

Certificados digitales

Práctica 4



Índice

[Crear una autoridad certificadora 2](#_Toc530847115)

[1. Creamos la estructura de directorios 2](#_Toc530847116)

[2. Preparamos el fichero de configuración ca/openssl.cnf 3](#_Toc530847117)

[3. Creamos la nueva clave raíz en ca/private 3](#_Toc530847118)

[4. Creamos el certificado raíz 3](#_Toc530847119)

[5. Archivos generados 4](#_Toc530847120)

[6. Valor de las claves generadas 5](#_Toc530847121)

[Cread una autoridad certificadora subordinada a la anterior. 7](#_Toc530847122)

[1. Creamos la estructura de directorios 7](#_Toc530847123)

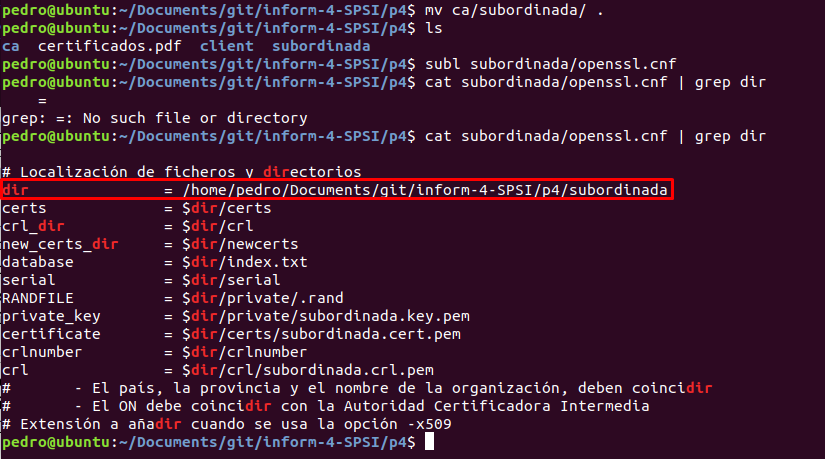
[2. Modificamos el archivo de configuración de la CA subordinada 8](#_Toc530847124)

[3. Creamos la clave privada de la CA subordinada 8](#_Toc530847125)

[4. Creamos el certificado de la CA subordinada 8](#_Toc530847126)

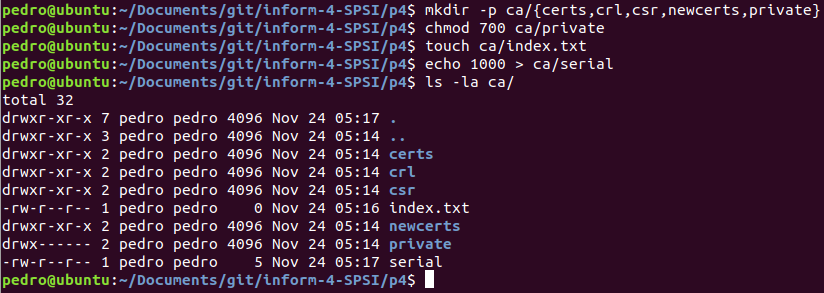
# Errores

Cuando he creado la Autoridad certificadora subordinada, la creé dentro de la autoridad certificadora raíz. Esto no es lo más correcto, por lo que llegado a un punto de la práctica que se especificará más adelante, lo cambié y la saqué fuera.



# Crear una autoridad certificadora

## Creamos la estructura de directorios

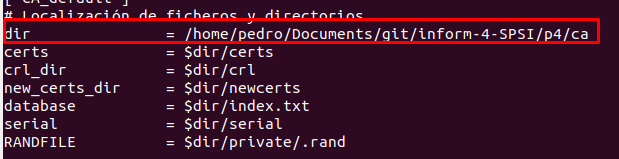


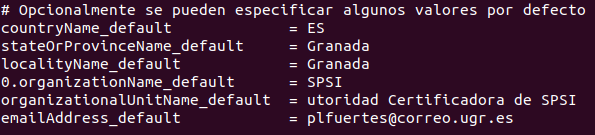
* Certs: es el directorio donde se guardarán los certificados emitidos.
* Newcerts: es el directorio donde se guardarán los nuevos certificados.
* Private: es el directorio donde se guardarán las claves privadas de la entidad.
* Csr: Es el directorio donde se guardarán las peticiones de firma de certificado. (Certificate Signing Request)

