UT 5 – Seguridad y Criptografía

Programación de Servicios y Procesos
Curso 2024-25

Profesor: Agustín González-Quel

Seguridad Informática

La seguridad informática es una rama de la seguridad que se dedica a proteger los sistemas informáticos de amenazas externas e internas.

Amenazas:

- Robo o daños en hardware.
- Robo o alteración de datos.
- Violación de privacidad.
  - En sistemas.
  - En comunicaciones.
- Interrupción de servicio.
- Suplantación de identidad.
- …

Información práctica

- INCIBE: https://www.incibe.es/
- National Cyber Security Centre: https://www.ncsc.gov.uk/

Seguridad Informática

La seguridad de la información ha pasado de ser un conjunto de normas a convertirse en una disciplina compleja.

- La seguridad es compleja porque los ataques son complejos.
- Debemos desarrollar una comprensión de los ataques antes de poder desarrollar sistemas seguros.

- El punto de entrada de un ataque puede ser un usuario del sistema, el propio sistema o la red entre ambos.

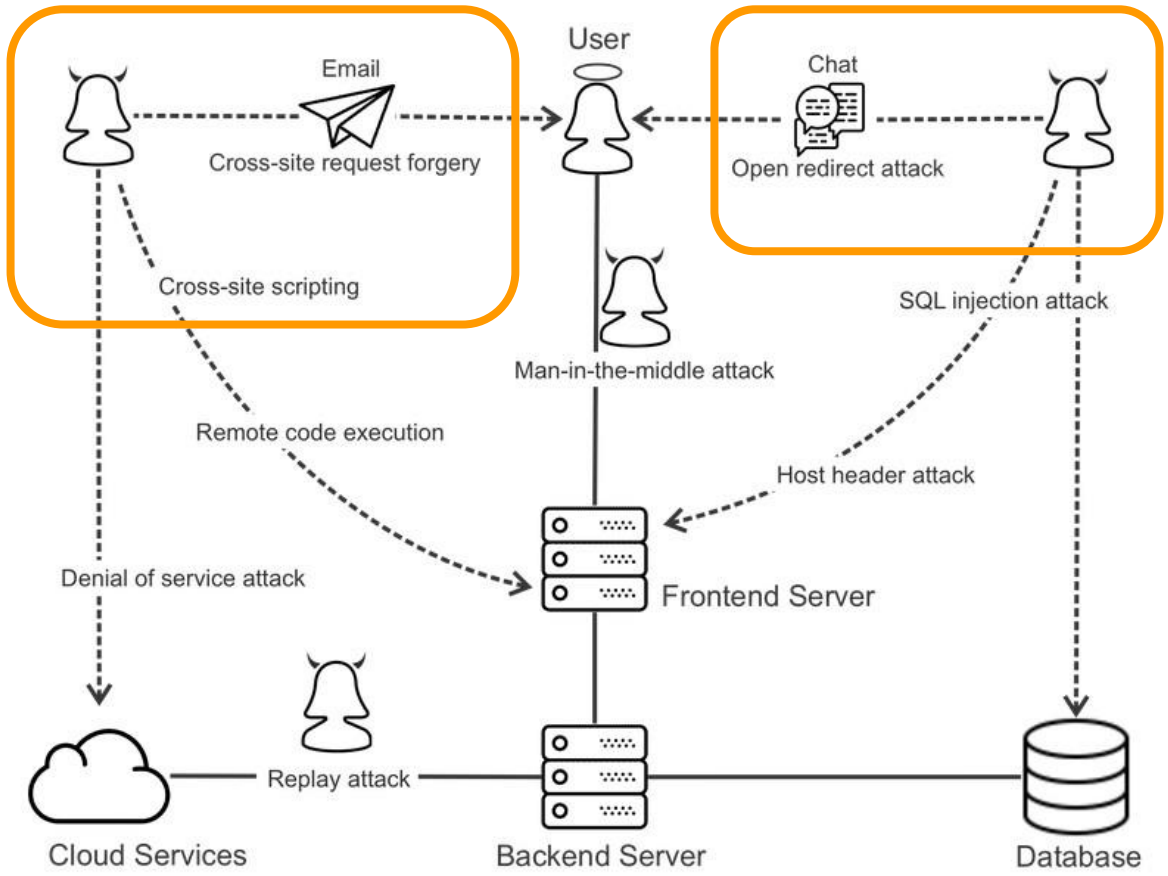
- Todo ataque comienza con un punto de entrada vulnerable.
  - La suma de todos los puntos de entrada potenciales se conoce como superficie de ataque.
  - Cada sistema tiene una superficie de ataque única. Los ataques y las superficies de ataque están en constante evolución.
  - Proteger la superficie de ataque de todos los componentes de una organización es un objetivo compartido de todos sus miembros.

