sy

[**Introducción**](#_heading=h.6oohgp16c8yd) **2**

[**Preparación del entorno de prácticas**](#_heading=h.8zhjlnd47weh) **2**

[**Tipos de cifrado**](#_heading=h.hl9tcu7efesn) **3**

[Cifrado simétrico](#_heading=h.3dy6vkm) 3

[Ejercicio de cifrado simétrico en GPG](#_heading=h.dr9y6xpir1yv) 4

[Cifrado asimétrico](#_heading=h.51yysyf2smq3) 5

[Ejercicio de cifrado asimétrico en GPG](#_heading=h.j1cq9cl33zkk) 6

[Firma digital](#_heading=h.193sw3gslyg1) 9

[Práctica gpg firma digital](#_heading=h.kmrzqrvqa575) 10

[Resumen](#_heading=h.ooi331h6h06u) 12

# Introducción

Desde que el Mundo existe, existen los secretos. Siempre ha habido información que no se podía hacer pública, ya fuera para mantener a salvo un país, un negocio o los intereses personales de alguien. Cuando había que comunicar esta información a un aliado, colaborador o amigo, había que cerciorarse de que sólo obtuviera la información el receptor al que iba dirigida y nadie más, para que la información permaneciera en secreto.

Una de las formas de transmitir información secreta es haciendo uso de la criptografía. La palabra criptografía proviene del griego kryptos, que significa oculto, y graphein, que significa escribir. El significado previsible a partir de esta etimología sería "escritura oculta". La criptografía es, pues, el arte y la ciencia de hacer las comunicaciones ininteligibles para todos excepto para el receptor autorizado, que poseerá la llave para desencriptar el mensaje.

El proceso de transformar un texto llano en texto cifrado o criptograma se llama encriptar o cifrar. Para este proceso se usa una llave que sólo conocen el emisor y el receptor. Cuando el receptor recoge el criptograma, lo puede desencriptar o descifrar con ayuda de la llave fácilmente. Sin embargo, si el mensaje es interceptado en el camino por un extraño, no es inteligible, y sin la llave, dependiendo de lo robusto que sea el método que se haya usado para encriptarlo, no podrá averiguar su contenido.

Los orígenes de la criptografía se remontan a la época de los egipcios, en la que ya se encriptaban mensajes haciendo uso de jeroglíficos. Desde entonces hasta ahora, las técnicas han evolucionado bastante.

El criptoanálisis es la ciencia que investiga cómo romper criptogramas, o sea, cómo poder averiguar el contenido de un texto cifrado sin conocer la llave que se usó para cifrarlo.

Las distintas aplicaciones de la criptografía y los avances que en su seno se han producido han determinado la existencia de tres grandes grupos de criptosistemas: de clave secreta, de clave pública e irreversibles.

En este documento analizaremos el funcionamiento de algunos de los tipos de algoritmos criptográficos más usuales.

# **Preparación del entorno de prácticas**

Para realizar las prácticas, instalaremos una máquina virtual con el sistema operativo Ubuntu Server; opcionalmente, utilizaremos NETinVM como entorno en el que ejecutar la máquina desde la cual hacer las pruebas.

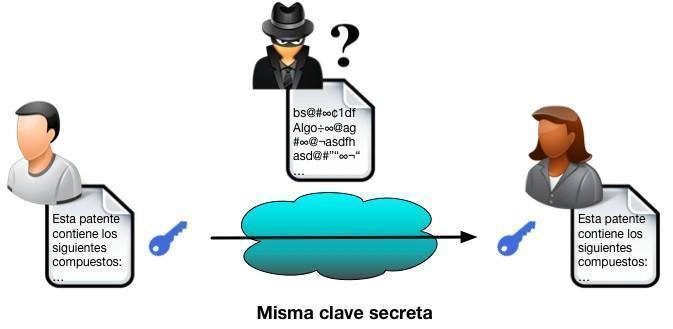
# **Tipos de cifrado**

## Cifrado simétrico

En el cifrado simétrico se utiliza un esquema de cifrado con una única clave, que debe ser usada y compartida en secreto por el emisor y el receptor. La principal ventaja que presentan es su simetría: los papeles del emisor y el receptor son intercambiables dado que ambos emplean la misma clave.

