Criptografía

Andrés Martínez Mendieta

andres.martinez@wizeline.com

Acerca de mí

Andrés Martínez

Software Engineer III

- Ingeniero Mecatrónico
- 6 años de experiencia en IT
- 2 años en Wizeline
- Backend engineer

Puntos importantes

Identifícate en Zoom utilizando tu nombre y apellido.

Mantén tu micrófono apagado durante el transcurso de la sesión.

Utiliza el chat para hacer tus preguntas durante la sección de Q&A.

Procura enfocar tus preguntas al tema presentado.

Apaga tu cámara en caso de tener problemas con tu conexión.

Código de conducta

Sé respetuoso, no hay malas preguntas o ideas.

Sé cordial y paciente.

Sé cuidadoso con tus palabras.

Objetivo

Al final de esta sesión podrás:

- Saber la importancia de los sistemas de cifrado para mantener confidencialidad de un dato enviado o guardado.
- Saber la importancia de mantener los datos confiables e íntegros.
- Utilidad de la criptografía en el mundo IT actual.

Contenido

Enmascaramiento

Significado y herramientas en el mercado

Codificación

Significado y aplicaciones

Encriptación Simétrica

Explicación, casos de uso y desafíos

Encriptación Asimétrica

Explicación, historia y ventajas de uso

Algoritmos de digestión

Explicación, historia y ventajas de uso

Algo de historia

Enmascaramiento de Datos

Enmascaramiento

El primer enmascaramiento es el **testado de datos**, que se refiere a la **ocultación de información sensible**.

Aunque no es propiamente una medida de seguridad, es particularmente importante para proveer la información a algún administrador o usuario externo sin exponer totalmente la información sensible.

	Dato Real	Dato Enmascarado
Nombre	Esteban García	EstebanXXXXXXX
Fecha de Nacimiento	Mar-10-1970	XXX-XX-1970
Tarjeta de crédito	1234 5678 0000 1234	1234 56XX XXXX 1234

Algunos usos

- Cumplimiento normativo
  - Algunos lineamientos solicitan que la información sea enmascarada para poder utilizarla. Por ejemplo, PCI DSS pide utilizar cómo máximo los primeros 6 y últimos 4 dígitos de las tarjetas de crédito 1234 56XX XXXX 1234
- Troubleshooting
  - Hay ocasiones en que se tienen que hacer pruebas en ambientes productivos para encontrar un error, sin embargo es importante que la información personal sensible de los clientes esté enmascarada
- Auditorías
  - Muy comúnmente se tienen que compartir capturas de pantalla de los sistemas a los auditores, por lo cual es necesario contar con una medida de enmascaramiento para poder compartir las capturas sin exponer datos personales de nuestros clientes

Algunos usos

- Cumplimiento con las fuerzas de la ley y el orden
  - Podrá existir la posibilidad que una autoridad de algún país o entidad nos solicite información de nuestras bases de datos para la investigación de algún delito. Es muy importante poder compartir la información sin exponer más de la requerida por la autoridad, por eso es necesario poder enmascarar la información.
- Investigaciones
  - Al igual que en el caso anterior, podrán existir investigaciones internas o externas que requieran información de nuestras bases de datos. Es muy importante poder compartir la información sin exponer más de la autorizada para compartir.

Enmascaramiento

El segundo tipo de enmascaramiento de datos se refiere a el **reemplazo de información sensible** con información ficticia muy similar a la real.

Esto es particularmente importante cuando se requiere un ambiente de pruebas con información lo más real posible. Antes de serle entregada, se recomienda hacer un enmascaramiento a los datos.

	Dato Real	Dato Enmascarado
Nombre	Esteban	Roberto
Fecha de Nacimiento	Mar-10-1970	Feb-21-1971
Email	esteban@domain.com	esteban@domain.test

Enmascaramiento

Con un buen enmascaramiento, no debe ser notorio que los datos son ficticios.

Esto disminuye mucho la superficie de ataque y ayuda a que si un intruso accede al sistema, al conseguir estos datos enmascarados piense que pudo lograr lo que quería. Algunas herramientos y tecnologías para enmascarar datos son:

- Apache Atlas
- BizDataX
- Oracle Advanced Security
- Satori Data Security

Codificación de Datos

Codificación

La codificación es todo tipo de **transformación en los datos** siguiendo una secuencia de pasos conocida, los cuales pueden ser aplicados de manera inversa para transformar los datos a su estado original. Es muy importante contemplar que no oculta la información

Un ejemplo claro es la codificación de URLs, donde caracteres especiales tienen su correspondiente representación en un mismo formato:

Caracter	Codificación de URL	Codificación en Base64
$	%24	JA==
&	%26	Jg==
?	%3F	Pw==
espacio	%20	IA==

Algunos usos

- JSON
  - Dado que hay algunos caracteres que pudieran deformar una consulta JSON, la codificación puede ser muy útil para preservar la cadena original sin utilizar caracteres riesgosos
- Canales de comunicación con caracteres limitados
  - Hay algunos canales de comunicación donde no se pueden utilizar ciertos caracteres como los latinos, cirílicos y demás. Para esto es muy útil codificar las cadenas sin correr el riesgo de perder la integridad de la cadena.
- Cookies
  - Al igual que en los casos anteriores, no cualquier caracter se puede almacenar en las cookies, por lo que es de mucho valor poder almacenar una cadena codificada.

