Resumen Teórica 6 : Criptografía (parte 3)

Tomás F. Melli

September 2025

Índice

1 Key Exchange usando criptografía de clave pública

El problema gira alrededor de la siguiente pregunta : *cómo acuerdan Alice y Bob una clave de sesión ?.*

- Cada participante tiene un par de claves:
  - Alice: clave pública e_A, clave privada d_A.
  - Bob: clave pública e_B, clave privada d_B.
- Las claves públicas son conocidas por todos.
- Las claves privadas son secretas y sólo las conoce su dueño.

Solución versión 1 (básica)

- Alice genera k_s (clave de sesión aleatoria).
- Envía $\{k_s\}_{e_B}$.
- Bob descifra con d_B y obtiene k_s.

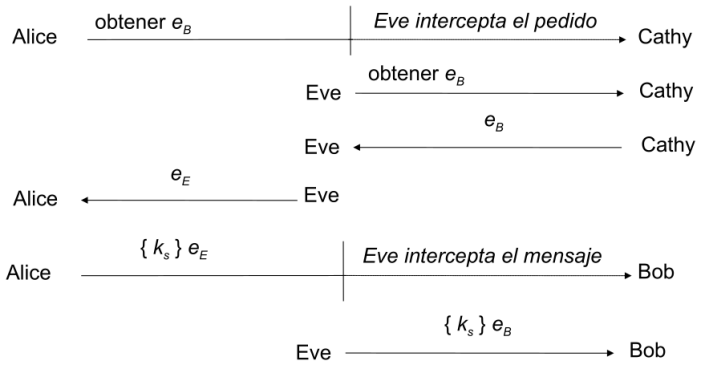
Problema: como e_B es pública, Bob no puede verificar que el mensaje vino de Alice.

Solución versión 2 (con autenticación)

- Alice envía $\{\{k_s\}_{d_A}\}_{e_B}$.
- Bob descifra con d_B y verifica la firma con e_A.
- Ahora Bob sabe que la clave proviene de Alice.

Supuesto: Bob posee e_A y Alice posee e_B. Si no, deben obtenerla de un tercero confiable (Cathy).
Problema: si la clave no está asociada a la identidad de su dueño, el sistema es vulnerable a un ataque de *Man-in-the-Middle*.

Ataque Man-in-the-Middle



1. Alice pide e_B a Cathy.
2. Eve intercepta y entrega e_E en lugar de e_B.
3. Alice cifra $\{k_s\}_{e_E}$ creyendo que es la clave de Bob.
4. Eve descifra con d_E, obtiene k_s, y reenvía $\{k_s\}_{e_B}$ a Bob.
5. Bob cree que recibió la clave de Alice, pero Eve ya la conoce.

1.1 Infraestructura de claves

Objetivo: asociar una clave pública con la identidad de su poseedor.

- Criptografía simétrica: no es posible (la clave es compartida).
- Criptografía asimétrica: sí es posible (la clave pública se vincula al dueño).

1.2 Certificados digitales

Estructura de datos firmada por una Autoridad de Certificación (CA).

- Contiene:
  - Identidad del poseedor.
  - Clave pública.
  - Fecha de emisión.
  - Información adicional (ej: emisor).
- Firmado por la CA con su clave privada:

$$CA = \{e_A \mid\mid Alice \mid\mid \ldots\}_{d_C}$$

1.2.1 Utilización de certificados

- Bob obtiene el certificado de Alice.
- Si Bob conoce e_C (clave pública de Cathy), puede:
  - Validar que el certificado pertenece a Alice.
  - Obtener la clave pública e_A.

Problema: ahora Bob necesita confiar en e_C. Esto lleva a la necesidad de **cadenas de firmas**.

2 Cadenas de firmas

Una cadena de firmas (o chain of trust) es la forma en que se resuelve el problema de *¿y cómo sé que la clave pública del certificador también es confiable?*. Una cadena de firmas es un mecanismo para que la confianza en una clave pública pueda ser "heredada" desde una entidad raíz confiable hacia otras entidades intermedias y finalmente hacia el certificado de la persona/servidor con el que quiero comunicarme.

2.1 PGP (Pretty Good Privacy)

2.1.1 Historia

- 1991: Philip Zimmermann publica la primera versión (1.0) de PGP.
- 1992: aparece la versión 2.0.
  - El código se escribe fuera de Estados Unidos para evitar leyes restrictivas sobre software criptográfico.
  - Zimmermann enfrenta problemas legales por estas restricciones.

2.1.2 Gestión de claves

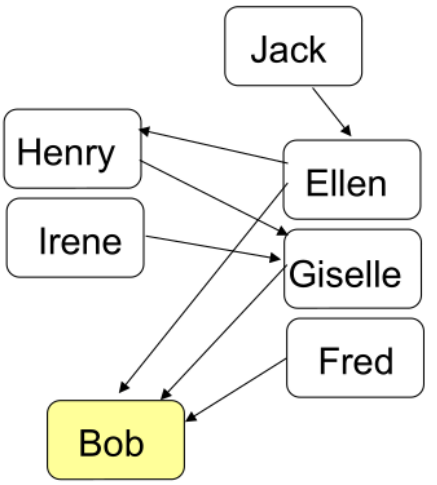
- Basada en la **confianza mutua**.
- Adecuada para **entornos privados** o **intranets**.
- Versiones recientes:
  - Incorporan la utilización de Autoridades Certificantes como certificadores de claves públicas.
- Varias implementaciones open-source. Ejemplo: **GnuPG (GPG)**.

