

Usando Criptografía

[Descargar este caso](#)

Índice

- [Objetivos](#)
- ▼ [Descripción de la tarea](#)
 - [Cifrado simétrico](#)
 - [Cifrado asimétrico](#)
 - [Funciones hash](#)
 - [Firma digital](#)
 - [Crea tu certificado digital](#)
 - [Certificados digitales web](#)
 - [Análisis de un protocolo de cifrado](#)
 - [Análisis de un caso real](#)
 - [Firma](#)
- [Entrega](#)
- [Criterios de evaluación](#)
- [Herramientas de evaluación](#)
- ▼ [Anexo 1: Cifrado y descifrado de documentos – GnuPG](#)
 - [Cifrado simétrico](#)
 - ▼ [Cifrado asimétrico](#)
 - [Generación de un nuevo par de claves](#)
 - [Listar las claves públicas y privadas](#)
 - [Exportar una clave pública](#)
 - [Exportar una clave privada](#)
 - [Importar una clave pública](#)
 - [Cifrado de documentos](#)
 - [Descifrado de documentos](#)
 - [Distribución de claves](#)
 - [Generación de un certificado de revocación](#)
 - [Referencias](#)
- ▼ [Anexo 2: Creación y verificación de firmas – GnuPG](#)
 - [Firmar digitalmente](#)
 - [Verificar un documento firmado](#)

- Verificar un documento firmado y extraer su contenido
- Documentos con firma en claro
- Firmas separadas
- Referencias

▼ Anexo 3: Creación de un certificado digital PKCS#12 con OpenSSL

- 1. Genera un par de claves RSA (pública y privada)
- 2. Genera el certificado autofirmado
- 3. Convierte el certificado a formato PKCS#12

Objetivos

- Comprender los conceptos básicos del cifrado de datos y aplicarlos.
- Comprender cómo los certificados y las firmas digitales garantizan la autenticidad, integridad y no repudio de las comunicaciones digitales aplicándolos.

Descripción de la tarea

Cifrado simétrico

Usa el comando gpg (de GnuPG) para cifrar y descifrar un mensaje de forma simétrica

[Anexo 1: Cifrado y descifrado de documentos – GnuPG](#)

1. Crea un nuevo documento que contenga algún texto y nombrarlo `simetrico` (sin extensión).
2. Cifra `simetrico`.
3. Intenta abrir `simetrico` ¿Puedes leer el texto?
4. Intenta abrir `simetrico.gpg` ¿Puedes leer el texto? ¿Quién ha creado este fichero con extensión `gpg` ?
5. Descifra `simetrico.gpg`
6. Explica cómo y cuándo has empleado la contraseña de cifrado/descifrado.

Cifrado asimétrico

Usa el comando gpg (de GnuPG) para cifrar y descifrar de forma asimétrica.

[Anexo 1: Cifrado y descifrado de documentos – GnuPG](#)

1. Genera un par de claves (privada y pública)
2. Lista las claves públicas y privadas que hay en tu anillo de claves.
3. Exporta la clave pública a un archivo en formato ASCII y nombrarlo `tunombre.pub`.
4. Realiza una copia de seguridad de la clave privada y guárdala como `tunombre.priv` (debe mantenerse segura y no compartirse).
5. Sube tu clave pública al foro de Aules (`tunombre.pub`).
6. Descarga la clave pública de un compañero, importarla y verifica que la importación es correcta.
7. Crea un archivo con algún texto y cifrarlo para el compañero, nombrándolo `PARacompaneroDEtunombre`.
8. Sube el fichero cifrado al foro de Aules.
9. Descarga los mensajes recibidos de compañeros y descifrarlos.

10. Almacena tu clave pública en un servidor de claves.
11. Sospechas que tu clave privada ha sido comprometida, revócala:
 - a. Crea un certificado de revocación.
 - b. Importa el certificado de revocación que has generado.
 - c. Lista primero tu anillo de claves públicas y luego el de claves privadas, y comprueba qué clave ha sido revocada.
 - d. Una clave pública revocada no puede utilizarse para cifrar mensajes, aunque sí puede emplearse para descifrar aquellos que se hayan generado antes de la fecha de revocación ¿Qué ocurrirá si intentas cifrar un archivo con una clave pública revocada?

