[**Introducción**](#_heading=h.k74v5uq02p0m) **1**

[El método tradicional César](#_heading=h.y2pkcvyt13ub) 1

[Características de un sistema de cifrado (CIDAN)](#_heading=h.awoefu97o56) 2

[**Tipos de cifrados**](#_heading=h.su8hpu4t1wyi) **3**

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

[¿Qué es?](#_heading=h.9zxnw2n3bnlx) 3

[Entorno de prácticas](#_heading=h.4d34og8) 4

[Descarga e Instalación de VirtualBox](#_heading=h.l64auvgmz2t8) 4

[Instalación de Ubuntu](#_heading=h.2s8eyo1) 4

[Ejercicio Cifrado simétrico en GPG](#_heading=h.xmiyt399sfx5) 7

[Cifrado asimétrico](#_heading=h.ej846cl9aj3f) 9

[Ejercicio Cifrado asimétrico en GPG](#_heading=h.ai8y4cagwcte) 10

[Firma digital](#_heading=h.mfx1ovkzje04) 13

[Práctica gpg firma digital](#_heading=h.pw0s61dhco7r) 13

[Resumen](#_heading=h.tw5gb3bdd2n5) 14

# Introducción

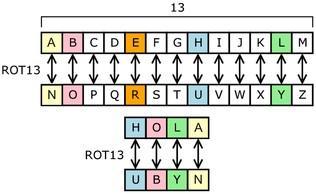
Una de las formas de transmitir información secreta es haciendo uso de la criptografía. La palabra criptografía proviene del griego *kryptos*, que significa oculto, y *graphein*, que significa escribir; es decir, algo así como "escritura oculta". La criptografía es, pues, el arte y la ciencia de hacer las comunicaciones ininteligibles para todos excepto para el receptor autorizado, quien poseerá la llave o clave 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 clave que sólo conocen el emisor y el receptor. Cuando el receptor recoge el criptograma, lo puede desencriptar o descifrar con ayuda de la clave. Sin embargo, si el mensaje es interceptado en el camino por un extraño, no le será inteligible, y, sin la clave, dependiendo de lo robusto que sea el método que se haya usado para encriptarlo, no podrá averiguar su contenido.

Un buen sistema de cifrado debe ajustarse al grado de seguridad requerido. El esfuerzo y los recursos consumidos deben ser acordes. Al igual que los mecanismos de cifrado y descifrado deben ser sencillos, la implantación de los algoritmos también debe ser sencilla.

## El método tradicional *César*

Uno de los sistemas de encriptación más famosos es el cifrado de César. Este sistema mueve cada letra del alfabeto un determinado número de veces.







## Características de un sistema de cifrado (CIDAN)

Se considera seguro un sistema que cumple con las propiedades de integridad, confidencialidad y disponibilidad de la información.



Disponibilidad

Garantiza que el sistema y los datos estarán disponibles para los usuarios; se deben aplicar medidas que protejan la información, así como crear copias de seguridad y mecanismos para restaurar los datos que accidentalmente o intencionadamente se hayan dañado o destruido.

Confidencialidad

Garantiza que el acceso a la información no se produce de forma no autorizada; dicho de otra forma: los componentes del sistema serán accesibles sólo por aquellos usuarios autorizados.

Para prevenir errores de confidencialidad, debe diseñarse un control de accesos al sistema: quién puede acceder, desde dónde, a qué parte del sistema, en qué momento y para realizar qué tipo de operaciones.

Integridad

Garantiza que la información no ha sido modificada sin autorización; es decir: los componentes del sistema sólo pueden ser creados y modificados por los usuarios autorizados.

Para evitar riesgos relacionados con la integridad, se debe dotar al sistema de mecanismos que prevengan y detecten cuándo se produce un fallo de integridad y traten y resuelvan los errores que se han descubierto.