2.1.3 Datos asociados a las claves

- Versión de PGP.
- Clave pública junto con el algoritmo (RSA, DSA, DH).
- Identidad del titular de la clave.
- Firma digital del titular (auto-firma).
- Periodo de validez.
- Algoritmo simétrico de cifrado preferido.
- Conjunto de firmas de terceros (opcional):
  - Definen nivel de confianza.
  - Definen nivel de validez.

2.1.4 Validación de claves

Ejemplo de cómo Alice valida la clave pública de Bob:



Las flechas indican firmas.
No se muestran las auto-firmas.

1. Alice necesita validar la clave de Bob.
   - No conoce a Fred, Giselle ni Ellen directamente.
2. Alice obtiene la clave de Giselle.
   - Conoce poco a Henry, pero su nivel de confianza no le resulta suficiente.
3. Alice obtiene la clave de Ellen.
   - Conoce a Jack (su esposo), confía en él.
   - Usa la clave de Jack para validar la de Ellen.
   - Luego, con la clave validada de Ellen, valida la de Bob.

2.1.5 Conclusión

PGP usa una red de confianza distribuida (web of trust) que permite que la confianza en una clave pública se transfiera desde una autoridad raíz confiable hacia entidades intermedias y, finalmente, al certificado que quiero validar.

2.2 Certificados X.509 v3

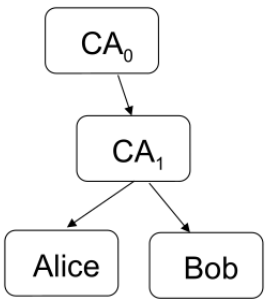
Es un certificado digital estandarizado que se utiliza para asociar una clave pública con la identidad de su dueño de manera confiable. Forma parte de una infraestructura de clave pública (PKI) (lo vemos después). Su objetivo principal es garantizar la autenticidad de la clave y permitir la comunicación segura entre usuarios o sistemas.

- Fueron creados como parte de los mecanismos de control de acceso de los Directorios X.500.
- Actualmente son la tecnología predominante en certificados digitales.

2.2.1 Campos de los certificados

- Versión de X.509 (ej. v3).
- Número de serie.
- Algoritmo de firma (ej. `sha-256withRSAEncryption`).
- Nombre del emisor.
- Periodo de validez.
- Nombre del titular.
- Clave pública del titular.
- Firma: hash del certificado cifrado con la clave privada del emisor.
- Extensiones (opcional).

2.2.2 Validación



1. Obtener la clave pública del emisor.
2. Descifrar la firma para obtener el hash del certificado.
3. Recalcular el hash del certificado y compararlo con el obtenido en el paso anterior.
4. Verificar el periodo de validez del certificado.
5. Repetir el proceso hacia arriba en la jerarquía de certificación hasta llegar a una CA raíz confiable.

2.2.3 Emisores

- Una **Autoridad Certificante (CA)** es la entidad que emite certificados.
- Problema: ¿qué ocurre cuando existen múltiples CAs?
  - Cada sistema (ej. navegador, SO) mantiene un almacén de CAs raíz confiables.
  - La confianza se transmite desde la CA raíz a las intermedias y finalmente al certificado del titular.

2.2.4 Almacenamiento de claves privadas

Es necesario proteger adecuadamente las claves privadas. Algunas medidas:

- Cifrar el archivo que contiene la clave.
  - Riesgo: un atacante puede registrar la contraseña o leer la clave en memoria.
- Utilizar dispositivos específicos (smartcards, tokens criptográficos, HSMs).
  - La clave nunca abandona el dispositivo.
  - Puede particionarse la clave en más de un dispositivo para reducir el riesgo en caso de robo.

3 FIPS 140: Estándar de Seguridad para Módulos Criptográficos

FIPS 140 es un estándar del NIST (*National Institute of Standards and Technology*, EE.UU.) que define los **requisitos de seguridad para módulos criptográficos**, ya sean de hardware o software. Su objetivo es garantizar que los módulos utilizados en sistemas criptográficos cumplan criterios mínimos de seguridad.

3.1 Versiones de FIPS 140

- **FIPS 140-1 (1994)**: versión inicial, establecida para evaluar la seguridad de los módulos criptográficos.
- **FIPS 140-2 (2002)**: actualización que adapta el estándar a los avances tecnológicos.
- **FIPS 140-3 (2019)**: versión actual, alineada con los estándares internacionales **ISO/IEC 19790** y **ISO/IEC 24759**, facilitando la certificación de proveedores a nivel global.

Nota: Este estándar no evalúa sistemas completos ni infraestructuras de PKI, sino exclusivamente los **módulos criptográficos**.
Referencias oficiales:

- https://csrc.nist.gov/pubs/fips/140-3/final
- https://csrc.nist.gov/projects/cryptographic-module-validation-program/validated-modules/search/all

3.2 Niveles de seguridad en FIPS 140-3

FIPS 140-3 define **cuatro niveles de seguridad**, que aumentan progresivamente los requisitos de seguridad física y lógica:

Nivel 1: Seguridad Básica

- Utiliza **algoritmos de cifrado aprobados**.
- No hay exigencias especiales de seguridad física más allá de la fabricación de los componentes.