Encriptación Simétrica

Encriptación simétrica

Documento plano con información sensible

+

Llave de encriptación

Documento **cifrado**

A través de algún algoritmo de encriptación, se usa una llave para transformar el contenido a una representación completamente diferente a la original.

Encriptación simétrica

Documento **cifrado**

+

Llave de encriptación

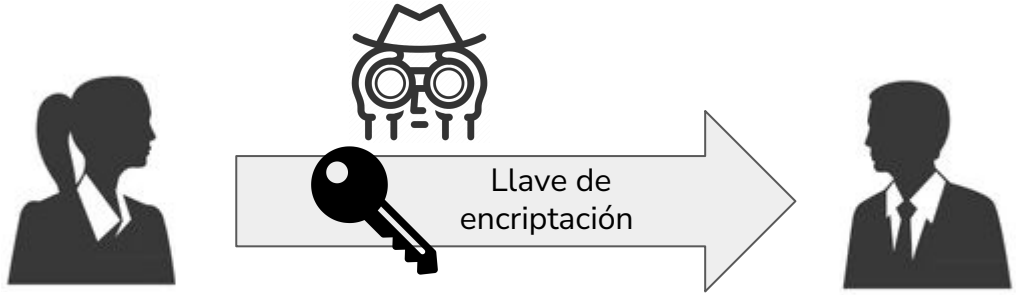
Documento **descifrado**

La única manera de transformar este contenido a su representación original es utilizando **la misma llave** de encriptación con la que se transformó previamente.

Encriptación simétrica

Para que esta metodología de encriptación sea segura, tanto el emisor como el receptor del contenido deben **conocer con anterioridad** la llave de encriptación.

De otra manera, surge el desafío de tener que **compartir esa llave** entre ellos antes de enviar el contenido encriptado...



Ejemplo de uso (archivos comprimidos)

- Compresión de Datos: Primero, los datos se comprimen utilizando algoritmos de compresión como ZIP o RAR. Esto reduce el tamaño del archivo original.
- Generación de Clave: Se genera una clave de encriptación única y se comparte de manera segura entre el remitente y el receptor.
- Cifrado de Datos: Utilizando la clave generada, los datos comprimidos se cifran, lo que los convierte en ilegibles sin la clave.
- Almacenamiento o Transmisión: El archivo comprimido y cifrado se almacena o transmite de manera segura.
- Recuperación: Al ser la misma llave de cifrado y descifrado, se puede recuperar la información en cualquier momento

Encriptación Asimétrica

Encriptación asimétrica

A diferencia de la encriptación simétrica, cada sujeto cuenta con **un par de llaves**. Las llaves privadas **nunca son compartidas**, mientras que las llaves públicas pueden ser usadas por cualquiera que desee enviar un mensaje protegido a esa persona.

