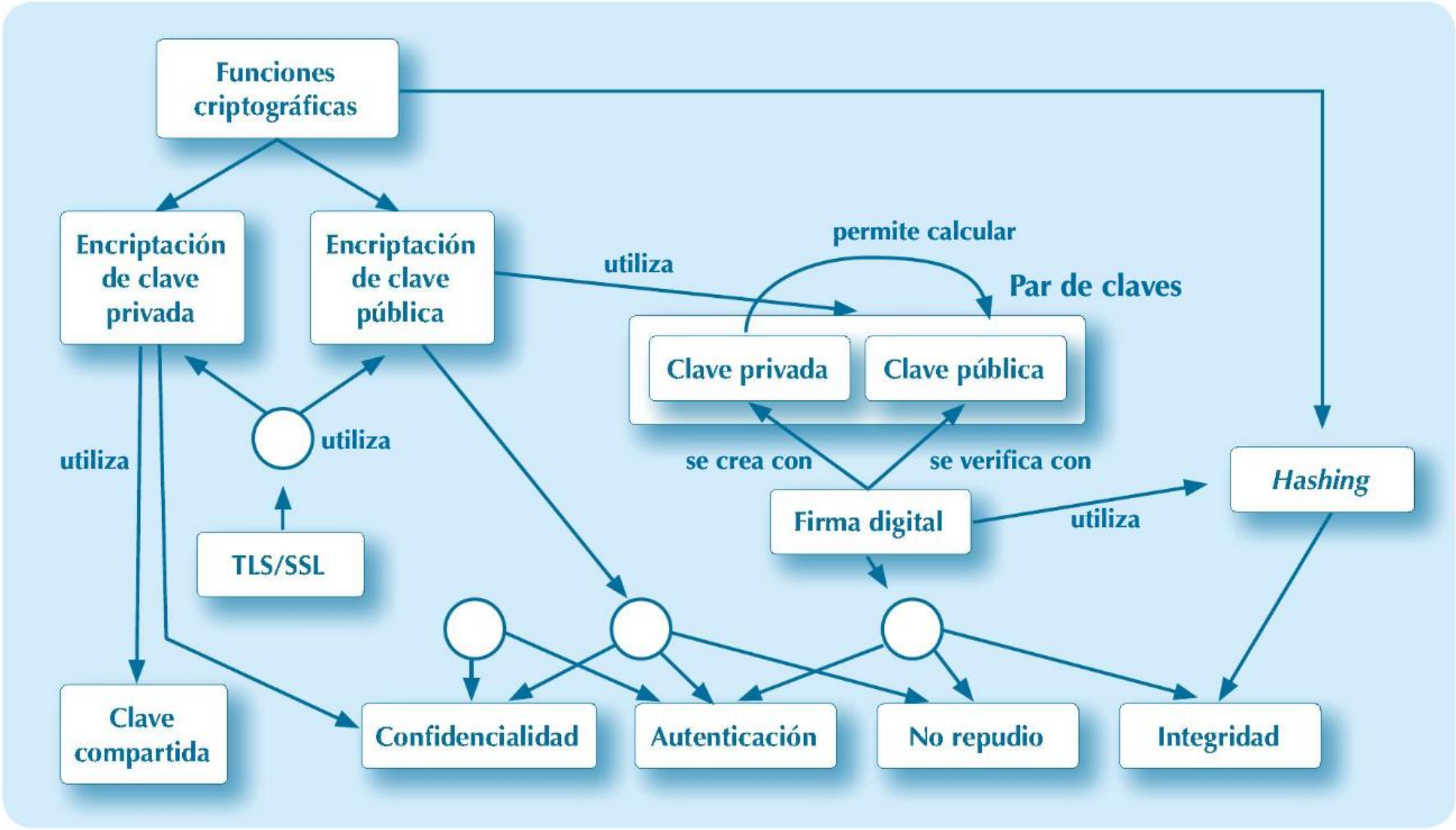
Programación de Servicios y Procesos

UD 4

Técnicas de Programación Seguras


Funciones criptográficas
Encriptación de clave privada
Encriptación de clave pública
utiliza
permite calcular
Par de claves
Clave privada
Clave pública
utiliza
utiliza
se crea con
se verifica con
Hashing
Firma digital
utiliza
TLS/SSL
Clave compartida
Confidencialidad
Autenticación
No repudio
Integridad