Nivel 2: Evidencia de Manipulación (Tamper-Evident)

- Añade requisitos de **seguridad física**, como sellos de evidencia de manipulación o cubiertas opacas.
- Requiere **autenticación basada en roles** para los operadores que acceden al módulo.

Nivel 3: Resistencia a la Manipulación (Tamper-Resistant)

- Mejora la seguridad física del Nivel 2, dificultando el acceso no autorizado a los componentes internos.
- Exige **autenticación basada en identidad**.
- Los **parámetros críticos de seguridad (CSPs)** deben introducirse o extraerse de manera cifrada.
- Protección de claves: si se detecta un intento de intrusión, el módulo debe ser capaz de **borrar automáticamente los CSPs**.

Nivel 4: Seguridad Física Avanzada

- Nivel más alto de seguridad definido por el estándar.
- Requiere protección física avanzada que pueda **detectar y responder a intrusiones en tiempo real**.
- Protege contra fluctuaciones de voltaje y temperatura que puedan comprometer la seguridad.
- Exige **autenticación multifactor** para el acceso a funciones críticas.

4 Revocación de Claves y Certificados

En los sistemas de infraestructura de clave pública (PKI), los **certificados digitales** pueden **invalidarse antes de su fecha de expiración** por diversas razones:

- **Compromiso de la clave:** cuando se sospecha que la clave privada asociada al certificado ha sido divulgada o robada.
- **Cambio de situación del titular:** por ejemplo, cuando un empleado cambia de cargo, deja la organización, o ya no debería mantener ciertos privilegios.

4.1 Problemas asociados a la revocación

La revocación de certificados introduce algunos desafíos importantes:

1. **Autorización:** solo la entidad que emite el certificado (la Autoridad Certificante, AC) está autorizada para revocarlo.
2. **Disponibilidad de información:** la información sobre certificados revocados debe estar disponible de manera rápida y confiable para que los usuarios puedan verificar la validez de un certificado antes de confiar en él.

4.2 Revocación en certificados X.509 v3

En el estándar X.509 v3, la revocación de certificados se gestiona mediante las **CRL (*Certificate Revocation List*)**:

- La CRL es una lista de certificados que han sido revocados antes de su fecha de expiración.
- Se puede comparar con las listas de tarjetas de crédito robadas: los certificados en la lista ya no son confiables.
- Solo la **AC que emitió el certificado** puede revocarlo.
- Para informar a los usuarios, la AC agrega el certificado revocado a la CRL y la publica de forma accesible para todos los sistemas que confían en dicha AC.

Resumen

La revocación permite mantener la confianza en una infraestructura de clave pública, asegurando que certificados comprometidos o caducados no sean usados de manera indebida.

5 Tiempo de Vida de las Claves

En criptografía, las **claves** pueden clasificarse según su **tiempo de vida**, es decir, el período durante el cual una clave se considera válida y segura para su uso. Esta clasificación permite gestionar la seguridad de manera más efectiva, reduciendo riesgos asociados al uso prolongado o indebido de claves criptográficas.

5.1 Claves de Corto Plazo

- Son **generadas automáticamente** por el sistema.
- Su uso está limitado a **un solo mensaje o sesión** criptográfica.
- Una vez utilizada, la clave se **descarta inmediatamente**, reduciendo la posibilidad de comprometer la información en caso de interceptación.
- Este tipo de claves se emplea principalmente en **sesiones efímeras**, como en protocolos de cifrado de flujo o comunicación temporal segura.

5.2 Claves de Largo Plazo

- Son **generadas explícitamente por el usuario** y permanecen activas durante un período más prolongado.
- Se utilizan para dos propósitos principales:
  1. **Autenticación:** verificar la identidad de los usuarios o dispositivos.
  2. **Confidencialidad (cifrado):** proteger la información que se mantiene durante un tiempo más largo.
- Las claves de largo plazo requieren medidas adicionales de protección, como almacenamiento seguro y control de acceso, debido a que su compromiso tendría un impacto más significativo.

Resumen

El **tiempo de vida de una clave** determina cómo se genera, cómo se utiliza y qué medidas de seguridad adicionales son necesarias. Las claves de corto plazo ofrecen mayor seguridad para comunicaciones efímeras, mientras que las claves de largo plazo facilitan autenticación y cifrado sostenido.

6 Manejo de Claves

El **manejo de claves** o *key management* es un conjunto de prácticas, procedimientos y políticas destinadas a **asegurar que las claves criptográficas se generen, distribuyan, almacenen, utilicen y destruyan de manera segura** durante todo su ciclo de vida. Una correcta gestión de claves es fundamental para mantener la **confidencialidad, integridad y autenticidad** de la información en cualquier sistema criptográfico.

6.1 Aspectos Clave en el Manejo de Claves

Al manejar claves criptográficas, se deben considerar los siguientes aspectos:

1. Generación de claves

- Las claves deben generarse siguiendo procedimientos seguros que garanticen suficiente entropía y resistencia a ataques de fuerza bruta.
- Dependiendo del uso, se puede optar por claves de **corto plazo** (efímeras) o de **largo plazo** (persistentes).