Funciones hash

1. Calcula las sumas MD5 (`md5sum` en Linux) de dos documentos y compara los valores obtenidos.
2. Analiza la diferencia en los valores hash de documentos que solo difieren en una letra.

Firma digital

Anexo 2: Creación y verificación de firmas – GnuPG

Usa el comando `gpg` (de GnuPG) para generar y verificar una firma digital

1. Crea un archivo de prueba `documento` .
2. Genera un nuevo par de claves asimétricas.
3. Firma `documento` y llama al resultado `firmado` .
4. Verifica `firmado` .
5. Verifica `firmado` y extrae su contenido.
6. ¿Cuándo deberíamos usar el parámetro `--clearsign` ? ¿Cómo se emplea?
7. ¿Qué ocurre si ejecutamos el siguiente comando `gpg --detach-sign -a documento` ? ¿Cuándo crees que se debe usar?
8. Verifica el fichero `documento.asc` .

Crea tu certificado digital

Anexo 3: Creación de un certificado digital PKCS#12 con OpenSSL

Crea un certificado autofirmado con OpenSSL y consérvalo en un lugar seguro, lo utilizaremos en otras unidades.

Certificados digitales web

Examina el certificado digital utilizado por cualquier sitio web seguro (*https*) y responde a las siguientes preguntas.

1. ¿Dónde podemos encontrar...
 - i. la clave pública almacenada?
 - ii. la firma digital de la autoridad de certificación (CA)?
 - iii. el periodo de validez del certificado digital?
 - iv. un hash del certificado?
2. ¿Cómo podemos averiguar...
 - i. quién es el propietario del certificado digital?
 - ii. cuál es la Autoridad de Certificación (CA)?

Análisis de un protocolo de cifrado

El protocolo de mensajería de WhatsApp es un ejemplo práctico de criptografía híbrida. Explica el siguiente texto con tus propias palabras.

WhatsApp utiliza criptografía híbrida para proteger los mensajes. Esto significa que combina dos tipos de cifrado: asimétrico y simétrico. Primero, emplea un algoritmo asimétrico llamado Elliptic Curve Diffie-Hellman (ECDH) para que los usuarios puedan establecer una clave secreta compartida, sin necesidad de enviarla directamente por la red.

Una vez que se ha intercambiado esa clave, se utiliza un cifrado simétrico, concretamente AES, para cifrar los mensajes. Este tipo de cifrado es mucho más rápido y eficiente para el intercambio continuo de datos.

Además del cifrado, WhatsApp garantiza la integridad y autenticidad de los mensajes usando HMAC-SHA256, un código de autenticación basado en hash. Esto sirve para verificar que los mensajes no han sido modificados durante la transmisión.

Por último, también se implementa forward secrecy o secreto perfecto hacia adelante. Esto significa que las claves simétricas se cambian con frecuencia. Así, si alguien consiguiera acceder a una de las claves de sesión, no podría descifrar mensajes anteriores, ya que cada sesión utiliza claves diferentes.

Análisis de un caso real



"Este es el motivo por el que WhatsApp y Google no han podido recuperar los mensajes de García Ortiz"

Publicado el 5 de mayo de 2025

Por Marta Gascón

Fuente: 20minutos.es

Ni el contenido de los chats ni los correos electrónicos han servido como prueba. El motivo está en cómo funcionan el cifrado, las copias de seguridad y las políticas de eliminación de datos.

Contexto del caso

La investigación del Tribunal Supremo de España contra el fiscal general del Estado, Álvaro García Ortiz, por una presunta revelación de secretos ha llegado a un punto crítico tras la revisión de los datos enviados por Google y WhatsApp. La Guardia Civil, encargada del análisis pericial, ha concluido que no se ha podido recuperar información relevante para el caso. Las tecnológicas han cumplido con los requerimientos judiciales, pero los correos y mensajes que podrían haber contenido pruebas clave ya no existen.

A finales de 2023, el juez del Supremo Ángel Hurtado abrió diligencias para investigar si García Ortiz filtró a la prensa datos confidenciales del expediente fiscal de Alberto González Amador, pareja de la presidenta madrileña Isabel Díaz Ayuso. Tras ser imputado, el fiscal general eliminó ese mismo día su cuenta de Google y los mensajes de WhatsApp, alegando motivos de seguridad personal.