Comunicaciones de datos por un medio inseguro

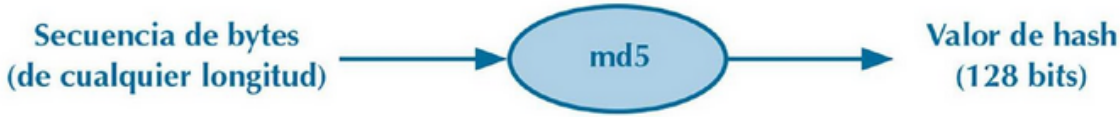
Medio de comunicación inseguro

- Medio no fiable:
  - Los mensajes se pueden deteriorar.
  - Los mensajes pueden no llegar a su destino.
- Medio compartido con terceros ajenos a emisor y receptor. O que pueden conseguir acceso de manera ilegítima al medio. Y que pueden:
  - Leer los mensajes.
  - Alterar el contenido de los mensajes.

Aspectos básicos de seguridad en las comunicaciones

- Integridad
  - El mensaje que recibe el receptor es el mismo que ha enviado el emisor o, en cualquier caso, el receptor puede verificar si es así.
  - Las alteraciones a posteriori en un mensaje envía el emisor podrían deberse a:
    - Fallos del medio.
    - Acciones ilegítimas de terceros con acceso al medio.
- Confidencialidad
  - Un mensaje transmitido por el medio solo es inteligible para su receptor previsto.
- Autenticación
  - El receptor de un mensaje puede verificar que su emisor es quien dice ser.
- No repudio
  - El receptor de un mensaje puede demostrar que fue emitido por el emisor.

Hashing



- No reversibilidad.

  Imposibilidad, en la práctica, de obtener una colisión de hash.

- Uniformidad.

  Cada posible valor de hash corresponde aproximadamente al mismo número de posibles secuencias de entrada.

- Discontinuidad.

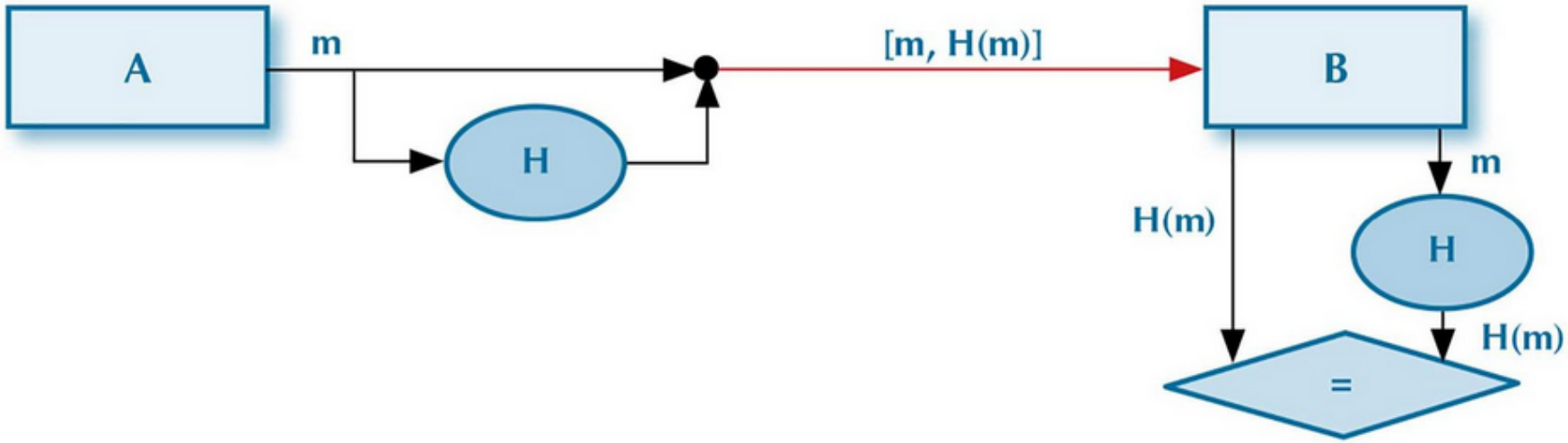
  Pequeñas variaciones en la secuencia de bytes dan como resultado grandes variaciones en el valor de hash calculado.

```
aGk1.Fa7dD ──────────────────► 2dccb73104eed557bec89f7831ec1903
aGk1.Ga7dD ──────────────────► b8a5c29885709f491240ff238c480b2b
```