2. Asociación de la clave a la identidad del poseedor

- Cada clave debe estar correctamente vinculada a la identidad de su propietario para asegurar la confianza en transacciones y comunicaciones.
- Esto se logra mediante certificados digitales emitidos por una autoridad certificante (AC) o mediante mecanismos de confianza mutua en entornos privados.

3. Distribución de claves

- Las claves deben transmitirse de manera segura, evitando que terceros puedan interceptarlas.
- Esto incluye el intercambio seguro de claves simétricas y la publicación controlada de claves públicas.

4. Establecimiento de claves compartidas

- Cuando dos partes necesitan una clave común, se utilizan protocolos de intercambio de claves (por ejemplo, Diffie-Hellman o RSA) que garantizan que solo los participantes autorizados puedan obtener la clave.

5. Almacenamiento seguro de claves

- Las claves deben protegerse en repositorios seguros o dispositivos criptográficos (smartcards, HSM, tokens) para prevenir accesos no autorizados.
- Se deben implementar controles de acceso, cifrado en reposo y medidas de respaldo confiables.

6. Gestión de compromisos

- Si una clave se ve comprometida, se deben tomar medidas inmediatas: revocación de certificados, reemplazo de claves y notificación a las partes afectadas.

7. Destrucción segura de claves

- Cuando una clave ya no es necesaria o ha caducado, debe destruirse de manera segura para prevenir su recuperación por terceros.
- Esto incluye sobrescribir la clave en memoria o almacenamiento físico, o eliminar de forma irreversible los archivos y dispositivos que la contengan.

Resumen

El manejo de claves abarca todo el ciclo de vida de las claves criptográficas: desde su **generación** hasta su **destrucción segura**, pasando por la **asociación con identidades**, la **distribución segura** y la **respuesta ante compromisos**. Una gestión adecuada garantiza la seguridad de la información y la confianza en los sistemas criptográficos.

PKCS – Public-Key Cryptography Standards

Los **PKCS** son un conjunto de especificaciones técnicas desarrolladas por **Netscape, RSA y otros**, cuyo objetivo es **uniformizar las técnicas y protocolos de criptografía de clave pública (*)**. La primera versión se publicó en 1991, y forman parte de estándares más amplios como **ANSI PKIX, X9, SET, S/MIME y SSL**.

Documentos principales

- **PKCS #1:** RSA Cryptography Standard – define cómo usar RSA para cifrado y firma digital.
- **PKCS #2:** Incluido ahora en PKCS #1 – antes cubría operaciones aritméticas de RSA.
- **PKCS #3:** Diffie-Hellman Key Agreement Standard – especifica el protocolo de intercambio de claves Diffie-Hellman.
- **PKCS #4:** Incluido ahora en PKCS #1 – soporte antiguo para RSA, ya obsoleto.
- **PKCS #5:** Password-Based Cryptography Standard – define métodos para derivar claves a partir de contraseñas.
- **PKCS #6:** Extended-Certificate Syntax Standard – proporciona un formato extendido para certificados (ya obsoleto, reemplazado por PKCS #7).
- **PKCS #7:** Cryptographic Message Syntax Standard – define cómo empaquetar datos cifrados y firmados digitalmente.
- **PKCS #8:** Private-Key Information Syntax Standard – especifica cómo almacenar y transportar claves privadas.
- **PKCS #9:** Selected Attribute Types – define atributos adicionales que se pueden asociar con certificados y solicitudes de certificados.
- **PKCS #10:** Certification Request Syntax Standard – define cómo un usuario solicita un certificado a una CA.
- **PKCS #11:** Cryptographic Token Interface Standard – especifica la interfaz de software para tokens criptográficos (smartcards, HSM).
- **PKCS #12:** Personal Information Exchange Syntax Standard – define cómo almacenar y transportar claves y certificados personales de manera segura.
- **PKCS #13:** Elliptic Curve Cryptography Standard – define el uso de curvas elípticas para criptografía de clave pública.
- **PKCS #15:** Cryptographic Token Information Format Standard – especifica cómo organizar información criptográfica en tokens y tarjetas inteligentes.

Estos estándares permiten **compatibilidad y seguridad** en la implementación de criptografía de clave pública, facilitando la interoperabilidad entre sistemas y aplicaciones. **Referencia:** http://www.rsasecurity.com/rsalabs/pkcs/

7 PKI – Public Key Infrastructure

La PKI (Public Key Infrastructure) es una combinación de hardware y software, políticas y procedimientos que permiten asegurar la identidad de los participantes en un intercambio de datos usando criptografía de clave pública. Los componentes más comunes son:

- **Autoridad certificante (CA):** Tercera parte confiable que da fe de la veracidad de la información incluida en los certificados que emite. Emite certificados digitales según su *Certificate Policy (CP)*, que define reglas de aplicabilidad de un certificado digital a una comunidad y/o a una clase de aplicaciones con requerimientos de seguridad en común; y perfiles de certificados.
- **Manual de procedimientos de la CA (CPS):** Declaración de prácticas para emitir, administrar, revocar y renovar certificados. Referencias útiles: RFC 3647, https://cabforum.org/.
- **Autoridad de registro (RA):** Entidad responsable de identificar y autenticar a los suscriptores de certificados, aprobar o rechazar solicitudes, iniciar revocaciones, procesar renovaciones y aprobar o rechazar las solicitudes de renovación de certificados.
- **Terceros usuarios y suscriptores:**
  - **Terceros usuarios**: receptores de un certificado que actúan basados en el mismo o en cualquier firma digital que se verifique con ese certificado.
  - **Suscriptor**: sujeto que solicita la emisión de un certificado.
- **Repositorios:** Estructuras encargadas de almacenar información relativa a la PKI, principalmente el repositorio de certificados y el de listas de certificados revocados.