Ataques más comunes (1)

Ataques de Usuario

Clave: Mecanismos que usan agentes externos para escribir en nuestros sistemas código que posteriormente se ejecutará

- Cross Site Scripting (XSS): Se almacena en la BD y aprovecha formularios de búsqueda, chats, etc. Otros usuarios acceden y se ejecuta el código.
- Cross Site Request Forgery: Al acceder a un sitio web, nos sitúa un script en nuestro navegador que se activa al acceder a tercer sitio (banco, etc.)
- Ingeniería social: phishing, smishing, etc
- Open redirect: El punto de entrada puede ser un email o sitios web que permiten redirigir la navegación por llamada URL.



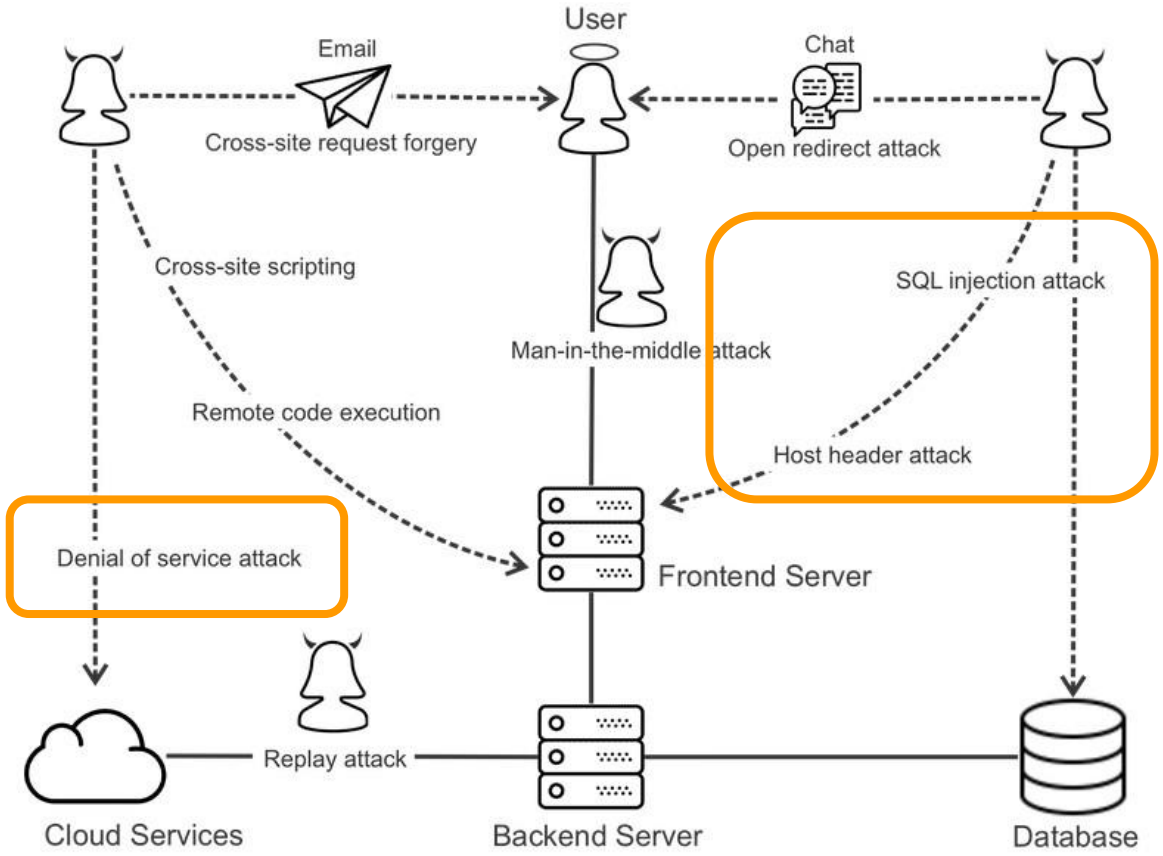
Ataques más comunes (2)