Suelen estar implementados por algoritmos muy eficientes. Así, no tienen alternativa cuando se trata de utilizar la criptografía para proteger información que no interviene en protocolos de comunicación (por ejemplo, cifrado de material reservado en discos duros, o de copias de seguridad para evitar su utilización indebida).

Además de asegurar la confidencialidad de la información, permiten autenticación del emisor: siempre que la clave permanezca en secreto, el receptor tiene la seguridad de que el emisor es quien dice ser y no un suplantador, dado que hipotéticamente son ellos dos los únicos que conocen la clave. Sin embargo esta autenticación no tiene valor fuera de los agentes de la comunicación, dado que se basa en la confianza mutua, y no puede ser utilizada ante terceros para evitar la repudiación de un mensaje por parte de quien lo ha escrito. Siempre se puede alegar que ha sido la otra parte quien lo ha escrito y cifrado o, si no, que ha sido otra persona a quien la otra parte ha suministrado indebidamente la clave.



Estos sistemas tienen otro punto débil: la necesidad de compartir la clave secreta entre el emisor y el receptor. Se supone que estos no consideran seguro su canal de comunicación normal (en otro caso, ¿para qué querrían usar la criptografía?), así que es preciso utilizar otro canal para compartir la clave: contacto personal, llamada telefónica, correo certificado... Esto es un problema severo en el mundo actual, en el que es necesario comunicarse a distancia con un gran número de personas u organizaciones con las que no se ha tenido jamás contacto previo.

Además, presentan graves problemas de gestión de claves, debido a que el número de claves secretas requeridas crece mucho con el de personas implicadas. Por cada dos interlocutores se necesita una clave, por lo que para n interlocutores el número de claves necesarias sería n(n-1)/2. Para una red de comunicaciones con 25 personas, esto supone la necesidad de gestionar el secreto de 300 claves. Además, cada persona debe recordar 24 claves, una por cada posible interlocutor, y ello utilizando métodos que no comprometan su confidencialidad.

Algunos ejemplos de algoritmos simétricos son DES, 3DES, RC5, AES, Blowfish e IDEA. Un buen sistema de cifrado pone toda la seguridad en la clave y ninguna en el algoritmo. Dado que toda la seguridad está en la clave, es importante que sea muy difícil adivinar el tipo de clave. Esto quiere decir que el abanico de posibilidades debe ser amplio.

### Ejercicio de cifrado simétrico en GPG

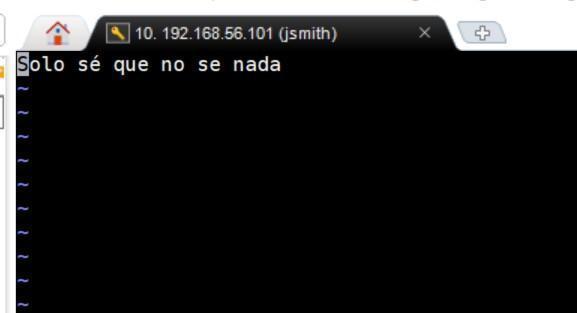
Crea un usuario de nombre jsmith

adduser jsmith

Loguéate con el usuario jsmith. Crea en el directorio home de ese usuario un fichero denominado mensaje001 e introduce una frase a cifrar.

login jsmith

nano mensaje001



Ahora, ejecuta el siguiente comando para cifrar el fichero utilizando el método de cifrado simétrico.

gpg --symmetric mensaje001

Introduce la contraseña deseada, por ejemplo: 112233445566. En un entorno real, sería preferible que fuera un conjunto de caracteres formados por minúsculas, mayúsculas, números y caracteres especiales además de tener una longitud de al menos 8-10 caracteres.