Principales funciones de hashing

Función	Descripción
MD5	Creado por Ronald Rivest en 1991. Produce valores de 128 bits. Se utiliza ampliamente para verificar la integridad de ficheros. Se han descubierto vulnerabilidades que hacen que, en la actualidad, no se considere apropiado para su uso en criptografía. De hecho, por su relativamente corto tamaño de *hash* (128), se considera que puede ser vulnerable incluso a ataques de fuerza bruta.
SHA-1	Produce valores de 160 bits. Desarrollado por la National Security Agency (NSA). Hoy en día tampoco se considera seguro.
SHA-2 y SHA-3	Producen valores de 224, 256, 384 o 512 bits, según la variante. SHA3 tiene variantes que permiten generar valores de *hash* de longitud arbitraria.

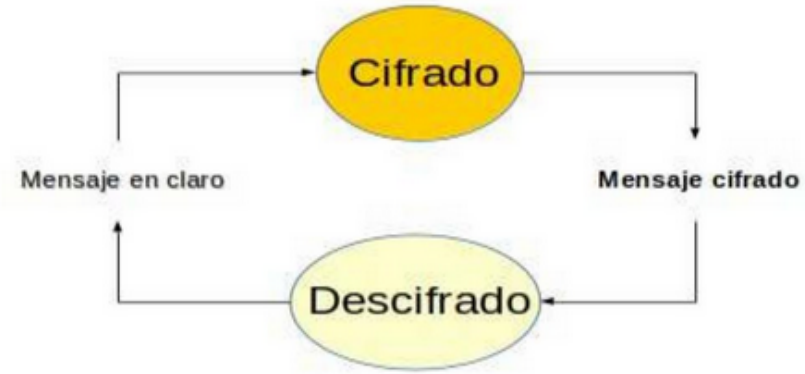
Uso de hashing para verificar la integridad de mensajes



Se adjunta al mensaje m el valor de calcular la función de hash H para él, es decir, H(m).

- Integridad. El receptor calcula el valor de H para el mensaje recibido. Si este valor es el mismo que el adjunto al mensaje, entonces el mensaje recibido es el mismo que el enviado.

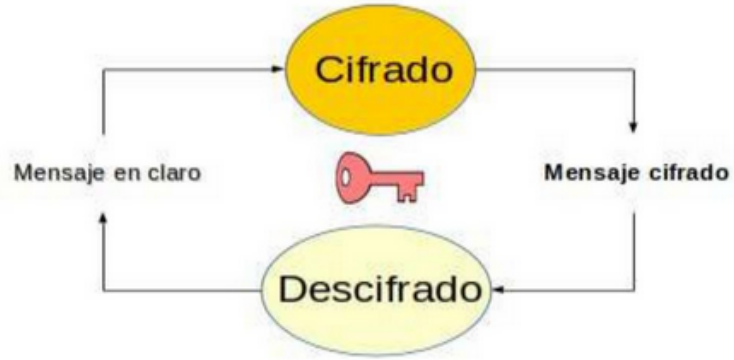
Cifrado (o encriptación) y descifrado (o desencriptación)



- El mensaje cifrado (o encriptado) se calcula mediante una función de cifrado, y es ininteligible.
- Se puede obtener el mensaje en claro a partir del mensaje cifrado, mediante una función de descifrado (o desencriptación).
- Es decir, una operación de cifrado es reversible (al contario que, por ejemplo, el cálculo de una función de hash).
- Para cifrado y descifrado se utiliza determinada información (claves).

Criptografía de clave privada (o simétrica)

- Se usa la misma clave para cifrado y para descifrado.
- Problema de generación y distribución de claves compartidas: no apropiada para comunicaciones ocasionales.