- Inyección SQL: Introduce código SQL en un formulario de forma que se ejecuta una query diferente a la que el desarrollador pretendía.

```
Select user from users where user =“a” and password =’xxx’
Select user from users where user =“a” or ‘1’=’1’ –‘ and password =’xxx’
```

- Host header attack: El header se usa en entornos web para redireccionar tráfico entrante a varios sitios web internos. Si no se controla la información de entrada de la petición, puede aceptarse código malicioso.

- Denegación de servicio: Llamadas masivas para bloquear un sistema.
  - A veces basado en malware.

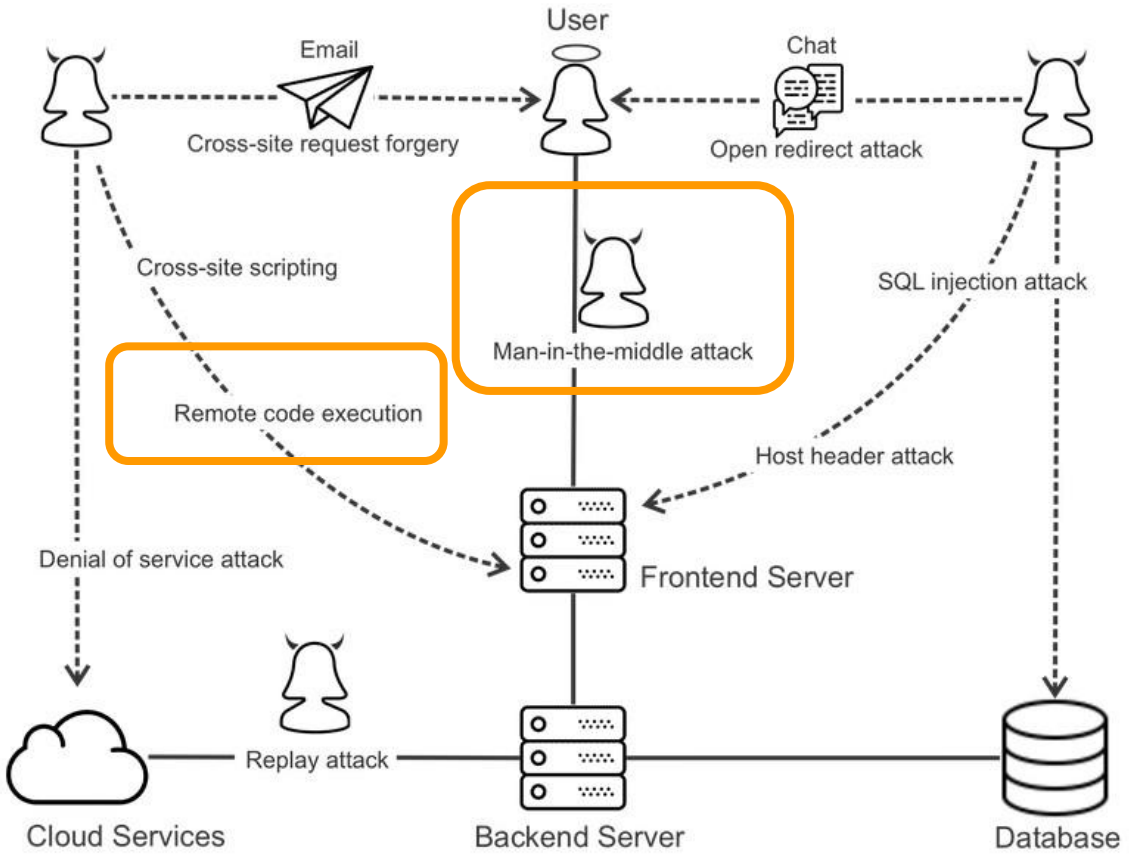


Ataques más comunes (3)