Ya tienes el fichero original cifrado. Ahora, borra el mensaje original.

Loguéate con otro usuario (*root*, por ejemplo). 

exit

login root

Si consultas el contenido del fichero cifrado, verás que no es legible.

cat mensaje.gpg

Desencripta el mensaje con:

gpg mensaje001.gpg



## Cifrado **asimétrico**

El cifrado asimétrico utiliza un par de claves relacionadas entre sí para el cifrado y descifrado de los mensajes. Una de las claves será la clave pública, que significa que será accesible para todo el mundo, y la otra será la clave privada.

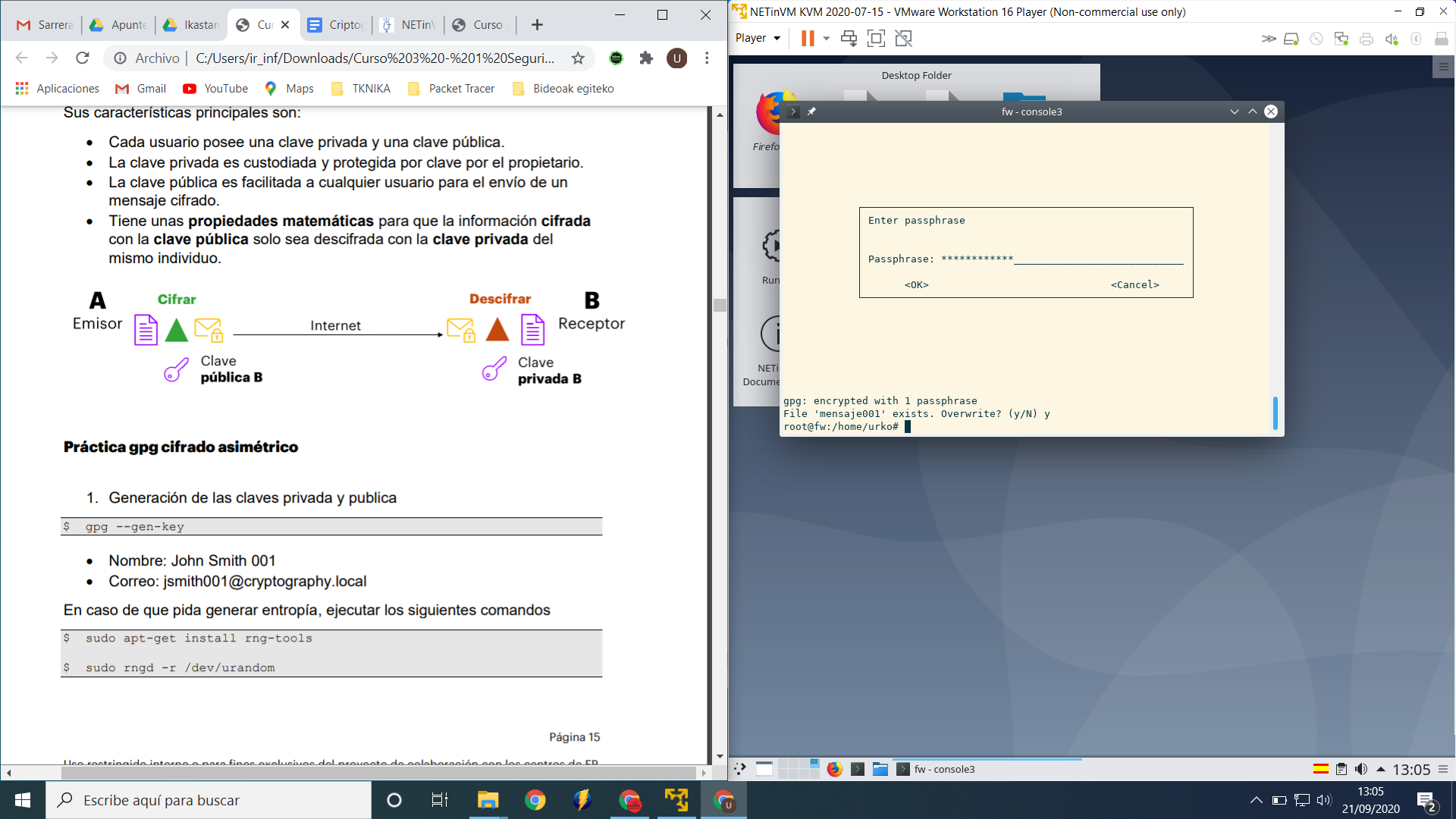
La clave pública se utiliza para encriptar y la clave privada para desencriptar. De tal modo que, si alguien quiere mandar un mensaje cifrado a un receptor, este mensaje deberá ser encriptado con la clave pública del propio receptor. Es imposible obtener la clave pública a partir de la privada.

