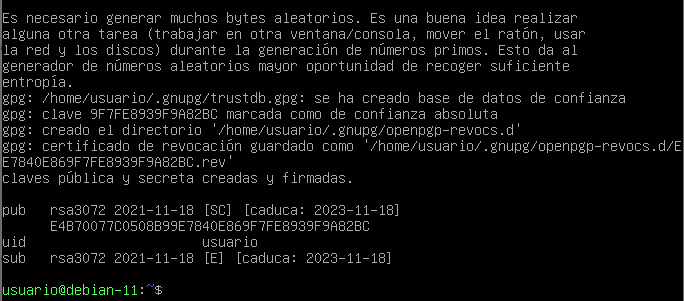
En el caso práctico Anterior, hemos utilizado la herramienta gpg para el cifrado simétrico. Esta misma herramienta la utilizamos para el cifrado asimétrico, La siguiente, es la secuencia de comandos que vimos en el ejemplo de clase:

Entramos con un usuario de la máquina.

Lo primero será generar un par de claves de criptografía asimétrica, nuestra propia clave pública y clave privada.

El comando es *gpg --gen-key.*

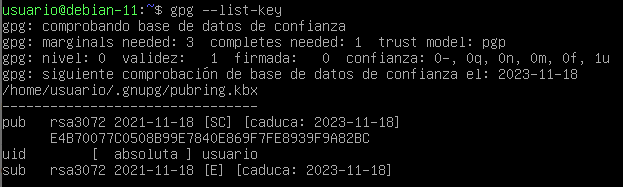


En el proceso de generación de la clave nos pregunta varios detalles.

El primero es el tipo de clave, la herramienta nos ofrece cuatro opciones.

Los nombres se corresponden con el tipo de algoritmo asimétrico asociado (hay varios tipos, como también ocurría en criptografía simétrica: DES, AES, etc.). Es decir, una clave de tipo DSA se utiliza en un algoritmo DSA, y una clave Elgamal, en un algoritmo Elgamal.

Podemos consultar las claves disponibles en el llavero con el mismo parámetro - -list-key que vimos anteriormente: *gpg --list-keys*

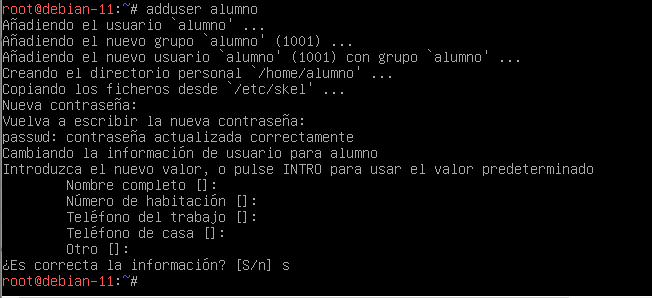


Comparado con el caso práctico del cifrado simétrico, todavía no hemos cifrado nada: como ya suponíamos, la criptografía asimétrica es un poco más complicada. Ahora tenemos que comunicar nuestra clave pública a quien interesado en enviar- nos un mensaje cifrado-

Primero tenemos que sacarla del llavero.

*Gpg - -export –a -o /tmp/claveusuario1.pub*

Hemos utilizado los parámetros a (ascii) para que el resultado no sea binario y o (salida) para guardarlo directamente en un fichero (si no, aparece por la salida estándar).

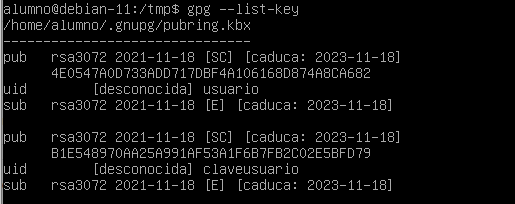


El fichero lo ponemos tranquilamente en / tmp porque no nos importa que otros usuarios lo copien. En la máquina tenemos un segundo usuario llamado Usuario2 . Entramos con este usuario para enviar un mensaje cifrado al alumno con la misma herramienta gpg.

Para coger las claves públicas de usuario 1 utilizamos el parámetro import

*gpg --import /tmp/alumno.pub*

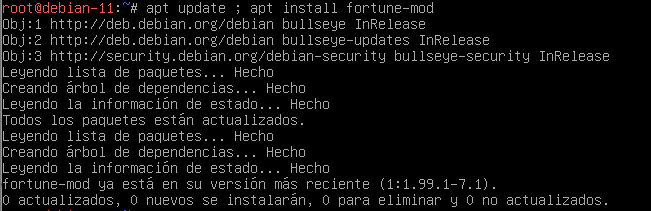
**

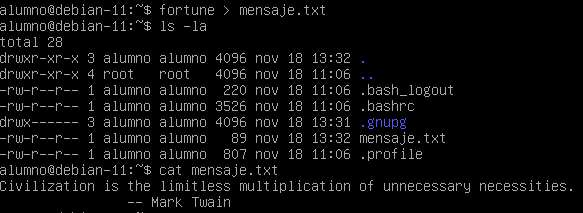
**

Vamos a crear un fichero llamado mensaje y lo cifraremos para enviárselo al usuario alumno.