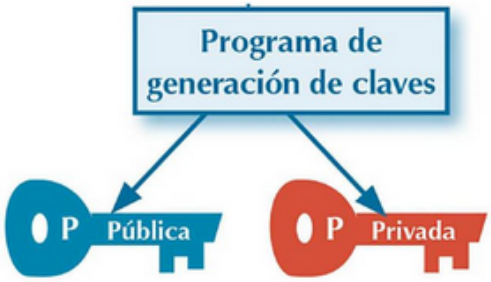
- Autenticación. Solo quien conoce la clave puede cifrar un mensaje con ella.
- Confidencialidad. Solo quien conoce la clave puede descifrar un mensaje cifrado.

✗ No repudio. El receptor no puede demostrar que el mensaje ha sido generado por el emisor. El emisor puede alegar que el mensaje ha sido generado por el receptor, porque también conoce la clave.

Principales algoritmos de criptografía de clave privada

DES (1976)	Cifrado por bloques de 64 bits. Tamaño de clave de 56 bits. 1976. Estuvo desde el principio sujeto a controversia, debido a su pequeño tamaño de clave y a aspectos secretos del diseño original.
3DES (1998)	Cifrado por bloques. Longitud efectiva de clave es 112 bits. En la actualidad no se considera seguro, pero se sigue usando ampliamente.
AES (2001)	Familia de algoritmos de cifrado, con longitudes de clave de 128, 192 y 256 bits.
Blowfish (1993)	Cifrado por bloques. Claves de 32 a 448 bits. Seguro, muy popular y ampliamente implementado. Diseñado como algoritmo de uso general, como alternativa a otros algoritmos habituales, de propiedad privada, patentados, o auspiciados por el gobierno de los EE. UU.
ChaCha20 (2008)	Cifrado de flujo.

Criptografía de clave pública (o asimétrica)



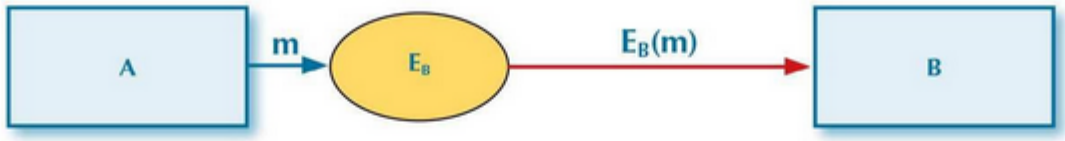
- Par de claves relacionadas entre sí, una pública y una privada, entre las que existe una relación matemática.
- No se puede obtener la clave privada a partir de la pública.
- El propietario B de un par de claves puede publicar su clave pública, pero debe mantener en secreto su clave privada.

Funciones relacionadas E_B y D_B.

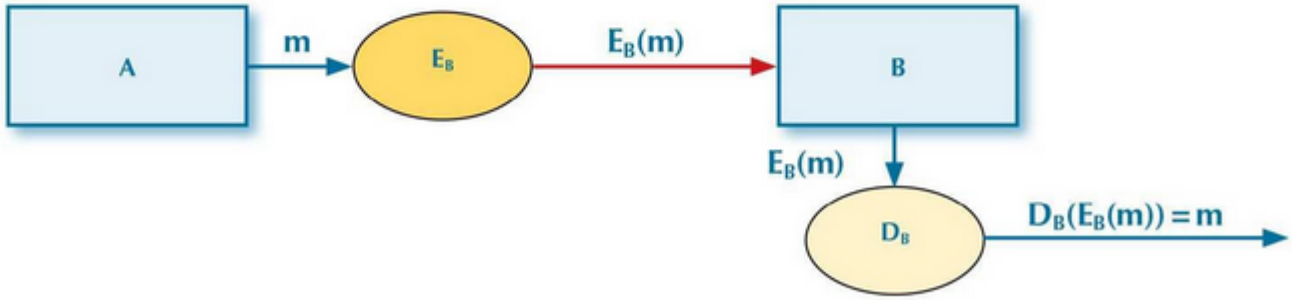
- E_B usa la clave pública de B.
- D_B usa la clave privada de B.

Inversas una de la otra: $D_B(E_B(m)) = m = E_B(D_B(m))$

Cifrado con clave pública y descifrado con clave privada



- Cualquiera puede cifrar un mensaje m con la clave pública de B, calculando $E_B(m)$. Porque para calcular E_B solo hace falta la clave pública de B.
- Solo B puede descifrar un mensaje cifrado con su clave pública, calculando $D_B(E_B(m)) = m$. Porque para calcular D_B hace falta la clave privada de B, que solo conoce B.