Debido a la posibilidad de que en esos correos y chats se encontraran pruebas de la filtración, el juez ordenó que se solicitaran los datos directamente a Google y WhatsApp. El objetivo: recuperar las conversaciones, mensajes y correos electrónicos que pudieran esclarecer el origen de la información publicada sobre González Amador. Sin embargo, como decimos, los informes recibidos por el juzgado revelan que la recuperación ha sido infructuosa, lo que deja a Hurtado ante dos opciones: archivar el caso o proceder a juicio.

La decisión final se tomará después del 23 de mayo, fecha en la que González Amador está citado a declarar como testigo.

Por qué no se pueden recuperar los mensajes de WhatsApp

WhatsApp utiliza un sistema de cifrado de extremo a extremo que impide que la propia compañía acceda al contenido de los mensajes. Solo los dispositivos del emisor y del receptor pueden descifrarlos. Esto significa que, incluso si un juez solicita los mensajes, la plataforma de mensajería no puede leer ni entregar su contenido.

El servicio de Meta es claro en sus términos y condiciones de uso: “WhatsApp no divulga el contenido de los mensajes de sus usuarios en respuesta a solicitudes gubernamentales, y tampoco puede hacerlo”.

Además, si un usuario elimina una conversación y no tiene una copia de seguridad activa en Google Drive (o iCloud, en el caso de iPhone), esos mensajes se pierden de forma definitiva.

En el caso del fiscal general, según consta en el informe, los mensajes fueron borrados voluntariamente tras la imputación. Sin acceso físico al dispositivo original ni respaldo disponible, reconstruir esas conversaciones no es técnicamente posible.

Esto deja una explicación muy sencilla: WhatsApp no accede a tus fotos ni a tus mensajes, por lo que no puede entregarlas porque no las tiene.

Lo que sí puede facilitar WhatsApp, y de hecho facilita en investigaciones, son los famosos metadatos: números de teléfono implicados, fechas, horas de envío o recepción, duración de llamadas o tamaño de los archivos —lo que podría indicar si estamos ante una imagen o un vídeo, por ejemplo—.

Sin embargo, según el informe de la Guardia Civil, ni siquiera los metadatos han aportado información útil.

¿Y las copias de seguridad? Ahí podría haber estado la clave. WhatsApp permite a los usuarios guardar sus conversaciones en la nube (en Google Drive o iCloud), pero esa función tiene que estar activada manualmente y puede borrarse en cualquier momento. Si no hay copia, o si fue eliminada, los mensajes desaparecen para siempre. Y eso parece ser lo que ocurrió.

Por qué no se pueden recuperar los correos y datos de Google

La situación con Google es distinta, pero el resultado ha sido el mismo: los datos no se han podido recuperar. El fiscal general eliminó su cuenta de Google justo el día que fue imputado, lo que activó el proceso automático de eliminación definitiva que aplica la compañía.

Google mantiene los datos de una cuenta borrada durante un periodo limitado, que según su web de soporte son 20 días. Pasado ese plazo, la información asociada —correos, documentos, archivos de Drive, historial, etc.— se elimina de forma permanente y ya no puede recuperarse, ni siquiera ante una orden judicial.

En este caso, cuando las autoridades solicitaron a Google el acceso a los datos, la cuenta ya estaba completamente eliminada. Se pudieron rescatar 159 correos electrónicos, pero solo uno estaba relacionado con la causa, y era un mensaje que ya conocía la investigación.

Situación actual de la investigación

La Guardia Civil concluyó que no se ha podido recuperar información relevante para el caso. Las tecnológicas han cumplido con los requerimientos judiciales, pero los correos y mensajes que podrían haber contenido pruebas clave ya no existen.

El juez Ángel Hurtado se enfrenta ahora a dos opciones: archivar el caso o proceder a juicio. La decisión final se tomará después del 23 de mayo, fecha en la que González Amador está citado a declarar como testigo.

Para más detalles, puedes consultar el artículo original en 20minutos.es

Tras leer esta noticia, relaciona el contenido del artículo con lo que sabes sobre el método de cifrado de Whatsapp y la criptografía asimétrica:

- Explica de una manera técnica y sencilla, empleando los conocimientos impartidos en esta unidad, por qué la información de Google podría haber sido recuperada si no se hubiese eliminado pero la de Whatsapp no.
- Averigua dónde almacena whatsapp la clave a empleada para descifrar la información transmitida y por qué esto es fundamental para garantizar la privacidad y seguridad del contenido de los mensajes.