El algoritmo de RSA es el primero y el más usado hoy en día. Nació en 1979 y además de para cifrar se utiliza también para firmar digitalmente documentos.

Sus características principales son:

* + Cada usuario posee una clave privada y una clave pública.
  + La clave privada es custodiada y protegida por clave por el propietario.
  + La clave pública es facilitada a cualquier usuario para el envío de un mensaje cifrado.
  + Tiene unas propiedades matemáticas para que la información cifrada con la clave pública solo sea descifrada con la clave privada del mismo individuo.

Este tipo de criptosistemas supera dos de los problemas inherentes a los de clave secreta, ambos relacionados con la gestión de claves: primero, no se necesita un canal seguro para enviar una clave, ya que esta es pública y podrá ser utilizada por todos aquellos que deseen remitir información al poseedor de la clave privada; segundo, si pensamos en una comunidad de n interlocutores, el número máximo de claves que se necesita es de 2n frente a las n(n-1)/2 que se necesitaban en los criptosistemas simétricos; además, cada usuario sólo debe recordar y proteger una clave.



Siguiendo el proceso del la imagen superior, ¿podemos asegurar la autenticidad del mensaje? La respuesta es que no podemos: como se ha cifrado el mensaje con la clave pública del receptor, cualquiera podría haberlo hecho. Para resolver eso, podríamos cifrar el mensaje con la clave privada del emisor, y descifrarla con su pública. Sin embargo, esta manera provocaría el inconveniente de que si el mensaje es interceptado por alguien más, podrá ser fácilmente descifrado con la clave pública del emisor. Afortunadamente, existen numerosas formas de utilizar la criptografía de clave pública para asegurar al mismo tiempo la confidencialidad y la autenticación, que estudiaremos más adelante cuando nos ocupemos de los mecanismos de firma digital.

### 

### Ejercicio de cifrado asimétrico en GPG

En este ejercicio, el usuario meeko cifra un mensaje con la clave pública del usuario jsmith. Posteriormente, éste desccifra el mensaje con su clave privada.

1. Generación de las claves privada y pública del usuario *jsmith*.

login jsmith

Loguéate con el usuario *jsmith*. A continuación, crea el par de claves.

gpg --gen-key

* + Nombre: John Smith
  + Correo: [jsmith@cryptography.local](mailto:jsmith001@cryptography.local)
  + Contraseña: 112233445566

En caso de que pida generar entropía, ejecuta los siguientes comandos:

sudo apt-get install rng-tools  
 sudo rngd -r /dev/urandom

1. Exportar la clave pública del usuario *jsmith,* obteniendo un fichero de extensión “.pub”.

Lo primero, lista las claves para comprobar que se ha generado correctamente el par.

gpg --list-key

Tras verlas, exporta la deseada identificándola con el correo al final del comando.

gpg --armor --export --output jsmith.pub jsmith@cryptography.local

Ve el contenido del fichero jsmith.pub

cat jsmith.pub



Copia el fichero a la carpeta temporal del sistema.

