Hqfulswdflrq

📝 2024-03 — ⏱️ 60 min.

Encriptación

y sensitive data

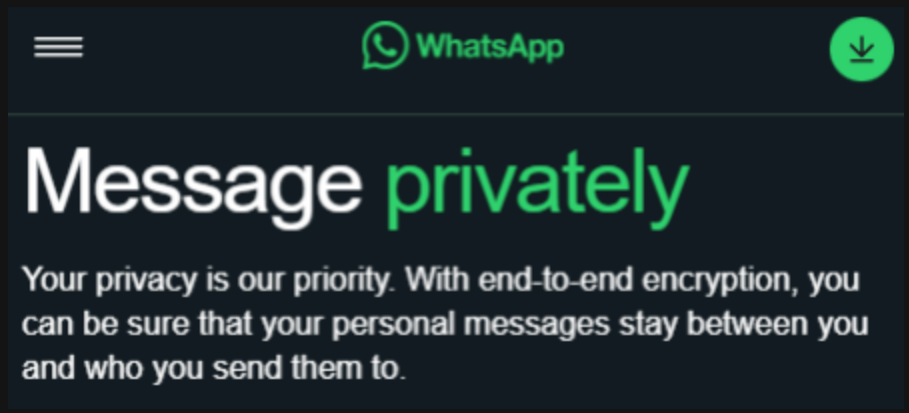
Que es?

El proceso de codificar informacion por razones de seguridad.

Porque nos interesa?

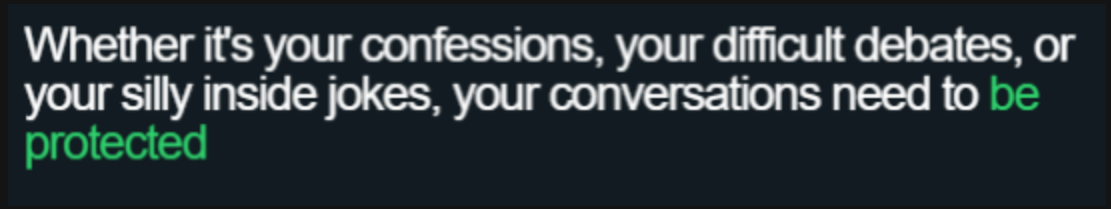
1. Privacidad
2. Seguridad
3. Integridad de Datos
4. Regulaciones

Porque nos interesa? - Privacidad



Mensajes, informacion personal, etc.

Porque nos interesa? - Seguridad



En los mensajes puede haber informacion sensible o de seguridad.

"Como es que es la contraseña?"

Porque nos interesa? - Seguridad

Y claro, de por si esa informacion sensible o de seguridad debería estar encriptada al ser almacenada.

Porque nos interesa? - Integridad de Datos

El encriptado de información asegura que los datos no fueron *vistos* ni *modificados* en el camino a su destino.

Porque nos interesa? - Regulaciones

Existen regulaciones gubernamentales y de la industria.

Como funciona?

1. Informacion a encriptar
2. Proceso de encriptado
3. Informacion encriptada
4. Proceso de desencriptado
5. Informacion desencriptada

Por ejemplo,

Paso	Encriptado?
Anita le escribe un mensaje a Juan.	
Whatsapp (de Anita) envía el mensaje al servidor.	
El servidor envía el mensaje a Juan.	
Whatsapp (de Juan) recibe el mensaje.	
Juan lee el mensaje.	

Por ejemplo,

Paso	Encriptado?
Anita le escribe un mensaje a Juan.	No
Whatsapp (de Anita) envía el mensaje al servidor.	
El servidor envía el mensaje a Juan.	
Whatsapp (de Juan) recibe el mensaje.	
Juan lee el mensaje.	

Por ejemplo,

Paso	Encriptado?
Anita le escribe un mensaje a Juan.	No
Whatsapp (de Anita) envía el mensaje al servidor.	Si
El servidor envía el mensaje a Juan.	
Whatsapp (de Juan) recibe el mensaje.	
Juan lee el mensaje.	

Por ejemplo,

Paso	Encriptado?
Anita le escribe un mensaje a Juan.	No
Whatsapp (de Anita) envía el mensaje al servidor.	Si
El servidor envía el mensaje a Juan.	Si
Whatsapp (de Juan) recibe el mensaje.	
Juan lee el mensaje.	