Llave **pública**

Llave **privada**

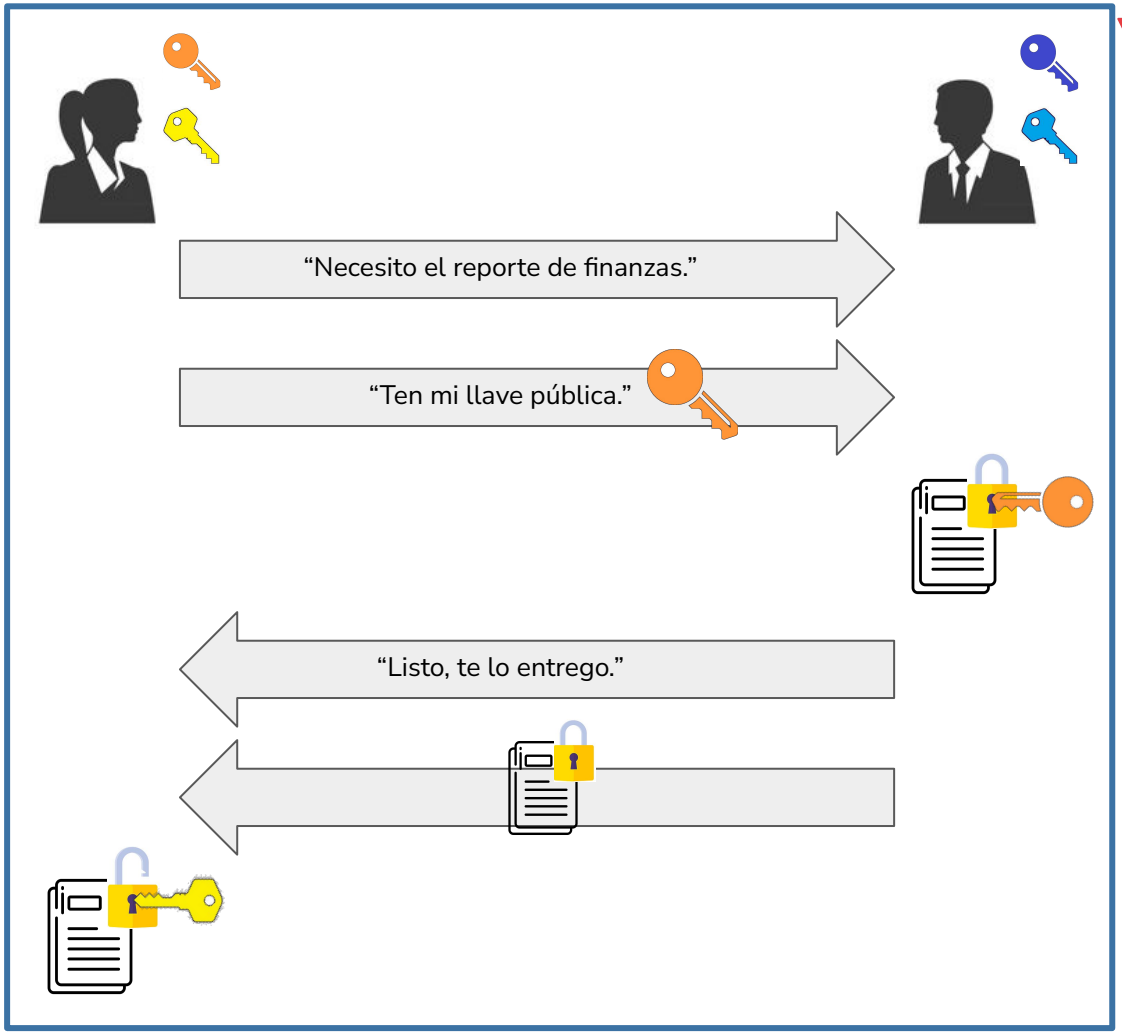
Llave **pública**

Llave **privada**

Encriptación asimétrica

Cada par de llaves pública y privada están **asociadas matemáticamente**.

Todo lo cifrado con una llave pública solo puede ser descifrado con su correspondiente llave privada.



Encriptación asimétrica

El algoritmo de encriptación asimétrica RSA es uno de los más usados para lograr este objetivo. Tiene ese nombre por las 3 personas que lo desarrollaron en 1977: ***R****on Rivest*, *Adi* ***S****hamir* y *Leonard* ***A****dleman*.

Actualmente continúa siendo usado como mecanismo de protección en muchas tecnologías:

- SSH
- HTTPS
- Bitcoin
- PGP / GPG
- Y muchas otras...

Ejemplo de uso (HTTPS)

- El servidor comparte una llave pública con el usuario
- La única forma de conocer la información que envía el usuario, es con la llave privada
- Al solo el servidor tener la llave privada, se garantiza la autenticidad del servidor, pues nadie más podría enviar información descifrable por la llave pública ni descifrar la información recibida.
- Nadie podría modificar los paquetes, ya que el receptor no podría descifrar la información correctamente

Algoritmos de digestión

Algoritmos de digestión

A diferencia de los algoritmos de encriptación y ofuscación, el objetivo de los algoritmos de digestión o hash es que **el resultado nunca se pueda recuperar**. Esto se logra al utilizar funciones matemáticas para convertir cualquier cosa en una cadena hexadecimal con un límite definido. Por ejemplo, SHA256 siempre brindará una cadena con longitud de 64 caracteres (32 hexadecimales)

→ **Algoritmo de digestión** →

b221d9dbb083a7f33428d7
c2a3c3198ae925614d7021
0e28716ccaa7cd4ddb79

Problemas de colisión

La fortaleza de un algoritmo de digestión se determina por qué tan improbable es que dos cadenas cuenten con exactamente el mismo hash (colisión).
Por ejemplo, MD5 se considera un hash débil por tener una probabilidad “considerable” de colisión (1 en 3.4 x (10^38))

```
C:\TEMP> md5sum hello.exe
cdc47d670159eef60916ca03a9d4a007
C:\TEMP> .\hello.exe
Hello, world!

(press enter to quit)
C:\TEMP>
```

```
C:\TEMP> md5sum erase.exe
cdc47d670159eef60916ca03a9d4a007
C:\TEMP> .\erase.exe
This program is evil!!!
Erasing hard drive...1Gb...2Gb... just kidding!
Nothing was erased.

(press enter to quit)
C:\TEMP>
```

Eduardo Diaz,
Exploiting MD5 collisions, 2005

Algunos usos

- Contraseñas
  - Dado que no debería haber interés en conocer las contraseñas de tus usuarios, es recomendable almacenarlas como hashes, de tal forma que solo el usuario las conozca.
- Integridad
  - Es posible validar la integridad de un archivo, formulario o cualquier activo informático al conocer el hash del activo original y calcular el hash del activo recibido para garantizar que sea el mismo
- Malware
  - Los antivirus cuentan con una base de datos de archivos maliciosos y sus hashes, de tal forma que pueden detectar rápidamente la presencia de malware en las computadoras solo analizando los hashes de todos los archivos sin tener que validar su contenido

Conclusiones y preguntas

Gracias.