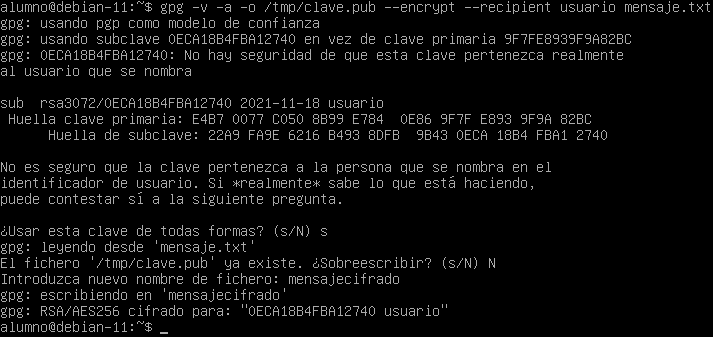
Los comandos:

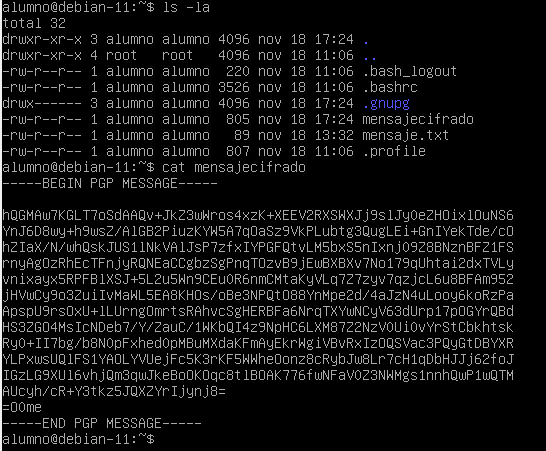
*fortune> mensaje*





*gpg -v -a -o /tmp/mensaje.cifrado --encrypt --recipient Usuario 1 mensaje*

**

**

Los parámetros a y o ya los conocemos (dejamos el fichero en / tmp porque no nos importa que otros usuarios puedan leerlo sin la clave privada, el contenido es ininteligible).

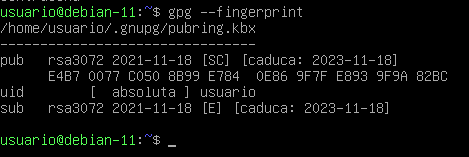
Hemos añadido el parámetro v (verbose) para obtener más información. En este caso, nos sirve para conocer que utilizará la clave subordinada (Elgamal), en el lugar de la clave primaria (DSA). El parámetro encrypt indica que deseamos cifrado asimétrico y el parámetro recipient va seguido del identificador de la clave pública que queremos utilizar. Como ya sabemos, el cifrado utiliza la clave pública del recepto

En nuestro ejemplo el usuario Usuario2 es la primera vez que utiliza la herramienta gpg, por lo que se le informa que se crea el directorio. gnupg y los archivos internos. ofrece la huella (fingerprint) de la clave y nos pide la confirmación de que es la clave que queremos Utilizar.

Podemos volver a la sesión del usuario alumno, obtener la huella de su clave pública y comparar con la que aparece en la sesión de profesor.

El parámetro para obtener la huella es fingerprint

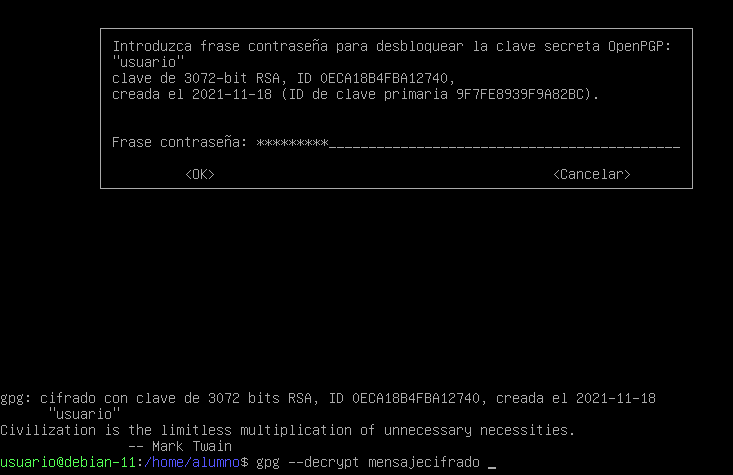
*gpg – fingerprint*



Efectivamente, las huellas coinciden y podemos confiar en que la clave importada en profesor es correcta. Confirmamos que queremos usar esa clave y se crea el fichero cifrado.

Ahora podemos volver a la sesión del usuario1 e intentar descifrarlo. El parámetro es decrypt:

*gpg --decrypt / tmp / mensaje. Cifrado*



Como esperábamos, el comando nos solicita la contraseña que hacer acceso a la clave privada.

En la ventana aparece que clave necesita (en nuestro caso, la clave del algoritmo Elgamal), con toda su identificación.

Una vez introducido ese comando, la respuesta es una advertencia: no hay seguridad de que esa clave pública sea realmente la clave pública de Usuario1. Cualquiera podría haber cambiado el fichero /tmp /alumno.pub antes de que Usuario2 hiciera el import de las claves.

Si hemos introducido bien la contraseña, aparecerá el mensaje que nos ha enviado el usuario Usuario2

Podemos comprobar que pasa si el fichero, por cualquier razón, resultó dañado (defecto físico del disco duro, error en la transmisión, etc

En el usuario Usuario1 hacemos una copia fichero y lo editamos para cambiar cualquier letra Si después lo intentamos descifrar, aparecerá un error.

Si ahora el usuario usuario1 quisiera enviar un fichero cifrado a Usuario2, primero profesor debería generar su propio par de claves, después exportar la clave pública, hacerla llegar hasta el alumno para que la importe, etc.