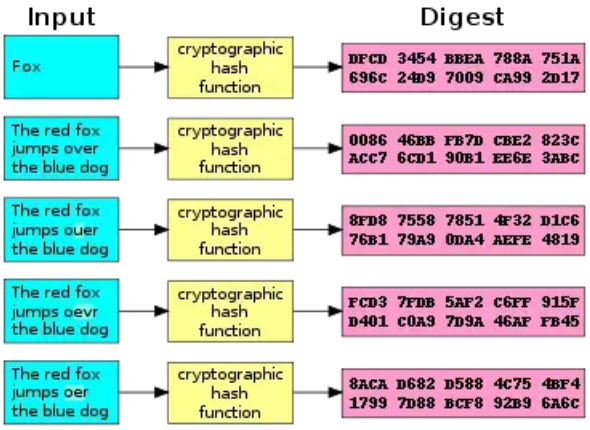
SAD - U2.3. Funciones Resumen

Descargar estos apuntes

Índice

1. Funciones Hash



Una **función hash criptográfica** es un tipo especializado de `función resumen` diseñada para su uso en diversas aplicaciones criptográficas, incluidas firmas digitales, códigos de autenticación de mensajes y otras formas de autenticación. Estas funciones desempeñan un papel crucial en las prácticas modernas de seguridad de la información, particularmente en protocolos como SSL/TLS.

> Es una función matemática unidireccional que opera sobre un documento digital, secuencia digital numérica, etc., todos de gran tamaño medido en bits, y brinda como resultado un valor más pequeño y de **tamaño fijo, cualquiera sea su entrada**, que se utiliza en aplicaciones criptográficas que protegen la integridad de la información.

Las funciones hash criptográficas poseen varias propiedades esenciales que las distinguen de otras funciones hash:

Propiedad	Descripción
Determinista	El mismo mensaje de entrada siempre produce el mismo valor hash.
Eficiencia	El valor hash se calcula rápidamente, independientemente del tamaño de entrada.
Resistencia a la colisión	Es computacionalmente inviable encontrar dos mensajes diferentes que produzcan el mismo valor hash. No es imposible, pero sí improbable.
Resistencia a la preimagen	Dado un valor hash, no es factible crear un mensaje que produzca ese hash específico.
Efecto avalancha / Difusión	Pequeños cambios en el mensaje de entrada dan como resultado cambios significativos y aparentemente no correlacionados en el hash de salida.

No es cifrado

Las funciones hash criptográficas **no son algoritmos de cifrado**, ya que no se pueden deshacer. Es decir, no se puede recuperar el mensaje original a partir del valor hash.

En cambio, se utilizan para verificar la integridad de los datos y garantizar que no se hayan modificado.

1.2 Aplicaciones de funciones hash criptográficas

Las funciones hash criptográficas tienen numerosas aplicaciones en ciberseguridad:

- Firmas digitales: se utiliza para crear un resumen de tamaño fijo de un mensaje, que luego se cifra con la clave privada del remitente.
- Verificación de la integridad del archivo: Los sitios web suelen publicar valores hash para archivos descargables, lo que permite a los usuarios verificar la integridad del archivo después de la descarga.
- Contraseña de Seguridad: Las contraseñas normalmente se almacenan como hashes en lugar de texto sin formato, lo que mejora la seguridad.
- Tecnología Blockchain: Las criptomonedas como Bitcoin utilizan funciones hash criptográficas (por ejemplo, SHA-256) para mantener la integridad y seguridad de los registros de transacciones.
- SSL /TLS Protocolos: Estos protocolos de comunicación seguros dependen en gran medida de funciones hash criptográficas para diversos mecanismos de seguridad.

1.3 Funciones hash más comunes

1.3.1. MD5 (Message Digest Algorithm 5)

MD5 es una función hash criptográfica que produce un valor hash de 128 bits. Aunque fue ampliamente utilizado en el pasado, MD5 se considera obsoleto y se ha demostrado que es vulnerable a colisiones, lo que significa que dos mensajes diferentes pueden producir el mismo valor hash.

1.3.2. SHA-1 (Secure Hash Algorithm 1)

SHA-1 es una función hash criptográfica que produce un valor hash de 160 bits. Al igual que MD5, SHA-1 se considera obsoleto y vulnerable a colisiones.

1.3.3. SHA-2 (Secure Hash Algorithm 256)

Una familia de funciones hash que producen resúmenes de varios tamaños: 224, 256, 384 o 512 bits.

SHA-256 (versión de 256 bits) es la variante más utilizada y produce una salida hexadecimal de 64 caracteres. Ampliamente adoptado en protocolos de seguridad como SSL/TLS.

1.3.4. SHA-3 (Secure Hash Algorithm 3)

SHA-3 es una función hash criptográfica que produce un valor hash de 224, 256, 384 o 512 bits. Es la última adición a la familia de funciones hash SHA y se considera seguro para su uso en aplicaciones criptográficas.

1.3.5 HMAC (Hash-based Message Authentication Code)

En criptografía, un **código de autenticación de mensaje (MAC)**, es una información breve `(resumen)` que se utiliza para autenticar un mensaje; en otras palabras, para confirmar que el mensaje proviene del remitente indicado (su `autenticidad`) y que el mensaje no haya sido alterado.