- Remote code execution: Los ataques RCE suelen comenzar como una vulnerabilidad en una aplicación de cara al público que es usada para ejecutar comandos en la máquina subyacente.
  - CVE-2019-8942: vulnerabilidad en WordPress 5.0.0, que permite a los atacantes ejecutar código cargando un archivo de imagen especialmente diseñado que incluya código PHP en sus metadatos.

Ataques a la red

- Man in the middle: Programa que se sitúa entre 2 elementos que se comunican e intercepta la comunicación
  - Encriptación, uso de protocolos seguros.
- Replay attack: similar, solo que envía repetidamente mensajes válidos.



Nos centramos en

- Repaso de técnicas de programación segura
- Seguridad en comunicaciones, requisitos y algunos mecanismos
  - Hash
  - Encriptación

Prácticas de programación seguras

- Análisis de datos de entrada, parámetros
  - Rangos de valores, tipos, patrones, …
- Políticas seguras de acceso: autenticación, tokens
  - Caducidad de credenciales.
  - Políticas de doble autenticación
  - ACL, roles
- Registro de sucesos del sistema (logs)
  - Uso de librerías: ejemplo: logging (https://docs.python.org/3/library/logging.html)
  - IP que acceden a nuestros servicios
  - Usuarios, actividades, etc.
- Protección ante errores, uso de excepciones controladas.
- Protección adecuada de credenciales.
  - Almacenamiento, envío, …
- Pruebas de nuestros sistemas, incluyendo pruebas de carga (Jmeter, https://jmeter.apache.org/)
- Auditorías de seguridad.

Atención

Este es un conjunto mínimo de prácticas que habrá que ir enriqueciendo y completando con nuestra experiencia profesional en el día a día

Seguridad en comunicaciones

Dentro del amplio campo de la seguridad nos centraremos en la seguridad de las comunicaciones entre procesos.

Se plantea la dificultad adicional de que las comunicaciones se realizan sobre un medio no seguro.

La seguridad en este campo implica:

- Integridad: El mensaje se recibe inalterado
- Confidencialidad en las comunicaciones: Ningún tercero puede ver los datos que se envían entre las partes.
- Autenticación (No suplantación): Cada parte de las comunicaciones es quien dice ser.
- No repudio: El receptor no puede negar que ha recibido un mensaje
- Acceso autorizado.

Building block	Solutions
Data integrity	Secure networking protocols Version control Package management
Authentication	User authentication System authentication
Data authentication	User registration User login workflows Password reset workflows User session management
Non-repudiation	Online transactions Digital signatures Trusted third parties
Authorization	User authorization System to system authorization File system access authorization
Confidentiality	Encryption algorithms Secure networking protocols

Cifrado o encriptación

- Codificación de la información para modificar la representación original.
  - El objetivo es hacerlo ilegible para alguien que pueda ver el mensaje.
  - El receptor del mensaje tendrá el mecanismo para poder descifrar la información.
- Se viene aplicando desde miles de años atrás y ha ido evolucionando con el tiempo.

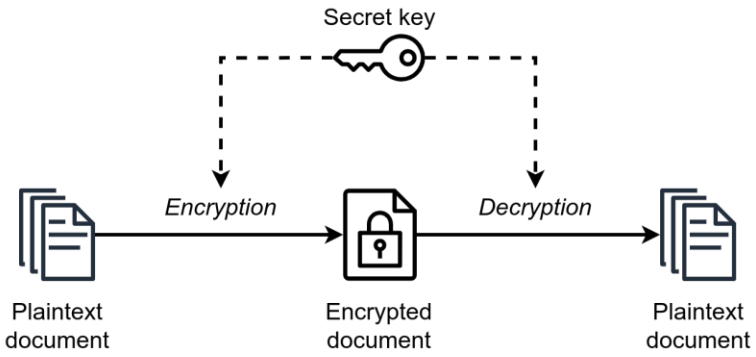
Métodos más sencillos: cifrado clásico y clave simétrica