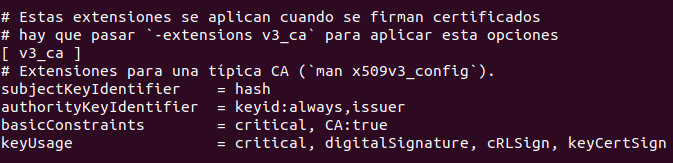
Además, se han recado los archivos:

* Index.txt: Es índice de la base de datos de los certificados.
* Serial: se guarda el número del siguiente certificado firmado.

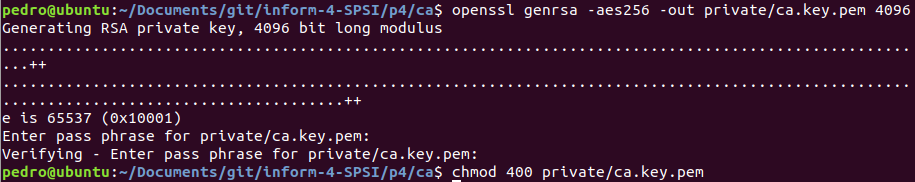
## Preparamos el fichero de configuración ca/openssl.cnf



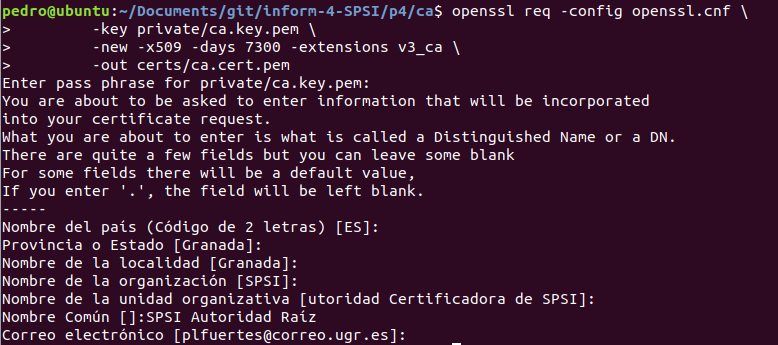




## Creamos la nueva clave raíz en ca/private

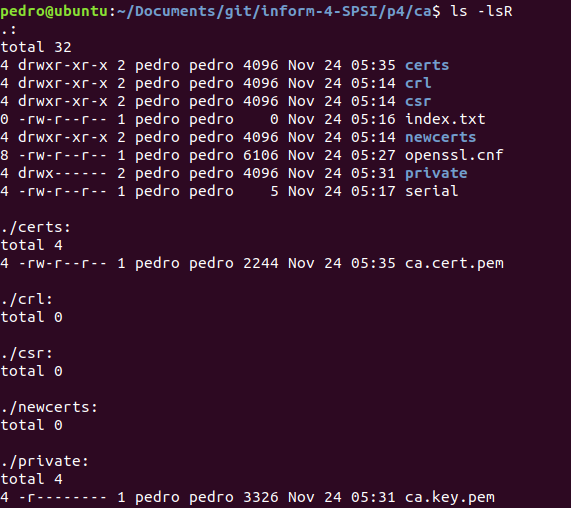


## Creamos el certificado raíz



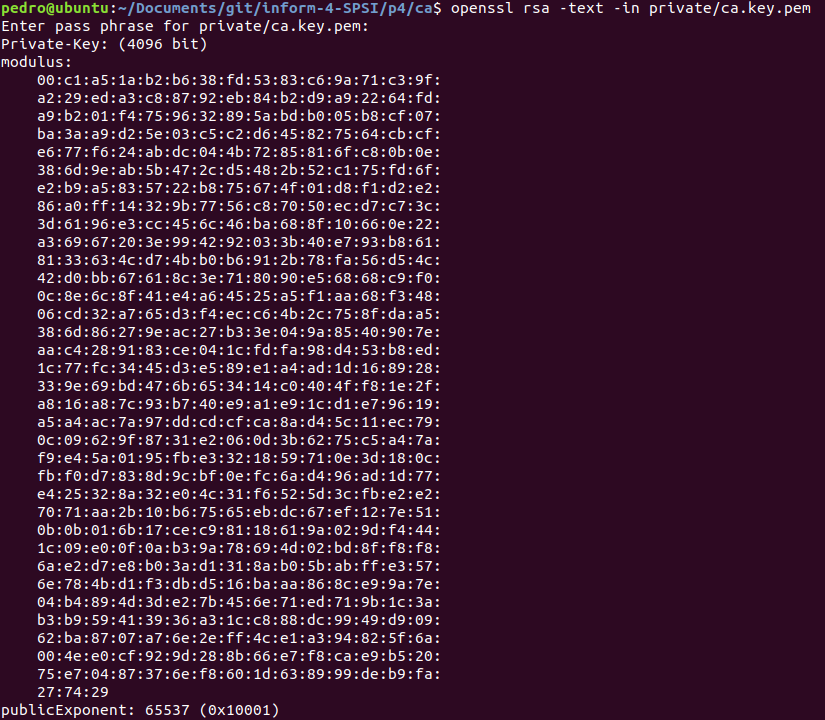
* Openssl req: permite solicitar certificaciones. Las solicitudes se pueden crear con claves previamente creadas o crearlas en el momento.
* - config: permite utilizar un archivo de configuración alternativo, en este caso, el creado previamente en el punto 2. Por eso algunos puntos ya nos salían rellenos.