7.1 Solicitud de certificados

1. El suscriptor genera un par de claves y firma la clave pública junto con su información personal con su clave privada. Luego envía todo a la CA.
   - Prueba que posee la clave privada correspondiente.
   - Protege la información enviada a la CA.
2. La CA verifica la firma del suscriptor en los datos recibidos y opcionalmente puede verificar la información por otros medios (presencia física, correo electrónico, legajo de personal, etc.). En este paso interviene la RA.
3. La CA firma la clave pública y parte de la información enviada por el suscriptor para crear el certificado, asociando así al suscriptor con su clave pública y datos.
4. El suscriptor recibe el certificado y verifica la firma de la CA y los datos del certificado para asegurar que no fueron modificados y proteger la información.
5. La CA publica el certificado.

7.2 Confianza y modelos

Para confiar en certificados digitales es necesario responder:

- ¿Cómo se determina en qué certificados se puede confiar?
- ¿Cómo se establece la confianza?
- ¿Bajo qué circunstancias la confianza puede ser limitada o controlada?

Los modelos de confianza en certificados X.509v3 definen cómo un sistema decide si un certificado digital es auténtico y válido. Existen varios modelos:

- **Jerárquico:** La confianza se basa en una estructura en árbol, donde una autoridad certificante raíz (Root CA) es completamente confiable y todas las demás CAs subordinadas heredan esa confianza. Común en PKI corporativas.
- **Modelo web:** La confianza se establece mediante la aceptación directa de certificados o CAs por el usuario o la aplicación, como los navegadores web que incluyen un conjunto de CAs confiables.

- **Bridge CA:** Conecta diferentes PKI jerárquicas mediante una CA puente (Bridge CA), permitiendo que varias jerarquías confíen entre sí sin unificar toda la estructura.
- **Certificación cruzada:** Dos CAs se reconocen mutuamente como confiables mediante la emisión de certificados entre ellas, estableciendo confianza bilateral entre distintas jerarquías.
- **Reconocimiento cruzado:** Similar a la certificación cruzada, se usa cuando múltiples organizaciones acuerdan explícitamente reconocer las CAs de las otras como confiables, sin fusionar jerarquías.
- **CTL (Certificate Trust List):** Listas predefinidas de certificados o CAs confiables que un sistema utiliza para decidir si puede confiar en un certificado. Común en sistemas Windows.

7.3 Tipos de certificados

Según el uso, se encuentran:

- **Certificados SSL:** Se utilizan para asegurar la comunicación entre servidores web y clientes mediante HTTPS.
- **Certificados S/MIME (correo electrónico):** Permiten cifrar y firmar digitalmente correos electrónicos para garantizar confidencialidad e integridad.
- **Certificados S/MIME (personales):** Similar al anterior, pero orientados a usuarios individuales para autenticación y firma de correos.
- **Certificados para la firma de código:** Garantizan la autenticidad e integridad de software y aplicaciones distribuidas.
- **Certificados para AC (Autoridad Certificante):** Emitidos a CAs para permitirles emitir certificados a otros usuarios o entidades.
- **Certificados para WPA-SPK:** Se usan en redes inalámbricas para autenticar dispositivos mediante claves precompartidas (Wi-Fi Protected Access).
- **Certificados para VPN:** Permiten la autenticación y el establecimiento seguro de conexiones de red privada virtual.
- **Otros:** Existen certificados especializados para distintos fines según las necesidades de seguridad.

7.4 Certificados digitales X.509 v3

Los certificados X.509 v3 contienen varios campos esenciales que permiten identificar, validar y utilizar la clave pública asociada al suscriptor:

- **version:** Indica la versión del estándar X.509 utilizada; para los certificados modernos suele ser v3.
- **serialNumber:** Identificador único del certificado asignado por la Autoridad Certificante (AC) para distinguirlo de otros certificados emitidos por la misma AC.
- **signatureAlgorithm:** Especifica el algoritmo de firma digital usado por la AC para firmar el certificado, asegurando su autenticidad e integridad.
- **issuer:** Nombre de la Autoridad Certificante que emite el certificado; sirve para verificar la validez de la firma.
- **validity:** Define el período de vigencia del certificado, con fecha de inicio (*notBefore*) y fecha de expiración (*notAfter*).
- **subject:** Nombre del suscriptor o entidad a la que pertenece el certificado; identifica quién es el propietario de la clave pública.
- **subjectPublicKeyInfo:** Contiene la clave pública del suscriptor y el algoritmo asociado; permite verificar firmas o cifrar información dirigida al titular del certificado.
- **signature:** Hash del certificado cifrado con la clave privada de la AC; garantiza la autenticidad e integridad del certificado.
- **extensions (opcional):** Campos adicionales que permiten asociar atributos al certificado, como restricciones de uso, identificadores alternativos, políticas de certificación, distribución de CRL, entre otros.

Identificación: Cada certificado se identifica inequívocamente mediante `serialNumber + issuer`.