- Un ejemplo sencillo es el cifrádo clásico o por sustitución
  - Ver fichero: u5-cifradoCutre.py

- Más sofisticado: sistemas de clave simétrica modernos, basados en:
  - Clave
  - Algoritmo matemático de generación del mensaje: AES, DES, etc.
- Aunque el atacante conozca el algoritmo, sin la clave no puede hacer nada.
- Puede averiguarse la clave usando combinaciones de caracteres, por lo que el tamaño de la clave es un recurso básico para garantizar la fortaleza del sistema

  AES: 56 bits, 2^{56} claves posibles (72.057.594.037.927.936 claves).
  3DES, Blowfish e IDEA tienen claves de 128 bits, 2^{128} posibilidades

- Resuelve
  - Confidencialidad en los mensajes.
  - Integridad en los mensajes
- Problema:
  - Cifrádo clásico: la distribución de la clave.
  - Métodos modernos: tamaño de la clave



Encriptado: funciones hash

Una función hash es una función matemática que convierte un valor numérico de entrada en otro valor numérico comprimido. La entrada a la función hash es de longitud arbitraria, pero la salida es siempre de longitud fija.

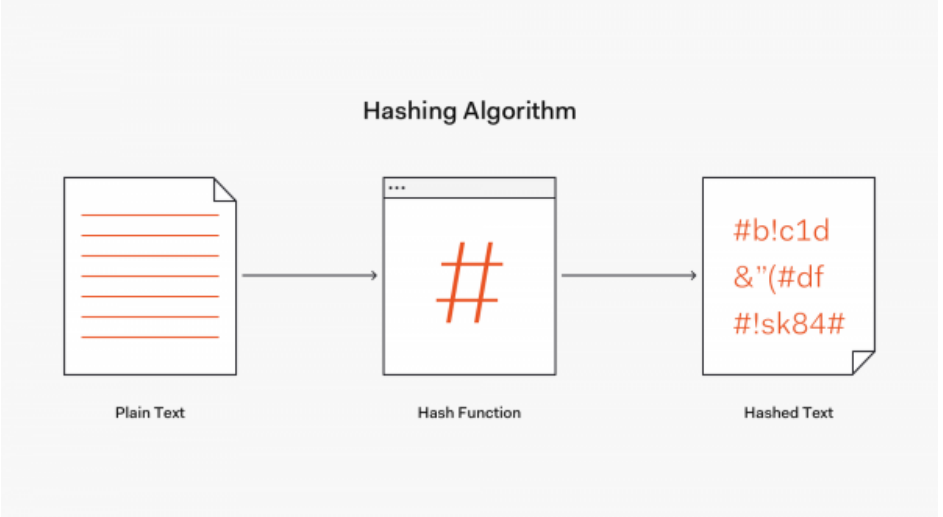
Características

- Cada entrada diferente dará un resultado diferente (minimizar colisiones)
- No reversibilidad: No es posible recuperar el contenido fuente teniendo el hash
- Discontinuidad: Pequeñas modificaciones de la entrada darán hashes muy distintos.