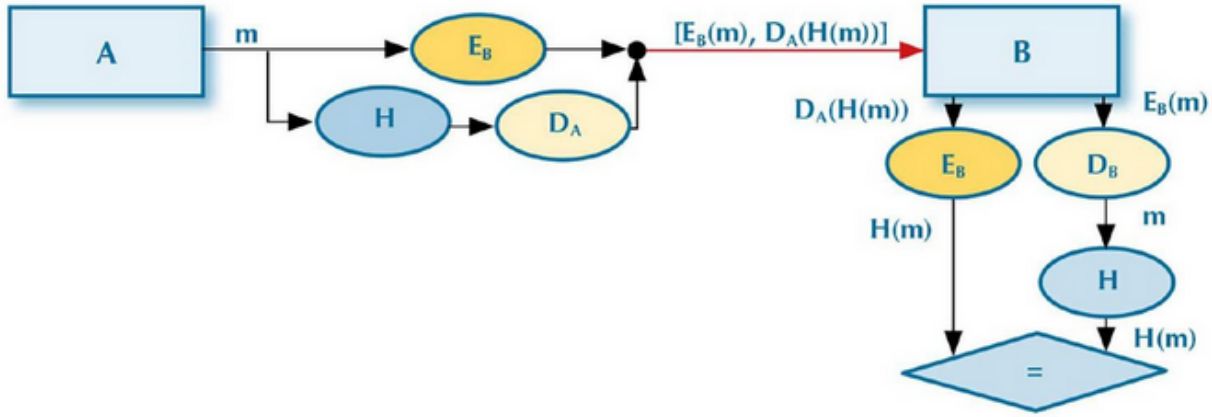


Confidencialidad sin claves secretas compartidas

Principales algoritmos de criptografía de clave pública

RSA	Es el algoritmo criptográfico más antiguo, llamado así por sus creadores (Rivest, Shamir y Adelman). Está basado en la factorización de números muy grandes en dos números primos. Sigue siendo el más utilizado.
ECC	Elliptic curve cryptography, o criptografía de curvas elípticas. Basado en las matemáticas de las curvas elípticas.
DSA	Digital signature algorithm. Como su nombre indica, sirve para firmar, pero no para cifrar información. Es mucho menos rápido que RSA para firma digital, pero más rápido para la verificación de la firma.

Firma digital



Se adjunta, al mensaje m, $D_A(H(m))$: el resultado de aplicar D_A a H(m).

- Autenticación. Solo A puede calcular $D_A(H(m))$, porque para calcular D_A hace falta la clave privada de A.
- Integridad. El receptor B puede obtener el valor del hash H(m) calculado por el emisor sobre el mensaje original m, como $E_A(D_A(H(m))) = H(m)$. Para ello solo necesita la clave pública de A. Puede además calcular el valor de H para el mensaje recibido. Si coinciden, el mensaje recibido es idéntico al original.
- No repudio. Solo A puede calcular $D_A(H(m))$, porque para ello hace falta su clave privada. Por tanto, el receptor B puede demostrar ante terceros que es A quien ha creado el conjunto $[m, D_A(H(m))]$.

- Para tener confidencialidad, basta enviar $E_B(m)$ en lugar de m. Solo B puede obtener el mensaje original, calculando $D_B(E_B(m)) = m$.