Por ejemplo,

Paso	Encriptado?
Anita le escribe un mensaje a Juan.	No
Whatsapp (de Anita) envía el mensaje al servidor.	Si
El servidor envía el mensaje a Juan.	Si
Whatsapp (de Juan) recibe el mensaje.	No
Juan lee el mensaje.	

Por ejemplo,

Paso	Encriptado?
Anita le escribe un mensaje a Juan.	No
Whatsapp (de Anita) envía el mensaje al servidor.	Si
El servidor envía el mensaje a Juan.	Si
Whatsapp (de Juan) recibe el mensaje.	No
Juan lee el mensaje.	No

Como funciona?

Se toma el texto original, llamado "texto plano" o "plaintext", se usa un algoritmo de cifrado con una llave y este retorna el texto cifrado, o "cyphertext".

El proceso de desencriptado es del mismo modo, en sentido contrario.

Tipos de encriptación

1. Asimetrica
2. Simetrica

Encriptación Asimetrica

Se tiene una llave publica, y una llave privada.

La llave publica se utiliza para encriptar.
La llave privada se utiliza para desencriptar.

Ventaja: "No pasa nada" si la llave publica se "filtra".

Desventaja: Es mas lento que el encriptado simetrico.
Desventaja: Requiere mas poder computacional.

Encriptación Simetrica

Se usa una sola llave privada, compartida en ambos lados.

Esa llave se usa para encriptar y desencriptar.

Ventaja: Es mas rapido que el encriptado asimetrico.
Ventaja: Requiere poco poder computacional.

Desventaja: Como distribuyes la llave de manera segura?
Desventaja: Para cada par de usuarios unicos, necesitas una llave.

Encriptación Hibrida

Se utiliza encriptación asimetrica para enviar la llave de la encripación simetrica.

Para el intercambio de información, se utiliza la encriptación simetrica.

Password Hashing

y sensitive data

Hashing vs Encriptacion

- Ambos son metodos de proteccion de datos.
- La encriptacion es un proceso reversible.
- El hashing no es reversible.

Encriptacion para contraseñas

Si se usa encriptacion para proteger contraseñas, estas son vulnerables.

Con la llave, se desencripta la contraseña.

Hashing para contraseñas

Cual llave? No hay llave.

"No sé cual es la contraseña, pero esa no es."

Como funciona la encriptacion?

Se toma el texto original, llamado "texto plano" o "plaintext", se usa un algoritmo de cifrado con una llave y este retorna el texto cifrado, o "cyphertext".

El proceso de desencriptado es del mismo modo, en sentido contrario.

Como funciona el hashing?

Se toma el texto original, llamado "texto plano" o "plaintext", se usa un algoritmo de hashing y este retorna el hash.

No hay manera real de saber que texto original era el hash.

Pero para todo texto plano hay un y solo un hash.

Como funciona el hashing?

"Pero para todo texto plano hay un y solo un hash."

Esto significa que aunque no sepas que texto era un hash, si te dicen un texto tu puedes verificar si ese texto es o no el hash.

Proceso de hashing

Por esto, se considera el hashing una mejor medida de seguridad para las contraseñas y datos similares.

Si no se utiliza hashing, cualquier persona con acceso a la base de datos tiene acceso a estos datos.

Contraseñas iguales

Que pasa si dos usuarios tienen la misma contraseña?

Si la contraseña de uno de los dos usuarios es comprometida, ambos usuarios se ven afectados.

Salt

Salt es una string generada aleatoriamente que es agregada a un texto previo al proceso de hash.

La salt se puede almacenar junto al password sin ser encriptada ni nada, asi en plaintext.

Salt

Si decifran la contraseña de un usuario, y ven el resultado del hash en la DB, no encontrarán (en teoría) otro hash igual, indiferente de si es o no la misma contraseña.

Es decir, protege ante **brute-force attacks**.

Salt

Tambien protege ante **rainbow tables**, que son tablas precomputadas de cualquier texto posible ante un proceso de hashing especifico.

Pepper

Pepper es un amiguito de Salt.

Pepper funciona igual que Salt, pero no está almacenado en la base de datos, y es la misma para todos.

Podría estar, por ejemplo, en el codigo (si, quemado).

Salt + Pepper

El uso de ambos significa que un atacante requeriría tanto acceso a la base de datos, como acceso al codigo fuente de la aplicación.

Algoritmos de Hashing de Contraseñas

Según la Competencia de Hashing de Contraseñas, el mejor algoritmo actualmente, y desde el 2017, es:

Argon2

Previamente, era bcrypt.