Los valores MAC se calculan mediante la aplicación de una función hash criptográfica con clave secreta K, que sólo conocen el remitente y destinatario, pero no los atacantes. **El valor MAC protege tanto la integridad de los datos de un**

mensaje como su autenticidad, al permitir que los verificadores (que también poseen la clave secreta) detecten cualquier cambio en el contenido del mensaje.

Los algoritmos MAC pueden construirse a partir de otras primitivas criptográficas, como las funciones criptográficas hash (como en el caso de HMAC).

El código de autenticación de mensaje basado en MAC `(HMAC)` es un algoritmo de autenticación de mensajes basado en funciones hash criptográficas. Combina un valor hash con una clave secreta para producir un código de autenticación de mensajes que se puede utilizar para verificar la **integridad y la autenticidad** de un mensaje.

HMAC se utiliza en una variedad de aplicaciones de seguridad, incluidos protocolos de autenticación como OAuth y TLS.

2. Funciones HASH con OpenSSL

OpenSSL es una biblioteca de software que implementa protocolos y algoritmos de cifrado seguros. Es ampliamente utilizada en aplicaciones de software para garantizar la seguridad de la información.

2.2. Uso de OpenSSL para funciones resumen

Para proceder a cifrar o descifrar mensajes se hará uso del comando `dgst`. Este comando permite calcular el valor hash de un mensaje utilizando una función hash específica.

El subcomando `dgst -list` lista los algoritmos simétricos disponibles en la instalación de OpenSSL.

```
$ openssl dgst -list
Supported digests:
-blake2b512               -blake2s256               -md4
-md5                      -md5-sha1                 -mdc2
-ripemd                   -ripemd160                -rmd160
-sha1                     -sha224                   -sha256
-sha3-224                 -sha3-256                 -sha3-384
-sha3-512                 -sha384                   -sha512
-sha512-224               -sha512-256               -shake128
-shake256                 -sm3                      -ssl3-md5
-ssl3-sha1                -whirlpool
```

En esta lista pueden verse algunos datos importantes: **algoritmos** de resumen disponibles (MD5, SHA1, SHA2, SHA3, Blake, RIPMD, SHAKE) y **longitud de bits** de los resúmenes (128, 160, 224, 256,512).

2.2.1. Cálculo de resúmenes

Para calcular el resumen de un mensaje, se utiliza el comando `dgst` seguido del algoritmo de resumen deseado y el archivo de entrada. Por ejemplo, para calcular el resumen SHA-256 de un archivo llamado `mensaje.txt`, se utiliza el siguiente comando:

```
$ echo "Hola, mundo" > mensaje.txt
$ openssl dgst -sha256 mensaje.txt
SHA256(mensaje.txt)= b54e884a20812eb4b4b9653b673eddbec0b0d6ca25d12849b3d40275e99a97cb
```

Podemos usar varios modificadores para personalizar la salida del comando `dgst`:

- `-hex` : Muestra el resumen en formato hexadecimal (opción por defecto si no se indica ninguna).
- - `-r` : Muestra el resumen en formato coreutils.
- `-c` : Muestra el resumen en formato `AA:BB:CC:.....`.
- `binary` : Muestra el resumen en formato binario.
- `-out archivo` : Guarda el resumen en un archivo.

2.2.2. Calculo de resúmenes HMAC

HMAC puede usarse para verificar simultáneamente la integridad de los datos y la autenticación de un mensaje, como con cualquier MAC. Cualquier función hash criptográfica, como SHA256 o SHA-3, puede ser utilizado en el cálculo de un HMAC; el algoritmo MAC resultante se denomina HMAC-X, donde X es la función hash utilizada (por ejemplo, HMAC-SHA256 o HMAC-SHA3).

Para calcular un resumen HMAC, se utiliza el comando `dgst` seguido del algoritmo de resumen deseado y la clave secreta. Por ejemplo, para calcular el resumen HMAC-SHA256 de un archivo llamado `mensaje.txt` con la clave secreta `secreto`, se utiliza el siguiente comando:

```
$ openssl dgst -sha256 -hmac "secreto" mensaje.txt
HMAC-SHA256(mensaje.txt)= e2207fafe0cc81dcfb8a648373851b1005148b4d9b226b6696d7fd243c80768a
```

Ahora podríamos enviar al destinatario el mensaje.txt (cifrado o no), y el HMAC generado. El destinatario podrá realizar el mismo procedimiento de cálculo del HMAC (sobre el archivo no cifrado), y comparar la salida con el que enviamos.

Si un atacante intercepta el mensaje, no podrá modificarlo ya que, al no poseer la clave de autenticación, no podrá regenerar un HMAC válido para que el destinatario verifique.

Así, si el destinatario calcula el HMAC y da correcto, implica que:

- El mensaje no fue modificado, verifica **integridad**.
- El HMAC únicamente pudo haber sido generado por el origen, que dispone de la clave de autenticación, por lo que también verifica **autenticidad**.
- Además, como se supone que el origen es el único que tiene la clave de autenticación, no puede desentenderse de haber firmado el mensaje, así que verifica **no repudio** también.