Firma

Adobe - Firmas basadas en certificados

Añade tu firma digital a los ejercicios antes de subirlos a Aules. Utiliza tu certificado generado con OpenSSL.

Entrega

Sube tu archivo de respuestas a Aules.

Criterios de evaluación

Se han utilizado sistemas de identificación como las firmas electrónicas y los certificados digitales.

Herramientas de evaluación

- **Cuestionario (70%):** preguntas de completar, verdadero/falso, opción múltiple.
- **Documento de respuestas (30%):** respuestas del estudiante entregadas en formato de texto.
-

Anexo 1: Cifrado y descifrado de documentos – GnuPG

Cifrado simétrico

Un documento puede ser cifrado con un algoritmo simétrico utilizando la opción `--symmetric` (`-c`).

```
$ gpg --output archivoCifrado.gpg -c archivo
```

Para descifrar un mensaje, se usa la opción `--decrypt` (`-d`):

```
# Opción 1
$ gpg --decrypt archivoCifrado.gpg

# Opción 2
$ gpg -d archivoCifrado.gpg

# Opción 3
$ gpg -o archivo -d archivoCifrado.gpg
```

Cifrado asimétrico

Generación de un nuevo par de claves

La opción de línea de comandos `--gen-key` se usa para crear un nuevo par de claves.

```
$ gpg --gen-key
```

Listar las claves públicas y privadas

Para listar las claves públicas de tu anillo de claves, usa la opción `--list-keys` (`-k`).

```
# Opción 1
$ gpg --list-keys

# Opción 2
$ gpg -k
```

Para listar las claves privadas en tu anillo de claves, usa la opción `--list-secret-keys` (`-K`).

```
# Opción 1
$ gpg --list-secret-keys

# Opción 2
$ gpg -K
```

Exportar una clave pública

Para enviar tu clave pública, primero debes exportarla empleando `--export`. Debes proporcionar el identificador de la clave a exportar.

```
$ gpg --output clavePublica.key --export idClave
```

La clave se exporta en formato binario, lo cual puede convertirse en inconveniente si debe enviarse por correo electrónico o publicarse en una página web.

GnuPG admite la opción `--armor` (`-a`) que genera la salida en un formato ASCII similar a los documentos sin codificar.

```
$ gpg --armor --export idClave
```

Exportar una clave privada

En la mayoría de los casos, la clave privada no necesita ser exportada y **no debe ser distribuida**. Si es necesario, ejecuta el siguiente comando para exportarla:

```
# Opción 1
$ gpg --export-secret-keys idClave > clavePrivada.key

# Opción 2 (en formato ASCII)
$ gpg --armor --export-secret-keys idClave > clavePrivada.key
```

Importar una clave pública

Puedes añadir una clave pública a tu anillo de claves con la opción `--import`.

```
$ gpg --import clavePublica.key
```

Cifrado de documentos

Para cifrar un documento, se usa la opción `--encrypt` (`-e`).

Debes tener las claves públicas de los destinatarios.

El resultado cifrado se coloca en la salida estándar o en el archivo especificado con la opción `--output`.

```
# Opción 1
$ gpg --output archivoCifrado.gpg --encrypt --recipient idClaveDestinatario archivo

# Opción 2
$ gpg -o archivoCifrado.gpg -e -r idClaveDestinatario archivo
```

Descifrado de documentos

Para descifrar un mensaje, se usa la opción `--decrypt` (`-d`).

Necesitas la clave privada correspondiente a la clave pública con la que se cifró el mensaje.

```
$ gpg -d archivoCifrado.gpg
```

Distribución de claves

Cuando el número de involucrados en la comunicación es bajo, las claves suelen distribuirse por correo electrónico.

Los servidores de claves públicas se usan para recopilar y distribuir de forma masiva las claves públicas.

```
$ gpg --send-keys --keyserver pgp.rediris.es idClave
```

Generación de un certificado de revocación

Después de crear tu par de claves, debes generar inmediatamente un certificado de revocación para la clave utilizando la opción `--gen-revoke`.

```
$ gpg --output revocar.asc --gen-revoke idClave
```

Referencias

[Manual de GnuPG](#)

[Guía de cifrado de GnuPG](#)

Anexo 2: Creación y verificación de firmas – GnuPG

Una firma digital certifica y fecha un documento, garantizando su integridad desde el momento en que fue firmado. Si el contenido del documento se modifica posteriormente de cualquier forma, la verificación de la firma fallará. Gracias a esta propiedad, una firma digital puede cumplir la misma función que una firma manuscrita, con la ventaja añadida de ser resistente a manipulaciones.

