante, no retirar el documento del lector de tarjetas hasta la finalización del proceso, porque el DNIe 3.0 podría quedar inservible. Si no fuere posible obtener la impresión dactilar de alguno de los dedos, el ciudadano deberá solicitar la renovación en un puesto de expedición atendido por un funcionario.



- Cualquier operación criptográfica que requiera el uso de una de las claves privadas debe ser ejecutada en el interior del chip
- Las claves públicas se envían, tras su generación en el acto de expedición del DNI-e, a la CA para su inclusión en los correspondientes certificados digitales
 - una vez emitidos los certificados, estos se incorporan a la tarjeta para ser empleados en operaciones posteriores
 - los certificados digitales pueden ser leídos y extraídos para su proceso de forma externa al chip



- En el ámbito del DNI-e se usa **OCSP** para las revocaciones
 - cuando una aplicación requiere el estado actual de un certificado, envía una petición OCSP (mediante HTTP), a la URL del servicio de validación
 - una vez recibida la petición, el *OCSP Responder* accede a las CRLs, y averigua si dicho certificado se encuentra ahí incluido
- En la PKI adoptada para el DNI-e se ha optado por asignar las funciones de **Autoridad de Validación** a entidades diferentes de la **Autoridad de Certificación**
 - con el fin de aislar la comprobación de la vigencia de un certificado
 - existen tres **Autoridades de Validación**:
 - FNMT
 - Ministerio de Administraciones Públicas
 - Ministerio de Industria



- El marco legal básico del DNI-e es el siguiente:
 - Directiva 1999/93/CE del Parlamento Europeo y del Consejo, de 13 de diciembre, por la que se establece un marco comunitario para la firma electrónica
 - Ley 59/2003, de 19 de diciembre, de Firma Electrónica
 - Ley Orgánica 15/1999, de 13 de diciembre, de Protección de los Datos
 - Real Decreto 1553/2005, de 23 de diciembre, por el que se regula documento nacional de identidad y sus certificados de firma electrónica
 - Real Decreto 1720/2007, de 21 de diciembre, relacionado con la protección de datos de carácter personal
 - Real Decreto 1586/2009, de 16 de octubre, Real Decreto 869/2013, de 8 de noviembre, y Real Decreto 414/2015, de 29 de mayo, por los que se modifica el Real Decreto 1553/2005

SOCIEDAD

CASTELLERS CIENCIA MEDIO AMBIENTE TIEMPO SANIDAD SUCESOS PRIMERA PLAN@ +PERSONAS

Desactivada la firma digital de los DNI electrónicos por un fallo de seguridad

La medida, que afecta a los expedidos desde abril del 2015, viene por un fallo en el chip del fabricante

El problema está en un protocolo de transacciones digitales que utilizan millones de máquinas

Carmen Jané

Barcelona - Jueves, 09/11/2017 | Actualizado el 10/11/2017 a las 18:00 CET



Un lector de DNI electrónico. / PERIODICO

¡SOLO 5 DÍAS!
Hasta el 24 de noviembre

Préstamo
NARANJA
desde
4,95% TIN
(5,06% TAE)*

<http://www.elperiodico.com/es/sociedad/20171109/desactivada-la-firma-digital-de-los-dni-2015-possibles-fallos-de-seguridad-6412261>