7.5 Extensiones

- Permiten asociar atributos al certificado digital.
- Se utilizan para manejar la herencia de certificación, restringir el uso y aportar precisión.

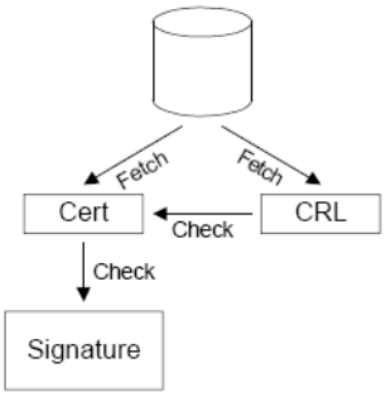
Algunos ejemplos son :

- **Authority Key Identifier / Subject Key Identifier:** Identificadores de clave que vinculan el certificado con la clave pública de la AC emisora o del propio sujeto, facilitando la validación de la cadena de certificados.
- **Key Usage:** Define los propósitos autorizados de la clave del certificado, como cifrado, firma digital, autenticación, etc.
- **Extended Key Usage:** Permite especificar usos más concretos, por ejemplo, SSL/TLS, correo seguro (S/MIME) o firma de código.
- **Certificate Policies:** Indican las políticas de certificación aplicables, incluyendo la confiabilidad del certificado y los requisitos para su uso.
- **Authority Alternative Name / Subject Alternative Name:** Permite incluir nombres adicionales del emisor o del sujeto, como direcciones de correo, DNS, IP u otros identificadores.
- **Basic Constraints:** Indica si un certificado puede actuar como Autoridad Certificante y limita la profundidad de la cadena de certificados subordinados.
- **CRL Distribution Points:** Especifica las ubicaciones donde se pueden consultar las listas de revocación de certificados (CRLs) emitidas por la AC.
- **Authority Information Access:** Proporciona información sobre cómo obtener certificados de la AC emisora o verificar el estado de revocación de un certificado mediante protocolos como OCSP.

7.6 CRLs – Certificate Revocation Lists

Una **CRL (Certificate Revocation List)** constituye un mecanismo utilizado para verificar si un certificado digital sigue siendo válido o ha sido revocado antes de su fecha de expiración. Estas listas son publicadas de manera continua por la Autoridad Certificante (AC) que emitió los certificados, y contienen información actualizada sobre los certificados revocados, incluyendo su número de serie y la fecha de revocación.
Cada CRL tiene un período de validez determinado, tras el cual debe ser actualizada para reflejar el estado vigente de los certificados emitidos. Este proceso garantiza que los usuarios y sistemas puedan confiar en la autenticidad de los certificados mediante la comprobación de su estado antes de realizar cualquier operación criptográfica segura.

Esquema



7.6.1 Campos principales

- **version:** Versión de la estructura, generalmente v2.
- **signature:** Algoritmo de firma utilizado por la AC para firmar la CRL.
- **issuer:** Nombre del emisor de la CRL (la Autoridad Certificante).

- **thisUpdate / nextUpdate:** Fecha de emisión de la CRL y fecha de próxima actualización.
- **revokedCertificates:** Lista de certificados revocados, incluyendo:
  - **userCertificate:** Número de serie del certificado revocado.
  - **revocationDate:** Fecha de revocación del certificado.
  - **crlEntryExtensions:** Extensiones específicas de cada entrada de certificado revocado.
- **crlExtensions:** Extensiones generales de la CRL.

7.6.2 Extensiones de la CRL

- **Authority Key Identifier (AKI):** Permite identificar de manera única la clave pública de la Autoridad Certificante que emitió la CRL. Facilita la validación de la cadena de confianza y asegura que la CRL fue emitida por la AC correcta.
- **CRL Number / Delta CRL Indicator / Delta CRL Distribution Point (Delta CRL DP):**
  - **CRL Number:** Número secuencial de la CRL, usado para diferenciar entre diferentes emisiones de CRL.
  - **Delta CRL Indicator:** Indica que la CRL contiene solo cambios respecto a la última CRL completa (certificados recién revocados).
  - **Delta CRL Distribution Point (DP):** Especifica dónde se pueden obtener las CRL completas correspondientes a la delta CRL.
- **Issuing Distribution Point:** Define el ámbito de aplicación de la CRL, indicando qué certificados específicos están cubiertos (por ejemplo, certificados de usuario o de AC subordinada).

Extensiones de una entrada individual de la CRL

- **Reason Code:** Indica el motivo por el cual un certificado fue revocado, como compromiso de clave, cambio de afiliación o cesación de la autoridad del certificado.
- **Hole Instruction Code:** Utilizado en escenarios donde existen huecos en la numeración de certificados o en CRLs, proporcionando instrucciones sobre cómo manejar estas irregularidades.

7.6.3 Problemas

- **No contienen el estado actual de los certificados:** Una CRL solo refleja el estado de revocación hasta el momento de su emisión. Si un certificado se revoca inmediatamente después de que se publicó la CRL, esta información no estará reflejada hasta la próxima actualización, lo que genera un desfase temporal en la validez de los certificados.
- **La responsabilidad por la verificación recae en el usuario:** Cada sistema o aplicación que use certificados debe descargar y verificar manualmente la CRL para asegurarse de que los certificados con los que interactúa no hayan sido revocados. Esto implica que el control de la validez recae en el usuario o sistema final, aumentando el riesgo de errores o descuidos.
- **Problemas de volumen y de distribución:** Las CRL pueden crecer mucho si la autoridad certificante administra una gran cantidad de certificados. Descargar listas completas puede ser costoso en términos de ancho de banda y almacenamiento, y distribuirlas eficientemente a todos los usuarios puede ser un desafío técnico, especialmente en sistemas con muchos clientes.