Un ejemplo práctico de esta tecnología lo encontramos en la distribución del código fuente de GnuPG, que se firma digitalmente para que los usuarios puedan comprobar que no ha sido alterado desde que fue empaquetado.

El proceso de creación y verificación de firmas digitales se basa en el uso de un par de claves pública y privada, pero sigue un procedimiento distinto al del cifrado y descifrado. La firma se genera utilizando la clave privada del firmante, mientras que su verificación se lleva a cabo con la clave pública correspondiente.

Como consecuencia, el uso de firmas digitales también tiene un componente de responsabilidad: resulta difícil negar la autoría de una firma, ya que hacerlo implicaría que la clave privada ha sido comprometida.

Firmar digitalmente

La opción de línea de comandos `--sign` se usa para crear una firma digital. Se introduce el documento a firmar (*doc*) y se genera el documento firmado como salida (*doc.sig*).

```
$ gpg -u keyID --output doc.sig --sign doc
```

El documento se comprime antes de ser firmado y la salida está en formato binario. Dado un documento firmado, puedes bien verificar la firma, bien verificarla y recuperar el documento original.

Verificar un documento firmado

Para verificar la firma, usa la opción `--verify`:

```
$ gpg --verify doc.sig
```

Verificar un documento firmado y extraer su contenido

Para verificar la firma y extraer el documento, usa la opción `--decrypt`. Se introduce el documento firmado (*doc.sig*) y se obtiene el documento recuperado como salida (*doc*).

```
$ gpg --output doc --decrypt doc.sig
```

Documentos con firma en claro

Un uso común de las firmas digitales es firmar mensajes de correo electrónico. En tales situaciones, no es deseable comprimir el documento mientras se firma. La opción `--clearsign` hace que el documento quede envuelto en una firma en formato ASCII sin modificarlo.

```
$ gpg -u keyID --clearsign doc
```

Firmas separadas

Un documento firmado digitalmente tiene una utilidad limitada si no se gestiona adecuadamente. Para verificar su contenido o reutilizarlo, otros usuarios deben recuperar el documento original a partir de la versión firmada. Incluso en el caso de firmas en claro, suele ser necesario editar el documento para extraer el contenido original.

Por este motivo, existe un tercer método de firma que resulta más práctico: la firma separada (detached signature), que genera un archivo independiente con la firma digital, manteniendo el documento original intacto y sin modificar.

Una firma separada se crea usando la opción `--detach-sig`.

```
$ gpg --detach-sign -a doc
```

Tanto el documento (*doc*) como la firma separada (*doc.sig*) son necesarios para verificar la firma. La opción `--verify` se usa para comprobar la firma.

```
$ gpg --verify doc.sig doc
```

Referencias

<https://www.gnupg.org/gph/en/manual.html>

Anexo 3: Creación de un certificado digital PKCS#12 con OpenSSL

OpenSSL es una herramienta de línea de comandos de código abierto que implementa funciones criptográficas. Vamos a utilizarla para crear un certificado digital autofirmado.

1. Genera un par de claves RSA (pública y privada)

```
$ openssl genrsa -out tunombre.keys
```

2. Genera el certificado autofirmado

```
$ openssl req -new -x509 -key tunombre.keys -days 365 -out tunombre.cer
```

Durante la ejecución del comando, se solicitarán los siguientes datos:

```
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Alicante
Locality Name (eg, city) []:Alicante
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IES Doctor Balmis
Organizational Unit Name (eg, section) []:[2SMRA|2SMRB]
Common Name (e.g. server FQDN or YOUR name) []:tu nombre
Email Address []:usuario@alu.edu.gva.es
```

3. Convierte el certificado a formato PKCS#12

```
$ openssl pkcs12 -export -out tunombre.p12 -inkey tunombre.keys -in tunombre.cer
```

Ahora puedes usar `tunombre.p12` para firmar archivos PDF