Autenticación

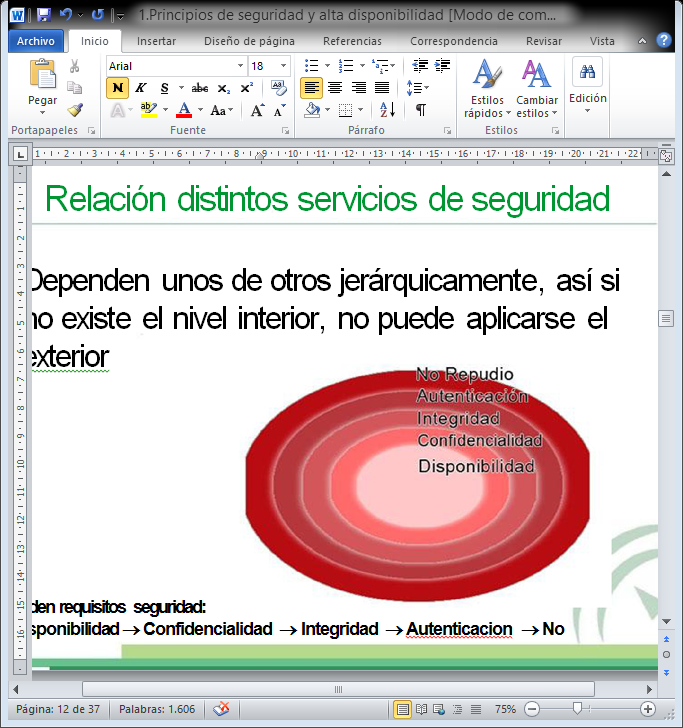
Garantiza que un documento ha sido elaborado (o pertenece) por quien el documento dice. Se trata, por lo tanto, de que la identidad del emisor no sea suplantada. Entraría en juego aquí la criptografía.

No repudio

Impide que el emisor niegue haber estado involucrado en una comunicación. Puede ser:

* No repudio en origen: el emisor no puede negar el envío; la prueba la crea el emisor y la recibe el destinatario.
* No repudio en destino: el receptor no puede negar la recepción ya que el emisor tiene pruebas de la recepción, creadas por el receptor y recibidas por el emisor.

Podemos decir que las propiedades conforman una estructura jerárquica, de tal manera que si no se cumple la interior, tampoco lo hará la exterior.



# Tipos de cifrados

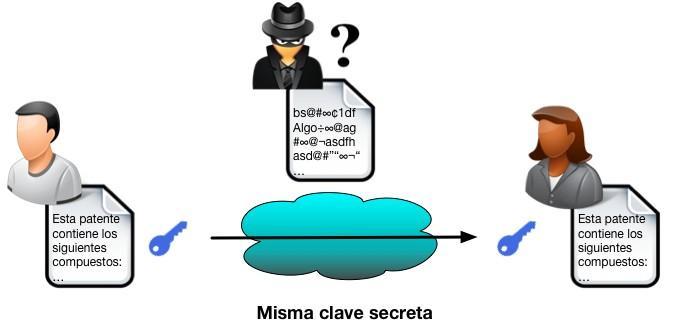
## Cifrado simétrico

### ¿Qué es?

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.

**Entorno de prácticas**

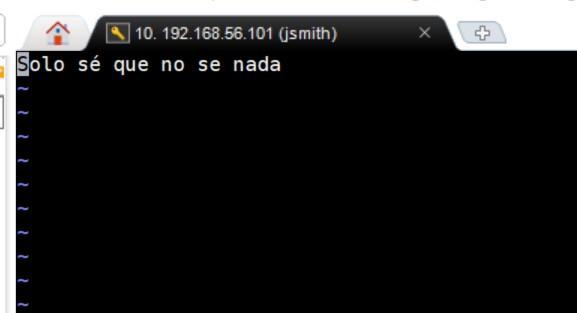
Para realizar las prácticas, utilizaremos una máquina virtual en Ubuntu. Ejercicio Cifrado simétrico en GPG

Crearemos un usuario de nombre: *jsmith*

A continuación, crea un fichero denominado *mensaje001* e introduce una frase a cifrar.