7.6.4 Soluciones

- **División del alcance de la CRL:** Al segmentar la CRL según criterios específicos (por ejemplo, por tipo de certificado, por área geográfica o por entidad emisora), se reduce la cantidad de certificados incluidos en cada lista. Esto hace que las CRL sean más manejables y rápidas de descargar y procesar.
- **Delta CRL:** Una delta CRL contiene únicamente los certificados que fueron revocados desde la última CRL completa. Esto disminuye considerablemente el tamaño de los datos que los usuarios necesitan descargar y actualiza la información de manera más frecuente y eficiente.
- **CRL indirectas:** Son CRL emitidas por entidades distintas a la autoridad certificante principal, utilizando claves diferentes. Esto permite distribuir la carga de verificación y actualización de revocaciones, mejorando la eficiencia y reduciendo el riesgo de congestión en la AC principal.

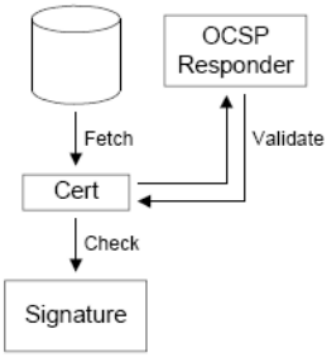
7.7 OCSP – Online Certificate Status Protocol

El **OCSP** es un protocolo que permite a los usuarios verificar en tiempo real el estado de validez de un certificado digital, sin necesidad de descargar una lista completa de certificados revocados (CRL). Para funcionar, el servicio requiere que la **Autoridad Certificante (AC)** responda a las consultas de los clientes.
Cada respuesta de OCSP está **firmada digitalmente** por la AC, garantizando la autenticidad de la información. El protocolo devuelve uno de tres posibles estados para un certificado:

- **Good:** el certificado es válido y no ha sido revocado.
- **Revoked:** el certificado ha sido revocado por la AC y ya no es confiable.
- **Unknown:** la AC no puede determinar el estado del certificado, por ejemplo porque no tiene registro del mismo.

OCSP representa una solución más eficiente y dinámica que las CRL tradicionales, ya que proporciona información actualizada de manera inmediata y reduce la necesidad de manejar grandes listas de revocaciones.

Esquema



Referencias

Para obtener información detallada sobre la infraestructura de clave pública y la validación de certificados digitales, se recomienda consultar las siguientes normas técnicas:

- **RFC 5280** (*Internet X.509 Public Key Infrastructure Certificate and CRL Profile*): define el perfil estándar de los certificados X.509 y las listas de revocación de certificados (CRL). https://www.rfc-editor.org/rfc/rfc5280.
- **RFC 6960** (*X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*): especifica el protocolo OCSP, que permite verificar en línea el estado de validez de un certificado digital.

7.8 CAA – Certification Authority Authorization

El **CAA (Certification Authority Authorization)** es un registro de recursos del DNS que permite a los propietarios de un dominio especificar qué Autoridades Certificantes (CAs) están autorizadas para emitir certificados digitales para ese dominio.

Esta medida aumenta la seguridad al prevenir la emisión no autorizada de certificados, reduciendo riesgos de ataques como el *man-in-the-middle* o la emisión fraudulenta de certificados.

Desde el 8 de septiembre de 2017, las Autoridades Certificantes están obligadas a consultar los registros CAA antes de emitir cualquier certificado para un dominio, asegurando así que solo las CAs autorizadas puedan hacerlo.

Referencia: RFC 6844 – DNS Certification Authority Authorization Resource Record (CAA)

7.9 Certificate Pinning

Certificate Pinning es una técnica de seguridad que tenía como objetivo detectar certificados maliciosos, incluso si estos respetaban la cadena de confianza habitual. Su propósito era prevenir ataques de intermediario (*man-in-the-middle*) asegurando que un cliente (por ejemplo, un navegador web) solo confiara en certificados específicos previamente conocidos para un dominio determinado.

En el contexto de HTTP, esta técnica se implementó mediante **HPKP (HTTP Public Key Pinning)**, que permitía a los servidores indicar a los navegadores qué claves públicas o certificados estaban autorizados para un sitio. Sin embargo,

esta práctica fue declarada *deprecated* debido a problemas de operatividad y riesgo de errores que podían bloquear el acceso legítimo a los sitios web.

Referencias:

- RFC 7469 – Public Key Pinning Extension for HTTP
- Artículo: Google declara obsoleto HTTP Public Key Pinning (HPKP)

7.10 Certificate Transparency

Certificate Transparency (CT) es un proyecto cuyo objetivo es proteger el proceso de emisión de certificados digitales mediante la creación de un marco abierto para supervisar y auditar los certificados de HTTPS. Este enfoque busca aumentar la confianza en la infraestructura de clave pública (PKI) al permitir que cualquier persona pueda verificar públicamente los certificados emitidos por las Autoridades Certificantes (CA).