Certificados digitales

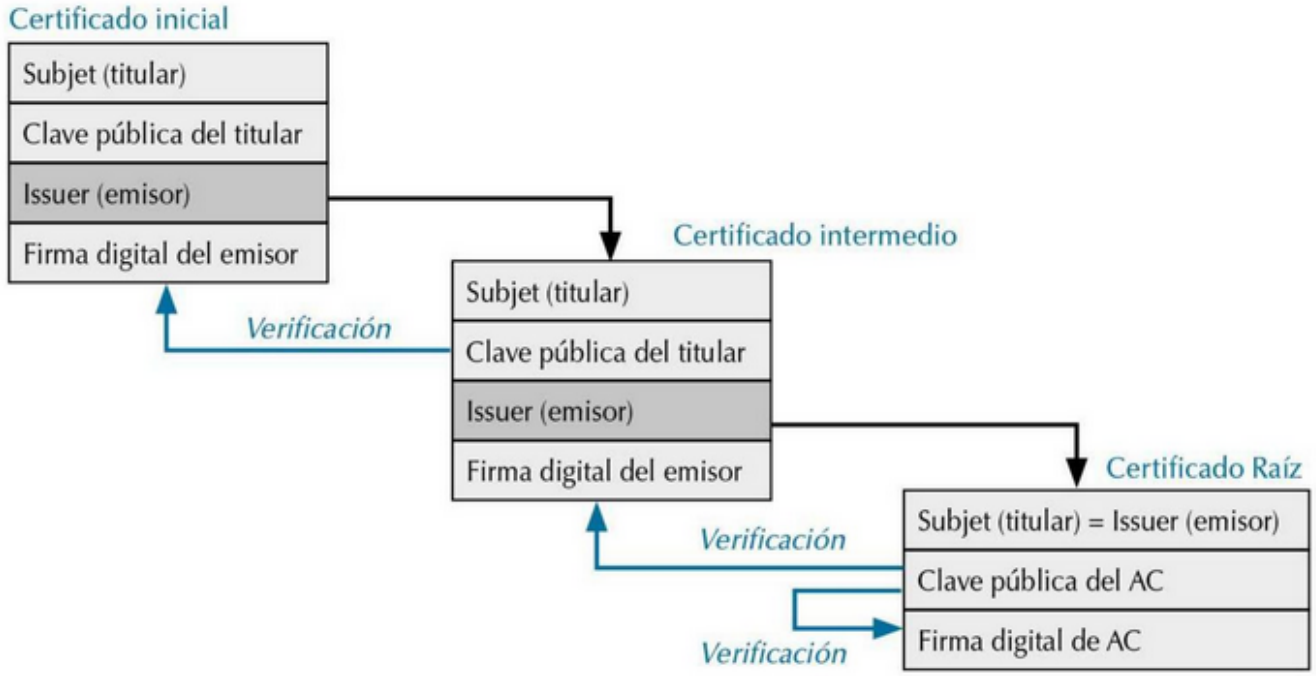
- Un certificado digital es un documento que contiene información utilizada para criptografía de clave pública y que permite acreditar la identidad de su poseedor o titular.
- Está firmado digitalmente por una autoridad certificadora (AC), para garantizar su integridad y para identificar a su creador.
- Un certificado raíz es un certificado digital que contiene la clave pública de una AC, y que está firmado por ella misma (es un certificado autofirmado).

Estructura de un certificado digital X.509

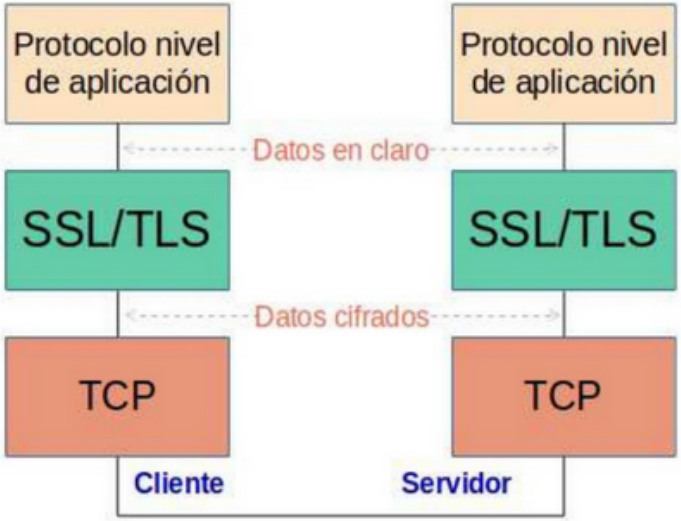
			Versión 1	Versión 2	Versión 3
Version		Identificador de la versión.			
Serial Number		Número de serie. Cada certificado digital emitido por una entidad certificadora debe tener un número de serie distinto.			
Signature Algorithm ID		Identificador del algoritmo utilizado para generar la firma digital del certificado.			
Issuer Name		Nombre del emisor, es decir, de la entidad que ha creado el certificado.			
Validity period	Not before	Periodo de validez del certificado, delimitado por la fecha inicial (*not before*) y la fecha final (*not after*).			
	Not after				
Subject name		Nombre de la persona o entidad a la que se identifica en el certificado, es decir, del titular.			
Subject Public Key Info	Public Key Algorithm	Información de clave pública. Incluye la identificación del algoritmo para el que está creada la clave (los distintos algoritmos utilizan distintos tipos de claves públicas), y la clave pública en sí.			
	Subject Public Key				
Issuer Unique Identifier		Identificador numérico correspondiente al campo anterior "Issuer Name". Si está presente, lo sustituye como identificador del emisor del certificado (*issuer*).			
Subject Unique Identifier		Identificador numérico correspondiente al campo anterior "Subject Name". Si está presente, lo sustituye como identificador del titular del certificado (*subject*).			
Extensions		Extensiones. Se explican a continuación.			
Certificate Signature Algorithm		Identificación del algoritmo de firma digital empleado para generar la firma digital de todos los contenidos previos, y firma digital realizada con este algoritmo utilizando la clave privada del emisor (*issuer*) del certificado.			
Certificate Signature					

