Seguridad Informatica - Practica1: Criptografía

El objetivo de esta práctica es entender y familiarizarse con los 3 tipos de criptografía: simétrica, asimétrica y hash. Sobre estos de criptografías es que se implementan los sistemas de seguridad digitales que usamos a diario.

Preparación del Entorno

Para esta práctica utilizaremos `openssl` que es una herramienta de línea de comandos que nos permite realizar operaciones criptográficas. En la mayoria de las distribuciones de Linux, incluyendo la maquina virtual proporcionada, `openssl` ya viene instalado. Si no lo tienes instalado, puedes hacerlo usando el gestor de paquetes de tu distribución. Por ejemplo, en Debian/Ubuntu puedes usar:

```
sudo apt-get install openssl
```

Aunque `openssl` tambien incluye soporte para criptografía hash, para esta práctica usaremos la herramienta `sha256sum` que también viene preinstalada en la mayoría de las distribuciones de Linux. Si no la tienes instalada, puedes hacerlo usando el gestor de paquetes de tu distribución. Por ejemplo, en Debian/Ubuntu puedes usar:

```
sudo apt-get install coreutils
```

OpenSSL es una herramienta ampliamente utilizada en el mundo real, en HTTPS, VPNs, Docker, Google Cloud, entre otros.

Criptografía Simétrica

En la criptografía simétrica, la misma clave se utiliza tanto para cifrar como para descifrar la información. Para esta parte de la práctica, utilizaremos el algoritmo AES (Advanced Encryption Standard) con una clave de 256 bits.

Explicado de una forma mas matemática

$$C = E(K, P)$$
$$P = D(K, C)$$

Sea:

- C: Texto cifrado (Ciphertext)
- P: Texto plano (Plaintext)
- K: Clave (Key)
- E: Función de cifrado (Encryption function)
- D: Función de descifrado (Decryption function)

Como es lógico, para que este sistema funcione, tanto el emisor como el receptor deben conocer la clave K y mantenerla en secreto. El hecho de distribuir y gestionar las claves de forma segura es uno de los principales desafíos de la criptografía simétrica.

Cabe mencionar tambien que en la mayoria de los casos la función de cifrado y la de descifrado son inversas matemáticamente, es decir, aplicar la función de descifrado a un texto cifrado con la misma clave nos devuelve el texto plano original. Es decir, aplicar la función de descifrado es como "deshacer" la función de cifrado, aplicando la inversa de sus operaciones matemáticas.

Criptografía Asimétrica

En la criptografía asimétrica, se utilizan dos claves diferentes: una clave pública y una clave privada. Las claves están matemáticamente relacionadas, pero no es posible obtener la clave privada a partir de la clave pública y viceversa. Ambas claves se utilizan para cifrar y descifrar la información, pero de manera diferente. Los datos cifrados con la clave pública solo pueden ser descifrados con la clave privada correspondiente, y los datos cifrados con la clave privada solo pueden ser descifrados con la clave pública correspondiente.

Explicado de una forma mas matemática:

$$C = E(K_{pub}, P)$$

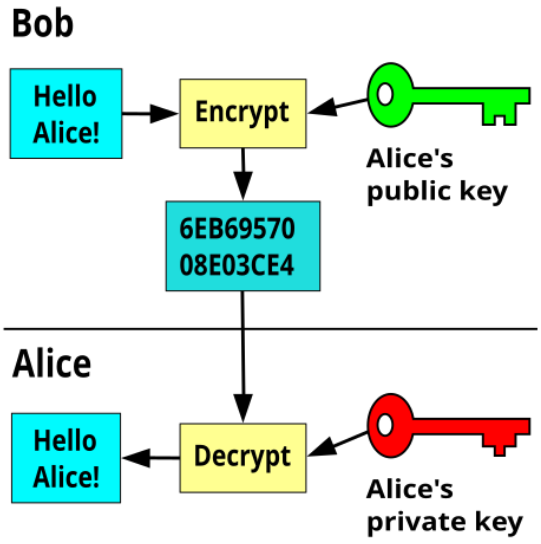
$$P = D(K_{priv}, C)$$

$$C' = E(K_{priv}, P)$$

$$P = D(K_{pub}, C')$$

Sea:

- C: Texto cifrado (Ciphertext)
- P: Texto plano (Plaintext)
- K_{pub}: Clave pública (Public Key)
- K_{priv}: Clave privada (Private Key)
- E: Función de cifrado (Encryption function)
- D: Función de descifrado (Decryption function)



Entidades Certificadoras Las entidades certificadoras (CA, por sus siglas en inglés) son organizaciones de confianza que emiten certificados digitales. Estos certificados lo que hacen es vincular claves públicas con la identidad de una entidad (persona, organización, dispositivo, etc.) y son fundamentales para la infraestructura de clave pública (PKI).