El proyecto recomienda que todas las CA registren los certificados que emiten en *logs* públicamente verificables, inmutables y que solo permitan la adición de nuevas entradas, garantizando la trazabilidad y la transparencia del proceso. De esta manera, se previenen emisiones no autorizadas y se facilita la detección de certificados maliciosos o erróneos. En el futuro, es posible que los navegadores rechacen certificados que no hayan sido registrados en estos *logs* de CT.

Referencias:

- RFC 6962 – Certificate Transparency
- Información oficial: What is Certificate Transparency?
- Artículo explicativo: Certificate Transparency: El qué y el cómo

7.11 El eslabón más débil

Aunque la infraestructura de clave pública (PKI) está diseñada para garantizar seguridad y autenticidad, ciertos factores prácticos y errores de implementación pueden convertirse en el "eslabón más débil" de la cadena de confianza.

Null Byte en certificados: X.509 utiliza ASN.1 para representar cadenas de texto, similar a la convención de Pascal, que incluye un indicador de fin de cadena. Muchas aplicaciones están programadas en C, lo que puede generar vulnerabilidades si no manejan correctamente estos indicadores. Por ejemplo, si un atacante solicita un certificado para `www.facebook.com\0.midomino.com`, algunas aplicaciones podrían interpretar la cadena hasta el carácter nulo y aceptar erróneamente un dominio malicioso como válido. Esta falla explota la manera en que los strings son interpretados en la memoria y puede permitir suplantación de identidad.

Manipulación de OCSP: El protocolo OCSP (Online Certificate Status Protocol) permite verificar el estado de un certificado en línea. Sin embargo, en algunas implementaciones, la respuesta no siempre está firmada, y un atacante que realice un ataque Man-in-the-Middle (MiTM) podría modificar el valor de estado a "3" (try later), provocando que la verificación falle y deje al sistema operar de manera insegura. Esto muestra cómo la dependencia en implementaciones imperfectas puede degradar la seguridad de la PKI.

Conclusión: Estos ejemplos ilustran que, aunque la criptografía subyacente sea fuerte, los errores de implementación y la manipulación de protocolos externos pueden convertir cualquier punto de la infraestructura en un eslabón débil. La seguridad completa depende no solo de los algoritmos, sino también de un manejo correcto de datos, validaciones y procesos complementarios.

Referencias:

- Null Byte: Asterisk Security Advisory AST-2015-003
- OCSP: OCSP vulnerability explained

8 Apéndice

8.1 Directorios X.500

Definición

- X.500 es un estándar internacional definido por la ITU-T para **directorios de información en red**.
- Permite que organizaciones y sistemas **almacenen, busquen y gestionen información sobre usuarios, recursos y servicios** de manera centralizada.
- Se puede imaginar como un *"libro de registros global"* para redes.

Características principales

1. **Estructura jerárquica:**
   - La información se organiza en forma de árbol (similar al DNS).
   - Cada nodo del árbol es una **entrada (entry)** que representa un objeto con atributos.
2. **Objetos y atributos:**
   - Cada entrada puede tener múltiples atributos.
   - Ejemplo de atributos de un usuario: nombre, correo, departamento, clave pública, etc.
3. **Protocolo de acceso:**
   - X.500 define DAP (*Directory Access Protocol*) para consultar y modificar entradas.
   - LDAP (*Lightweight Directory Access Protocol*) es una versión ligera inspirada en X.500.

Relación con certificados X.509

- Los certificados X.509 surgieron como parte de los mecanismos de **control de acceso de X.500**.
- X.500 proporcionaba la infraestructura de directorios donde se podían almacenar claves públicas y certificados de los usuarios.
- Esto aseguraba que cada clave estuviera asociada correctamente a la identidad del propietario.

Resumen

- X.500 es un estándar de directorios de información en red.
- Organiza datos de manera jerárquica y estructurada.
- Facilita la búsqueda y gestión de usuarios y recursos.
- Sirve como base para almacenar certificados X.509 y soportar PKI.

8.2 uniformizar las técnicas y protocolos de criptografía de clave pública (*)

Cuando se dice que los **PKCS** buscan "*uniformizar las técnicas y protocolos de criptografía de clave pública*", se refiere a que su objetivo es **establecer un estándar común** para que diferentes sistemas, aplicaciones y fabricantes puedan **usar e intercambiar información cifrada o firmada digitalmente de manera compatible y segura**.
En otras palabras:

- **Técnicas uniformes:** Define cómo implementar operaciones criptográficas básicas, como cifrado, descifrado, firma digital, verificación de firmas o derivación de claves, para que todos lo hagan de la misma manera.
- **Protocolos uniformes:** Establece formatos de datos y procedimientos para intercambiar claves, certificados, solicitudes de certificados, mensajes cifrados o firmados, etc.
- **Compatibilidad e interoperabilidad:** Permite que productos distintos (software, hardware, tokens, smartcards) puedan trabajar juntos sin problemas, usando los mismos formatos y reglas.
- **Seguridad confiable:** Evita errores de implementación que puedan surgir al usar métodos distintos o incompatibles.

Por ejemplo, si dos empresas diferentes implementan RSA según **PKCS #1**, sus sistemas podrán cifrar y descifrar mensajes entre sí sin problemas de incompatibilidad.