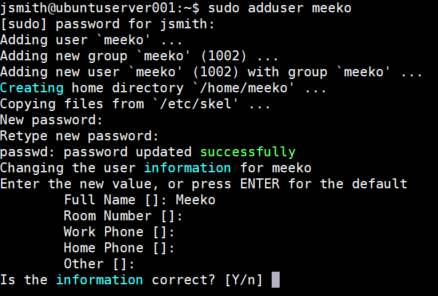
cp jsmith.pub /tmp/.

Cierra la sesión del usuario jsmith

1. Crear el usuario emisor.

Crea un usuario en el sistema de nombre *meeko*.

sudo adduser meeko



Contraseña: Password123!

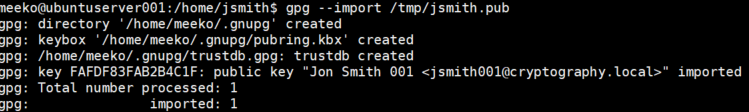
1. Cifra un mensaje

Para poder cifrar un mensaje para *jsmith*será necesario que te loguees con el usuario *meeko*.

exit

login meeko

Ahora, importa la clave pública de jsmith

gpg --import /tmp/jsmith.pub

Crea un fichero (que cifrarás a continuación) con un contenido.

nano /tmp/xa\_jsmith.txt

Una vez escrito el mensaje, cífralo con la clave pública de jsmith.

gpg --output /tmp/xa\_jsmith.txt.gpg ----encrypt recipient jsmith@cryptography.local --encrypt /tmp/xa\_jsmith.txt

Aparece una advertencia: la clave pública que vas a utilizar no está certificada; es decir, nadie nos asegura que esa clave sea de quien creemos que es.

En este caso, como estamos en un entorno de práctica y sabemos cómo y quién ha creado la clave, nos fiaremos -pero en un entorno real sería muy arriesgado hacerlo-. Responde con Y a la pregunta,

El contenido del mensaje está cifrado.

cat /tmp/xa\_jsmith.txt.gpg

Borra el archivo /tmp/xa\_jsmith.txt simplemente para deshacernos del archivo original y no confundirlo posteriormente con el cifrado

1. Descifra el mensaje

Para descifrar el contenido del mensaje, primeramente, deberemos desloguearnos como usuario *meeko* y loguearnos como *jsmith*.

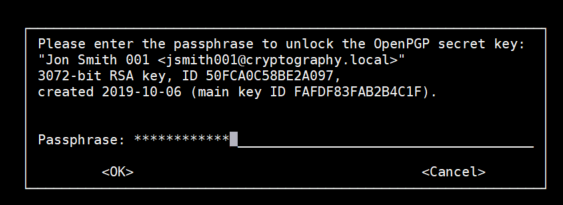
exit

login jsmith

A continuación, estando logueados como jsmith, descifraremos el contenido con la clave privada de *jsmith*.

gpg --output xa\_me.txt --recipient jsmith@cryptography.local --decrypt /tmp/xa\_jsmith.txt.gpg

Nos pedirá la contraseña para el acceso a la clave privada.



¡Ojo! Esa Passphrase que introducimos (la introducida al crear el par de claves: 112233445566) no es la clave privada; se trata de una medida de seguridad adicional para acceder a ella. La clave privada ha sido creada al generar el par de claves, y está en posesión del usuario *jsmith*.

Una vez descifrado, veremos el contenido del mensaje.

cat xa\_me.txt

Ten en cuenta que para llevar a cabo esta práctica nos hemos pasado las claves a través del directorio /tmp/. Digamos que ha sido una simulación; en la realidad, las claves públicas, al no ser confidenciales, suelen quedan registradas en algún repositorio.