$ adduser jsmith

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



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. 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, loguéate con otro usuario (*su*, por ejemplo)

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

$ cat mensaje.gpg

Desencripta el mensaje con:



$ gpg mensaje001.gpg

Existe la creencia de que el sistema de cifrado simétrico es más débil que el sistema asimétrico con clave privada y pública. Esto se debe a que en el cifrado simétrico se tiene que compartir la clave mediante otro medio de comunicación seguro; como en el cifrado asimétrico ese problema está resuelto, por eso se utiliza más.

## 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.

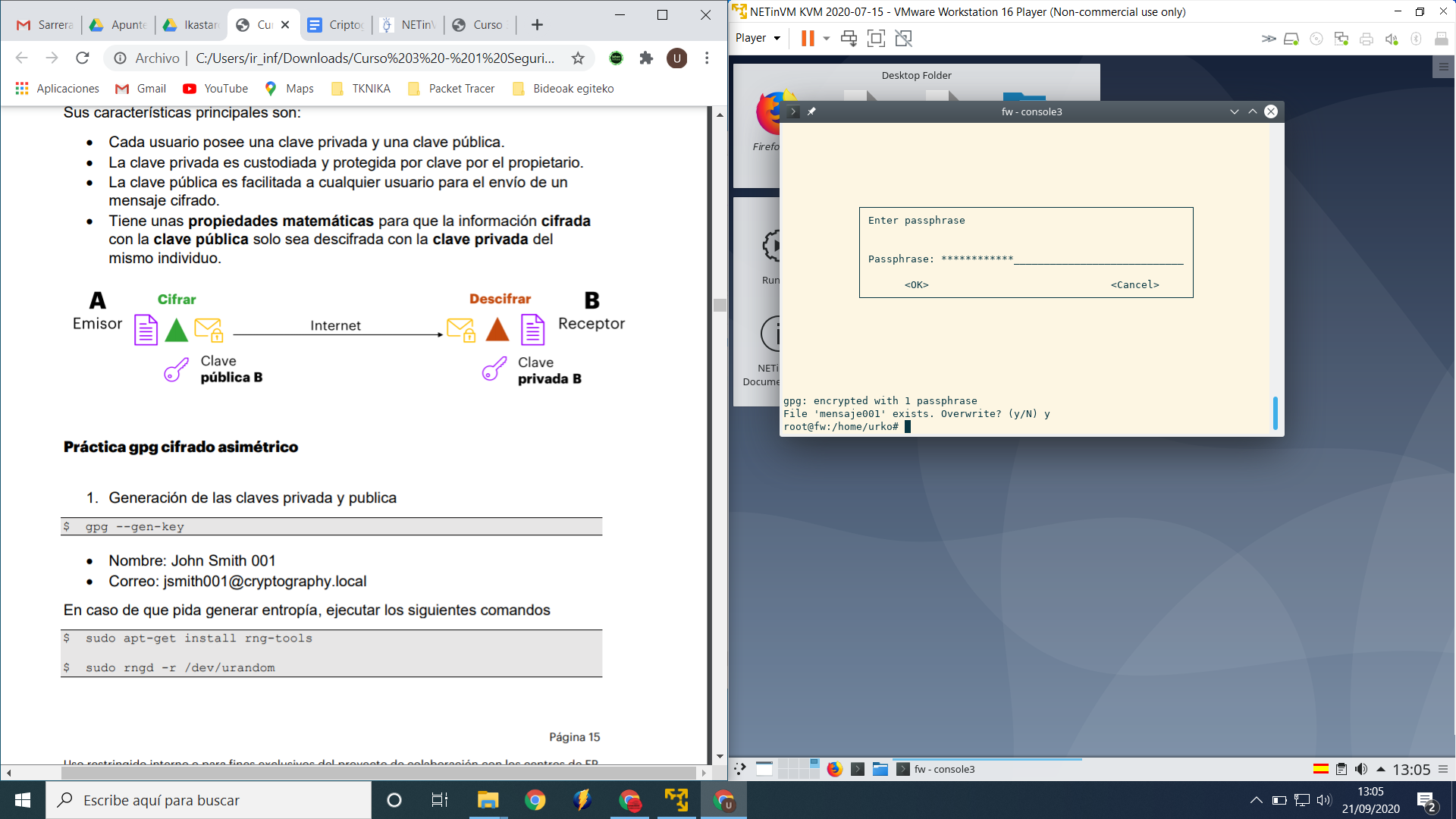
La clave pública se utiliza para encriptar y la clave privada para desencriptar. De tal modo que, si alguien quiere mandar un mensaje cifrado a un receptor, este mensaje deberá ser encriptado con la clave pública del propio receptor. Es imposible obtener la clave pública a partir de la privada.