* -key: indica la clave privada con la que firmar las solicitudes
* -new: crea una nueva solicitud de certificado
* -x509: para que el cetificado que se cree esté autofirmado
* -days: el número de días en los que expirará el certificado. (20 años en este caso)
* -extensions: Indica la extensión del certificado que se agregará al emitir cada certificado, en este caso v3\_ca como se ha especificado en el archivo openssl.cnf
* -out: indica dónde se almacenará el certificado que permitirá firmar otras solicitudes de certificado como verificar la identidad de los solicitantes.

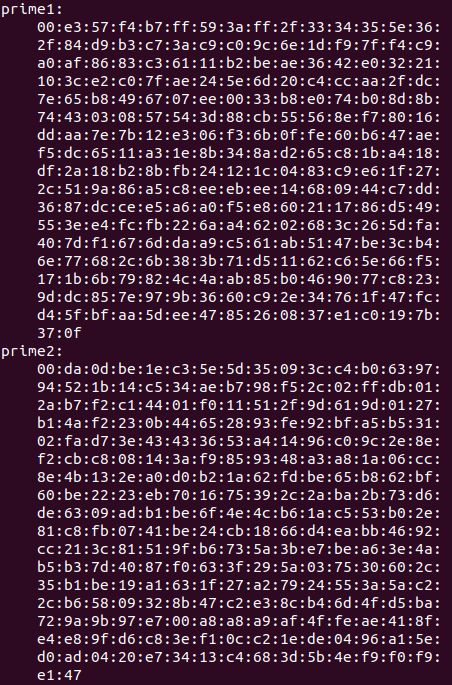
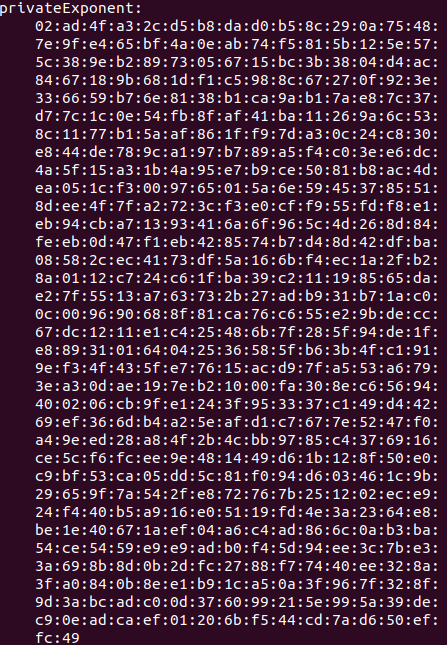
## Archivos generados