## **Firma digital**

La firma digital permite comprobar la **autenticidad** del emisor frente a un documento y obligar a cumplir lo que en el documento se ha acordado. En la **era digital,** donde la información fluye a gran velocidad, hace falta agilizar la comprobación de los documentos sin tener que personarse para firmarlos.

Así se crea la firma digital:

1. Se crea una **huella digital (hash)** del documento a firmar.
2. Cifrado de la **huella digital** con la **clave privada del emisor.**
3. Se envía la **huella digital cifrada.**

Así se comprueba la firma digital:

1. Descifrar la **huella digital cifrada** usando la **clave pública del emisor**.
2. Se obtiene la **huella digital (hash)** del documento.
3. Se **comparan** las dos huellas digitales. Si son iguales, el documento ha llegado de manera íntegra.

### Práctica gpg firma digital

Loguéate con el usuario *meeko*. Crea el par de claves para el usuario *meeko*. Exporta la clave pública, y déjala en el directorio */tmp*; es decir, repite los pasos que has hecho previamente para el usuario *jsmith*. Ahora, recuperando el fichero anterior *xa\_jsmith.txt.gpg*, vamos a añadir la firma digital. Para poder firmar el fichero como *meeko* (creador y emisor del mensaje), ejecuta el siguiente comando (recuerda que debes estar logueado con él).

gpg --armor --detach-sign /tmp/xa\_jsmith.txt.gpg

Nota: este comando lo que hace es firmar el archivo indicado; en este caso: crear el hash del fichero gpg (cifrado del original) y cifrar ese hash.



Copia el fichero *.asc* en la carpeta temporal del sistema:

cp xa\_jsmith.txt.gpg.asc /tmp

Y copia también el mensaje enviado (cifrado) si no lo estuviera.

cp xa\_jsmith.txt.gpg /tmp

Ahora, loguéate como jsmith de nuevo.

exit

login jsmith

Con el usuario jsmith, importa la clave pública de meeko.

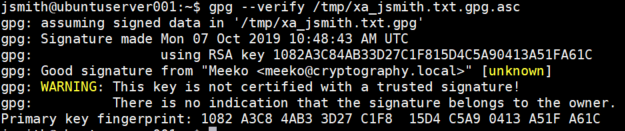
gpg --import /tmp/meeko.pub

Y realiza la comprobación del fichero ejecutando el siguiente comando.