Aplicaciones

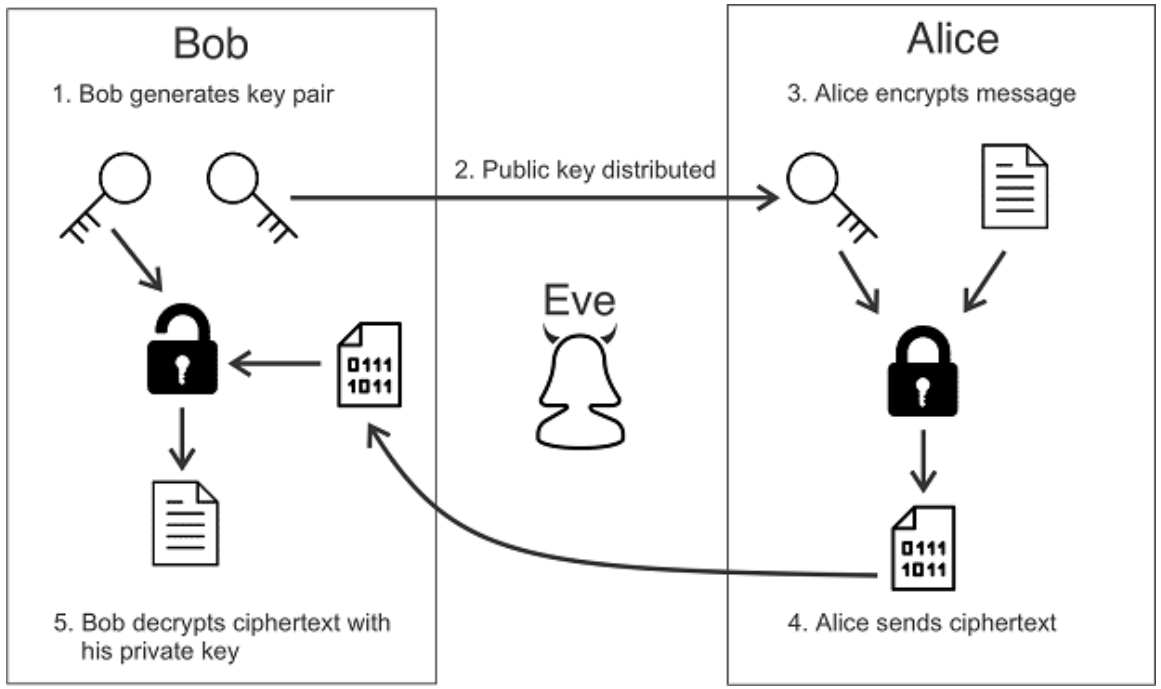
- Integridad en los mensajes.
- NO RESUELVE:
  - Problemas de suplantación o no-repudio.
  - Confidencialidad (a medias …)

Ejemplos: u5-compareFilesHash.py, u5-checkHash4File.py



Clave asimétrica

- Es un algoritmo de cifrado que utiliza una clave para cifrar diferente que la clave de descifrado.
- Eso permite que el emisor y receptor de un mensaje tengan cada uno su clave.
- En la práctica, una clave nunca viaja del propietario y la otra la distribuye a quien quiere que reciba sus mensajes
- En la figura
  - Sólo la clave privada de Bob puede descifrar el texto cifrado producido por la clave pública de Bob.
  - Aunque Eve tenga la clave pública de Bob y el texto cifrado de Alice, no podrá descifrar el mensaje.
- Escalabilidad
  - Alice puede enviar su mensaje tantas personas como tengan par de claves.
  - Si Eve consigue la clave privada de una persona, no afectará a los demás participantes.



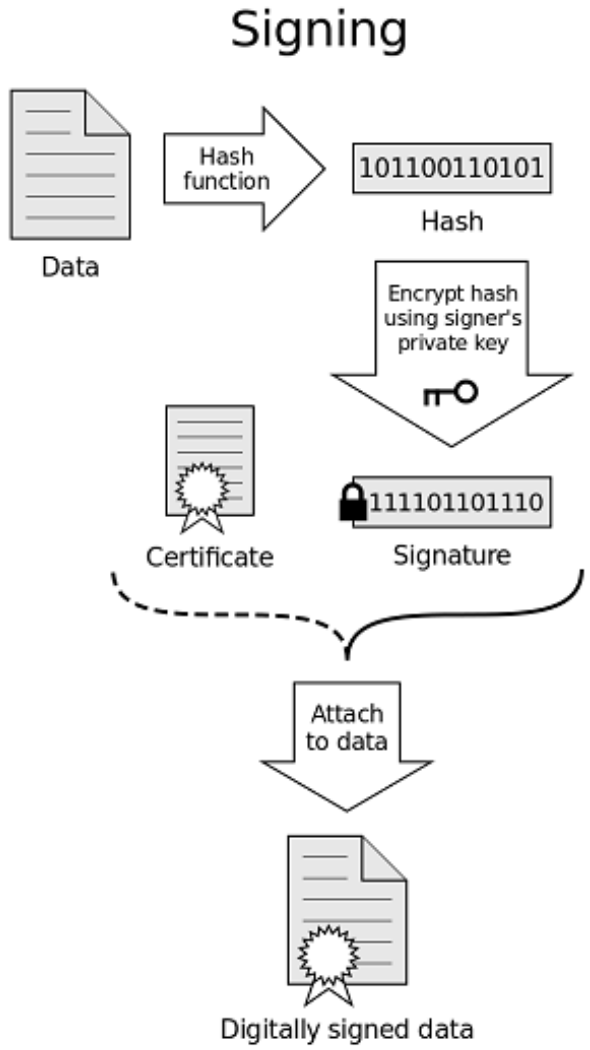
Clave asimétrica (2)