Validación de un certificado

- Firma digital correcta.
- Fecha actual dentro del período de validez.
- Cadena de confianza desde el certificado hasta un certificado raíz de una AC (autoridad de certificación).



TLS/SSL



- Capa o nivel para encriptación de datos sobre el protocolo de transporte TCP.
- Permite transmisión de datos cifrados sobre TCP.
- Protocolos de nivel de aplicación funcionan sin cambios sobre TLS/SSL. Solo cambia el proceso de establecimiento de conexión.
- Combina técnicas de criptografía de clave privada y de clave pública (esta última no es apropiada para transmisión de grandes cantidades de datos).
- Durante una fase inicial de negociación o handshake, cliente y servidor acuerdan un cipher suite o conjunto de algoritmos que utilizar, y una clave de sesión compartida y secreta para un algoritmo de clave privada, que se usa en adelante para cifrar los datos transmitidos sobre TCP.
- Utiliza certificados digitales, lo que permite autenticación de servidor y de cliente.

Seguridad en Java

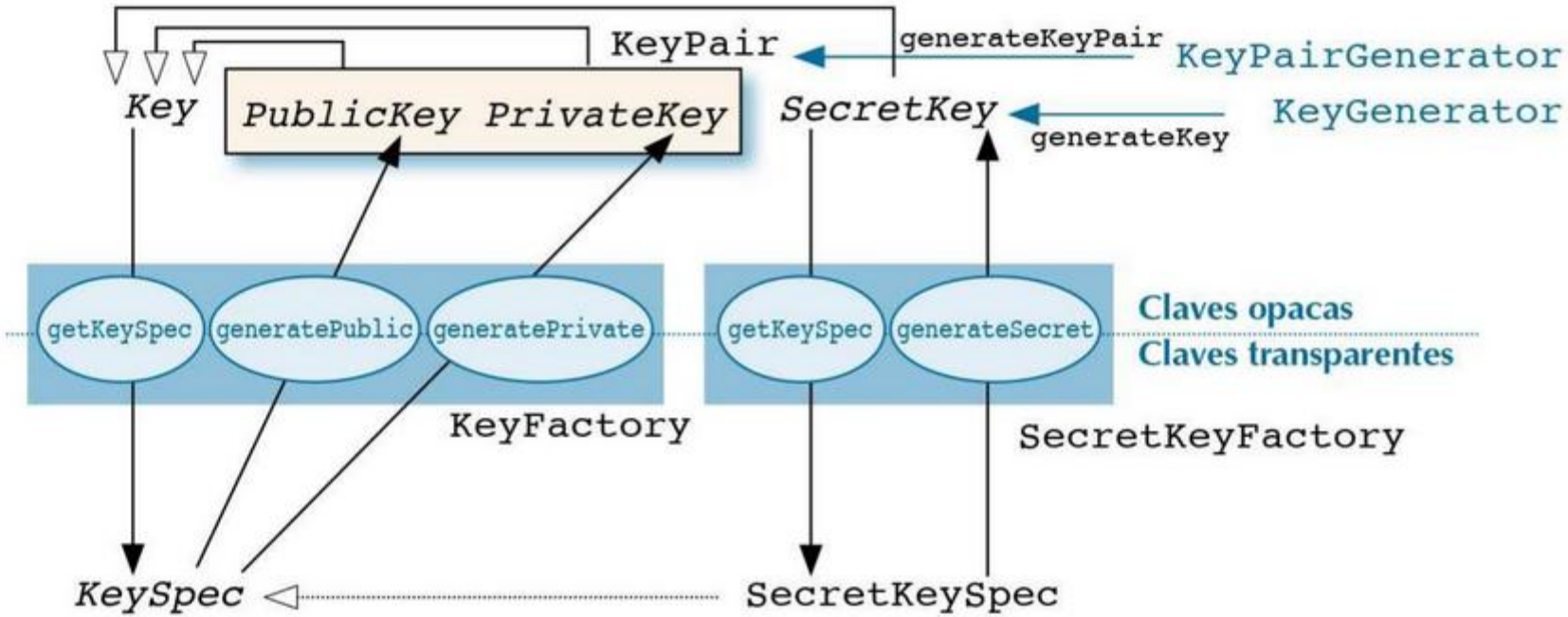
Existen algunas API de Java para diversos aspectos de la seguridad.