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 ésta 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 de 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

1. Generación de las claves privada y publica

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. Exporta la clave pública. 

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

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

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

$ gpg --list-key

Ve el contenido del fichero jsmith.pub

$ cat jsmith.pub



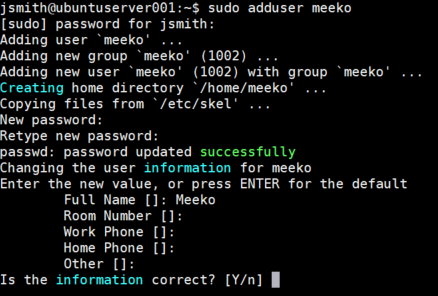
Copia el fichero a la carpeta temporal del sistema.

$ cp jsmith.pub /tmp/

1. Crea el usuario emisor: *meeko*

Crea otro usuario en el sistema para intercambiar los mensajes cifrados. Para ello:

$ sudo adduser meeko



Contraseña: Password123!

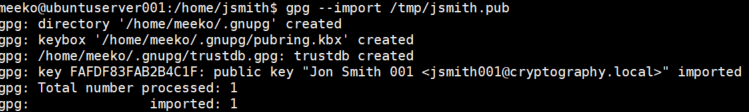
1. Cifra el mensaje

Para poder cifrar un mensaje para *jsmith*será necesario que te loguees con el usuario *meeko*e importes la clave pública de *jsmith*. Para ello:

$ su meeko

Importa la clave.

$ gpg --import /tmp/jsmith.pub



Crea un fichero con un contenido a cifrar.

$ nano /tmp/xa\_jsmith.txt

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

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

Responde con Y la pregunta.

El contenido del mensaje está cifrado:

$ cat /tmp/xa\_jsmith.txt.gpg



1. Descifra el mensaje

Para descifrar el mensaje, lo primero será desloguearnos como usuario *meeko* y loguearnos como *jsmith*.

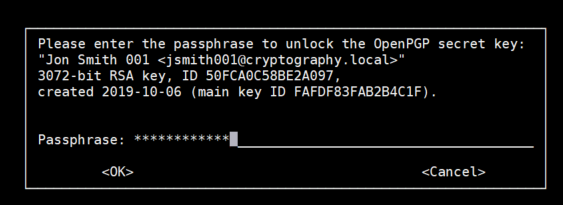
$ exit

$ login jsmith

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

$ 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 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 realidad, las claves públicas, al no ser confidenciales, quedan registradas en algún repositorio.



Siguiendo este proceso, ¿podemos asegurar la procedencia del mensaje? La respuesta es que no podemos, ya que, como hemos 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 varios inconvenientes; ¿cuáles se te ocurren? 

## 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.

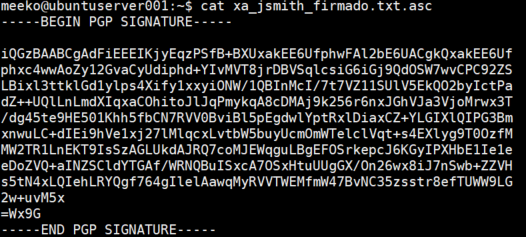
### 

### Práctica gpg firma digital

Crea el par de claves para el usuario *meeko*. Exporta la clave pública, y déjala en el directorio */tmp*; es decir, repito 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*, ejecutaremos 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 mensaje; es decir: crear el hash del fichero gpg 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

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

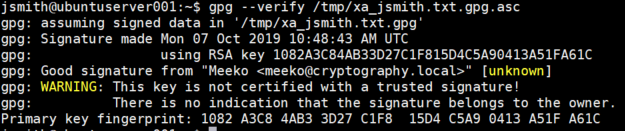
$ gpg --import /tmp/meeko.pub

Y realiza la comprobación del fichero ejecutando el siguiente comando.