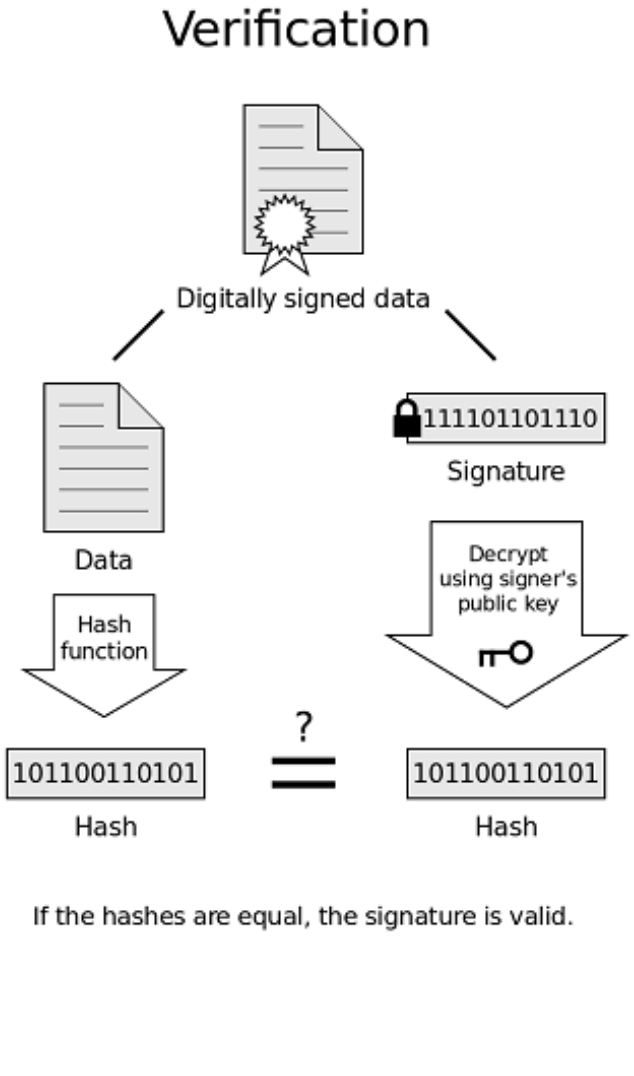
Ventajas

- Tiene una alta tasa de confidencialidad e integridad.
- Permite utilizar canales abiertos y públicos de comunicación.

Base para los sistemas de Firma Electrónica:

- La firma de un documento es el hash del documento encriptado con el par de claves de la persona/entidad firmante.





Ejemplo en Python

Pasos del ejemplo

- Creación de par de claves: privada y pública.
- Uso de las claves en operaciones de encriptado/desencriptado.

- Lo presentamos en Jupyter Notebook (https://jupyter.org/)
  - Jupyter permite crear documentos que mezclan código y texto.
  - Se ejecutan en un entorno propio, en servidor web o como extensión de VS Code.
  - El texto se escribe en un lenguaje de marcas sencillo (MD) usado también en Gitlab y Github.
    - https://www.markdownguide.org/cheat-sheet/

  - Google Drive tiene una herramienta similar: Colab Notebooks

Final UT 5 - Seguridad y Criptografía