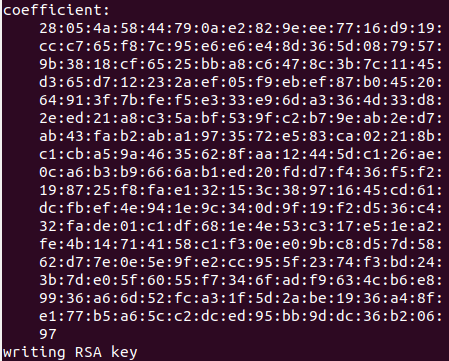
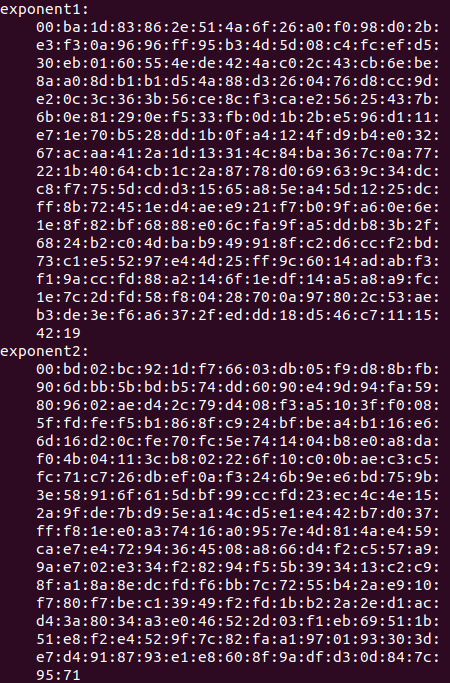


Aquí puede verse como a parte de los ficheros generados, en el punto 1, se ha creado una clave privada llamada ca.key.pem (Punto 3) en la carpeta ca/private, así como un certificado autofirmado en /ca/certs llamado de la misma forma (Punto 4).