- JCA (Java chriptography architecture). Operaciones criptográficas. Un CSP (proveedor de servicios criptográficos) puede proporcionar clases que implementan interfaces incluidas en JCA. Incluye engine classes:
  - `MessageDigest`. Funciones de hash o digest.
  - `KeyGenerator`. Generación y gestión de claves para criptografía simétrica.
  - `KeyPairGenerator`. Generación y gestión de pares de claves para criptografía de clave pública.
  - `Cipher`. Cifrado de datos.
  - `Signature`. Firma digital.
- JSSE (Java secure socket extension). Soporte para TLS y SSL mediante clases tales como `SSLServerSocket` y `SSLSocket`.
- JAAS (Java Authentication and Authorization Service).
  - Autenticación de usuarios que ejecutan código de Java.
  - Autorización de usuarios para realizar determinadas acciones.

Engine Classes de JCA

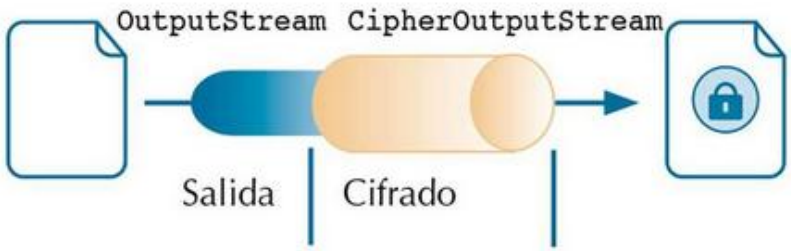
Clase	Descripción
`MessageDigest`	Para funciones *hash* o *digest*.
`KeyGenerator`	Para generación y gestión de claves para criptografía simétrica.
`KeyPairGenerator`	Para generación y gestión de pares de claves para criptografía asimétrica o de clave pública.
`Cipher`	Para algoritmos de cifrado de datos.
`Signature`	Para algoritmos de firma digital.

Interfaces y clases para generación y gestión de claves

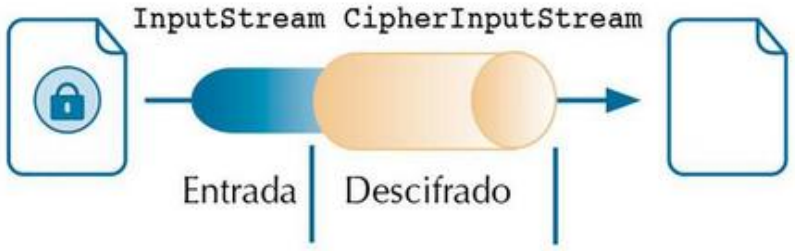


Streams cifrados

- Son un filtro más en la jerarquía de clases `Stream`
- Apropiados para criptografía de clave privada.



`CipherOutputStream (OutputStream os, Cipher c)`



`CipherInputStream (InputStream is, Cipher c)`