Las CA verifican la identidad de las entidades antes de emitir un certificado, lo que ayuda a prevenir ataques de suplantación de identidad. Cuando un usuario recibe un certificado digital, puede confiar en que la clave pública contenida en el certificado pertenece realmente a la entidad que dice ser. Ya que el certificado viene encriptado con la clave privada de la CA, cualquier intento de modificar el certificado sería detectable. La clave pública de la CA se utiliza para verificar la firma del certificado y son almacenadas en repositorios públicos.

Criptografía Hash

La criptografía hash se utiliza para garantizar la integridad de los datos. Un hash es una función matemática que toma una entrada (o 'mensaje') y devuelve una cadena de caracteres de longitud fija, que parece aleatoria. Cualquier cambio en la entrada, por pequeño que sea, produce un hash completamente diferente. Esto hace que los hashes sean útiles para verificar la integridad de los datos, ya que cualquier alteración en los datos originales resultará en un hash diferente. En resumen, una función hash cumple las siguientes propiedades:

1. **Determinista**: Para una misma entrada, siempre produce la misma salida.

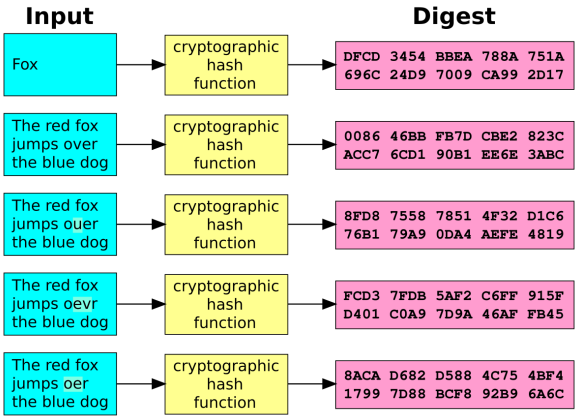
2. **Rápida de calcular**: Es computacionalmente eficiente calcular el hash de cualquier entrada.
3. **Resistente a colisiones**: Es difícil encontrar dos entradas diferentes que produzcan el mismo hash.
4. **Resistente a preimágenes**: Dada una salida hash, es difícil encontrar una entrada que produzca esa salida.
5. **Avalancha**: Un pequeño cambio en la entrada produce un cambio drástico en la salida.

Explicado de una forma mas matemática:

$$H : \{0,1\}^* \to \{0,1\}^n$$
$$H(m) = h$$
$$H(m) \neq H(m') \quad \forall m' \neq m$$

Sea:

- H: Función hash
- m: Mensaje de entrada
- h: Hash del mensaje
- n: Longitud del hash



Relación con los Principios de la Seguridad Informática

La criptografía se relaciona estrechamente con los principios de la seguridad informática, que incluyen la confidencialidad, integridad, autenticidad y no repudio.

1. **Confidencialidad**: La criptografía asegura que la información solo sea accesible para las personas autorizadas. Esto se logra mediante el cifrado de datos, de modo que solo aquellos con la clave adecuada puedan descifrarlos.

2. **Integridad**: A través de funciones hash y firmas digitales, la criptografía garantiza que los datos no sean alterados de manera no autorizada. Cualquier modificación en los datos originales resultará en un hash diferente, lo que indica que la integridad ha sido comprometida.

3. **Autenticidad**: La criptografía permite verificar la identidad de los usuarios y la procedencia de los datos. Las firmas digitales, por ejemplo, aseguran que un mensaje provenga realmente de quien dice ser.

4. **No repudio**: Mediante el uso de firmas digitales, la criptografía proporciona evidencia de que un mensaje fue enviado y recibido, evitando que el remitente niegue su participación en la comunicación.
5. **Disponibilidad**: No depende directamente de la criptografía.

Entregable

A continuación os detallo los elementos que deberéis entregar para esta práctica. El código/comandos necesarios lo teneis disponible y explicado en la carpeta de `Ejemplos/`. Buscad o cread archivos a utilizar en cada parte, pueden ser cualquier cosa, fragmentos de un libro en txt, un pdf, un word, uma imagen, un video, etc. Solo tened en cuenta que el cifrado asimétrico tiene limitaciones en el tamaño de los archivos que puede cifrar. Ademas puede llegar a ser bastante lento. Por esto utilizaremos cifrado hibrido en la parte 4.