## Valor de las claves generadas

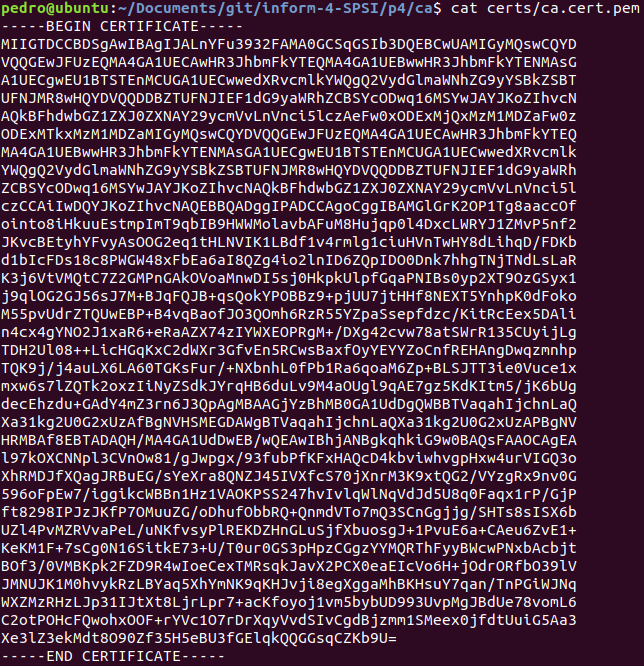






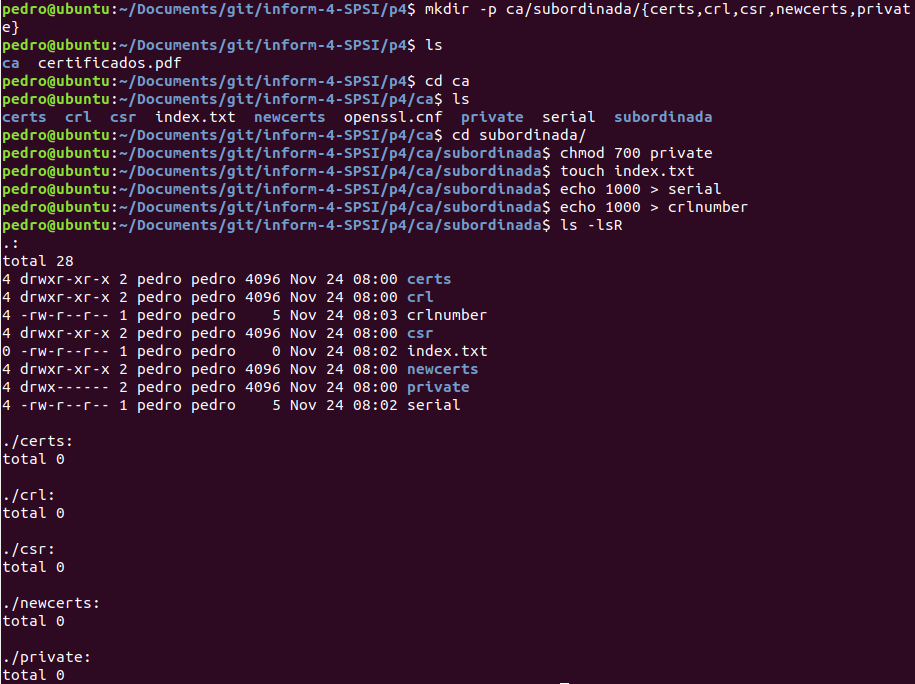


## Valor de los certificados generados



# Cread una autoridad certificadora subordinada a la anterior.

## Creamos la estructura de directorios



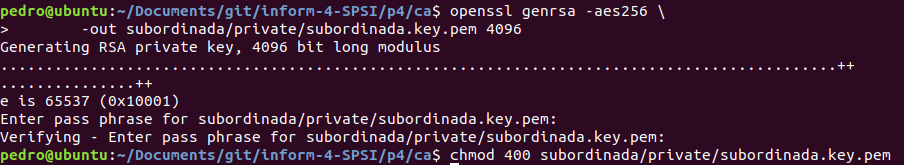
El sistema de archivos es similar salvo por que ha creado el fichero crlnumber donde se almacena el número de serie de las listas de revocación de certificados.

## Modificamos el archivo de configuración de la CA subordinada

Debemos modificar los siguientes parámetros

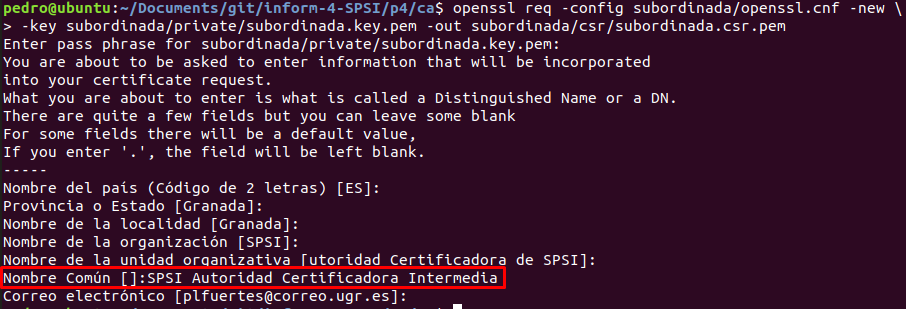
|  |  |
| --- | --- |
| /ca/openssl.cnf | /ca/subordinada/openssl.cnf |
| Dir =  /home/pedro/Documents/git/inform-4-SPSI/p4/ca | dir =  /home/pedro/Documents/git/inform-4-SPSI/p4/**ca/subordinada** |
| private\_key =  $dir/private/ca.key.pem | private\_key =  $dir/private/**subordinada**.key.pem |
| certificate =  $dir/certs/ca.cert.pem | certificate =  $dir/certs/**subordinada**.cert.pem |
| crl =  $dir/crl/ca.crl.pem | crl =  $dir/crl/**subordinada**.crl.pem |
| Policy = policy\_strict | policy = policy\_**loose** |
|  | copy\_extensions = copy |