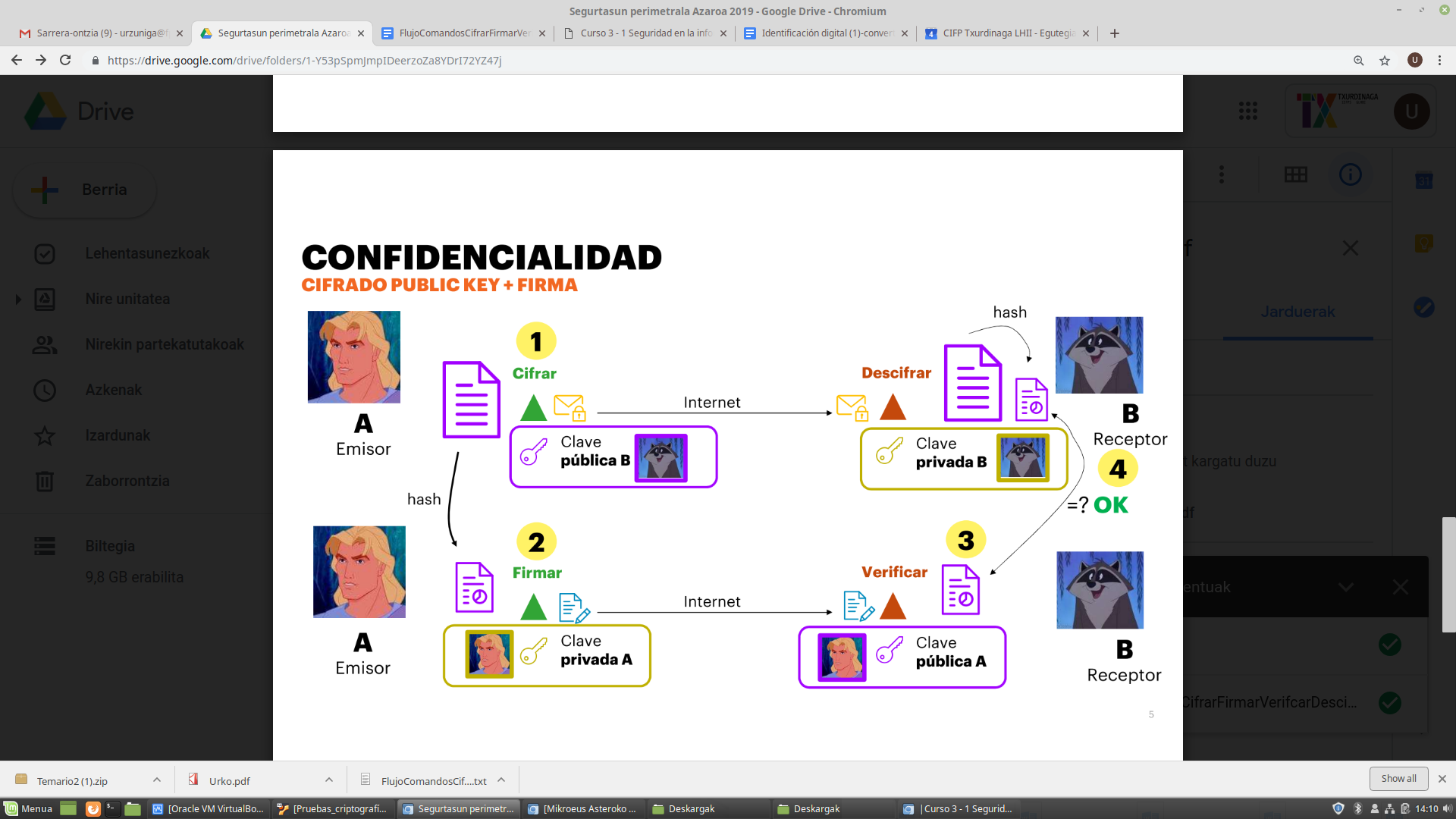
$ gpg --verify /tmp/xa\_jsmith.txt.gpg.asc

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



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