Parte 1: Criptografía Simétrica En esta parte de la práctica, deberás generar una clave simétrica utilizando el algoritmo AES con una longitud de clave de 256 bits. Luego, utilizarás esta clave para cifrar un archivo. Debereis entregar el archivo original, la contraseña utilizada, el archivo cifrado y el archivo descifrado.

Parte 2: Criptografía Asimétrica En esta parte de la práctica, deberás generar un par de claves asimétricas (pública y privada) utilizando el algoritmo RSA con una longitud de clave de 2048 bits. Luego, utilizarás la clave privada cifrar un archivo. Debereis entregar el archivo original, las claves generadas, el archivo cifrado y el archivo descifrado.

Parte 3: Criptografía Hash En esta parte de la práctica, deberás calcular el hash SHA-256 de un archivo y luego lo modificaremos para ver cómo afecta al hash. Debereis entregar el archivo original, el archivo modificado y el hash de ambos.

Parte 4: Firma digital + Cifrado híbrido Por último, es importante mencionar que en la práctica, a menudo se combinan diferentes técnicas criptográficas para lograr un nivel de seguridad más robusto. En este caso utilizaremos las técnicas de firma digital y cifrado híbrido para proteger la información de manera efectiva. Esto es especialmente útil en entornos donde la seguridad de los datos es crítica, como en transacciones financieras o en la comunicación de información sensible.

- **Firma digital**: Esta técnica criptográfica consiste en crear un hash del mensaje y cifrarlo con la clave privada del remitente. Esto garantiza la autenticidad e integridad del mensaje, ya que solo el remitente puede generar la firma y cualquier alteración en el mensaje cambiará el hash. Esto es utilizado por ejemplo en el DNIe o en la firma de documentos electrónicos.
- **Cifrado híbrido**: Combina la criptografía simétrica y asimétrica para aprovechar las ventajas de ambos enfoques. Se utiliza una clave pública para cifrar una clave simétrica, que a su vez se utiliza para cifrar el mensaje. Esto permite un intercambio seguro de claves y una cifrado eficiente del mensaje.

En esta parte de la práctica, deberás firmar digitalmente un archivo utilizando un par de claves asimétricas (pública y privada) generadas con el algoritmo RSA. Luego, deberás cifrar el archivo original junto con su firma digital utilizando cifrado híbrido. Para ello, generarás una clave simétrica para cifrar el archivo y luego cifrarás esta clave simétrica con la clave pública del receptor.

Formato de Entrega Deberás entregar un archivo comprimido (zip) con el nombre `Practica1SI-<tu_nombre>-<tu_ape` sin acentos. Por ejemplo, yo entregaría un archivo `Practica1SI-Fabian-Murgado.zip`. El archivo comprido debe tener la siguiente estructura:

- `Practica1SI-<tu_nombre>-<tu_apellido>/`
  - `Parte1/`
    - `<archivo>` (Archivo utilizado para esta parte)
    - `simetrica.key` (Archivo de texto con la contraseña utilizada para crear la clave de criptografía simétrica)
    - `<archivo>.enc` (Archivo encriptado con la clave simétrica)
    - `<archivo>.enc.dec` (Archivo desencriptado con la clave simétrica)
  - `Parte2/`
    - `<archivo>` (Archivo utilizado para esta parte)
    - `asimetrica.priv` (Clave privada generada)
    - `asimetrica.pub` (Clave pública generada)
    - `<archivo>.enc` (Archivo encriptado con la clave privada)
    - `<archivo>.enc.dec` (Archivo desencriptado con la clave pública)
  - `Parte3/`
    - `<archivo>` (Archivo utilizado para esta parte)
    - `<archivo>.hash` (Hash del archivo utilizado)
    - `<archivo_modificado>` (Archivo utilizado para esta parte)
    - `<archivo_modificado>.hash` (Hash del archivo utilizado)
  - `Parte4/`
    - `<archivo>` (Archivo utilizado para esta parte)
    - `firma.priv` (Clave privada utilizada)
    - `firma.pub` (Clave pública utilizada)
    - `<archivo>.hash` (Hash del archivo utilizado)
    - `<archivo>.hash.sig` (Hash del archivo firmado con la clave privada)
    - `<archivo>.zip` (Comprimido con el archivo original y el hash firmado)
    - `cifrado.key` (Archivo de texto con la contraseña utilizada para crear la clave de criptografía simétrica)
    - `<archivo>.zip.enc` (Comprimido encriptado)