## Creamos la clave privada de la CA subordinada



Cambiamos los permisos con chmod para que sólo se pueda leer y siendo superusuario.

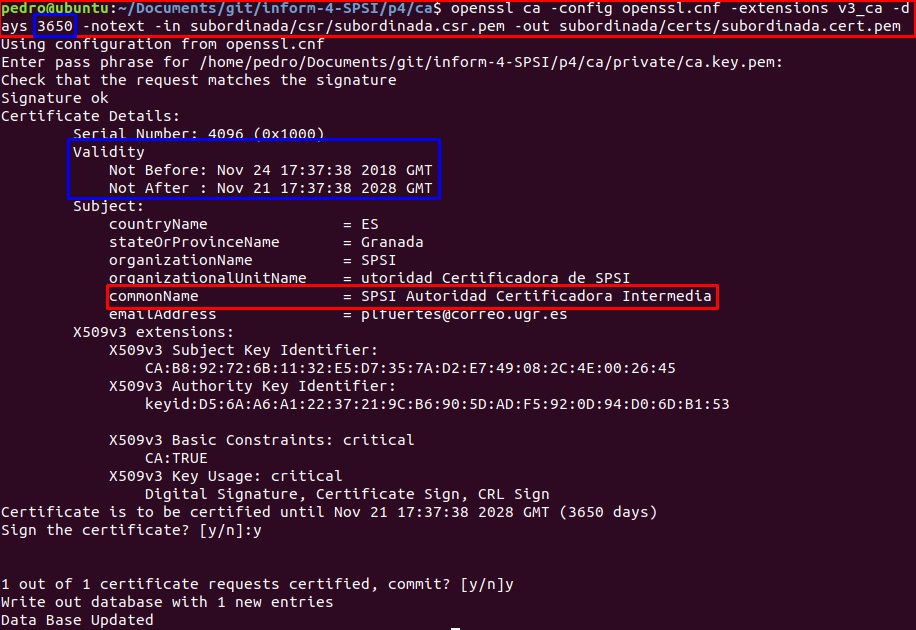
## Creamos el certificado de la CA subordinada



En este caso, como puede verse, cambia dónde se encuentra la clave privada de la CA .

Además, en este caso, la CA primaria tendrá que autorizar a la intermedia, por lo que se crea una CSR y se guarda en ../ca/subordinada/csr/subordinada.csr.pem

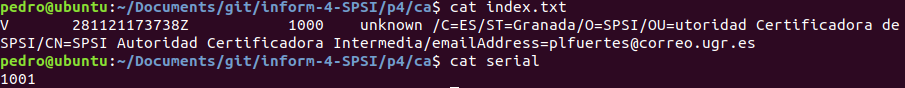
## Firmamos la autoridad de la CA subordinada con la autoridad raíz