gpg --verify /tmp/xa\_jsmith.txt.gpg.asc

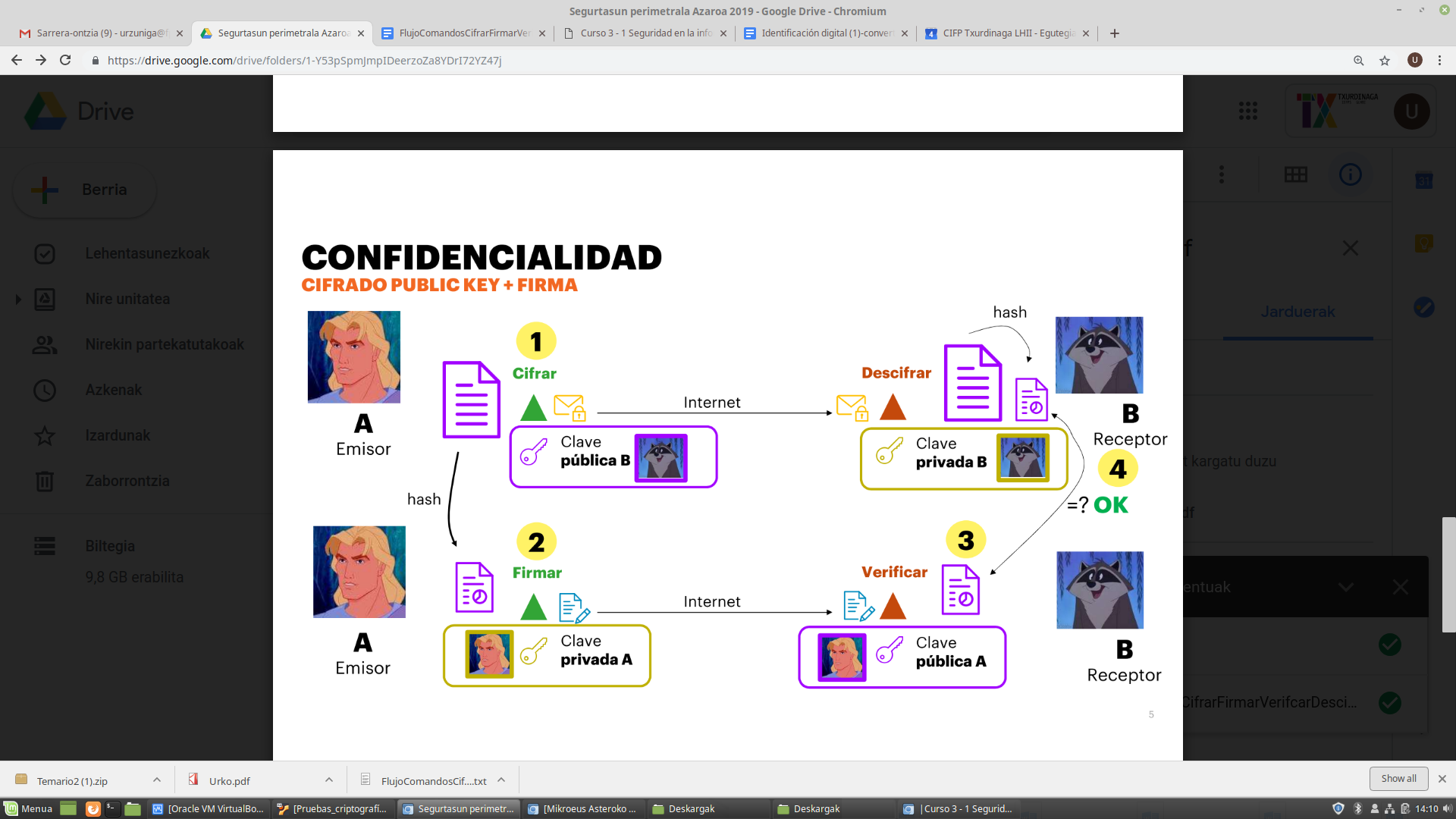
Nota: este comando lo que hace es:

1. Obtener el hash del documento *.gpg* recibido.
2. Descifrar el fichero .asc con la clave pública del emisor.
3. Comparar los hashes resultantes de los pasos 1 y 2.



### **Resumen**

Nota: en el dibujo de debajo el usuario *jsmith* es quien envía un mensaje al usuario *meeko*.



Resumiendo el proceso, estos son los pasos que se siguen a la hora de enviar un mensaje hoy en día (aunque el proceso suele ser transparente para nosotros):

1. El emisor cifra el documento original con la clave pública del receptor, y lo envía.
2. El emisor hashea el documento original cifrado, cifra el hash resultante con su clave privada, y lo envía. (AUTENTICACIÓN+NO REPUDIO)
3. El receptor hashea el documento original cifrado recibido. Por otra parte, descifra, con la clave pública del emisor, el hash recibido. Compara ambos hashes. Recuerda que estos son hashes del documento original CIFRADO.
4. Si los hashes son iguales, continúa (INTEGRIDAD).
5. El receptor descifra el documento original cifrado utilizando su clave privada (CONFIDENCIALIDAD).

Por lo tanto, este sistema permite garantizar que el documento no se ha modificado en el transcurso de la comunicación (integridad), que sólo lo puede leer el receptor autorizado (confidencialidad), que el documento lo envía quien dice ser (autenticidad) y que ni el emisor ni el receptor puedan repudiar el envío y recepción del mensaje (no repudio), respectivamente.