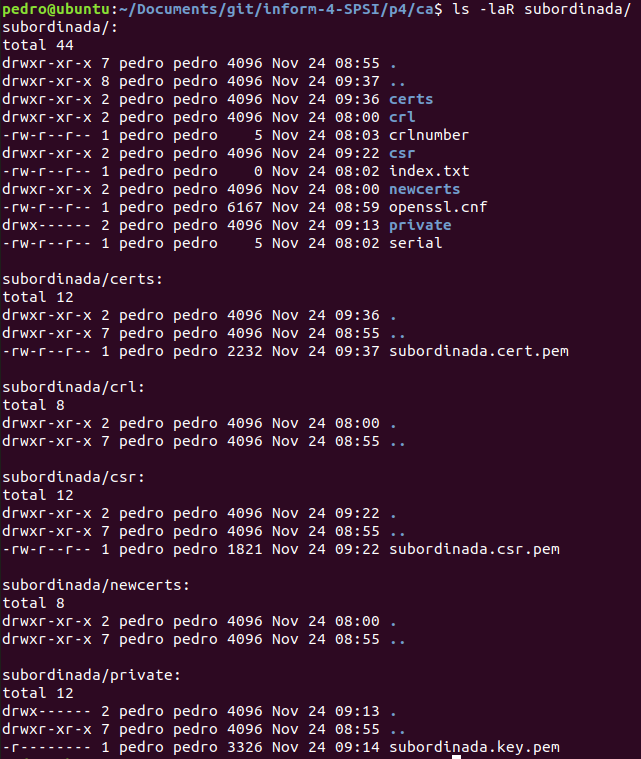
* - config: usamos el archivos de configuración de la CA raíz.
* -extensions: La extensión será v3\_ca como se ha especificado en el archivo openssl.cnf
* -days: el número de días en los que expirará el certificado. 10 años en este caso, como se puede ver en el recuadro azul
* -in: indica la CSR que se quiere firmar.
* -out: indica dónde se guardará el certificado

También se puede ver como se ha cogido el nombre común introducido en el punto 1 de esta sección.

Finalmente, si hacemos un cat del index o del serial podemos ver como en index se ha almacenado el certificado el certificado firmado y como el serial se ha incrementado en uno.

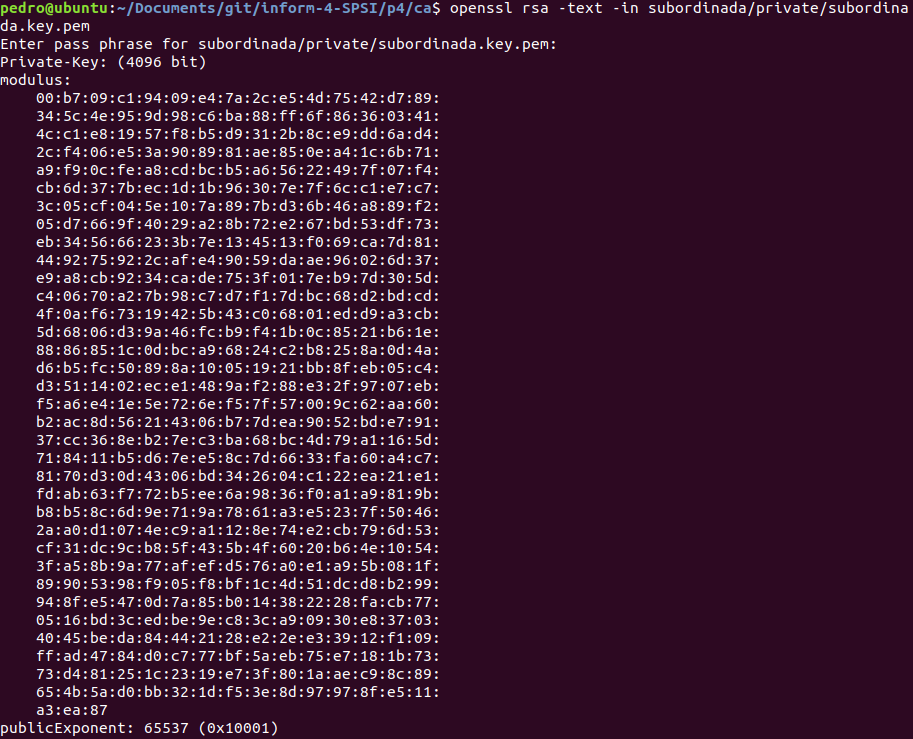


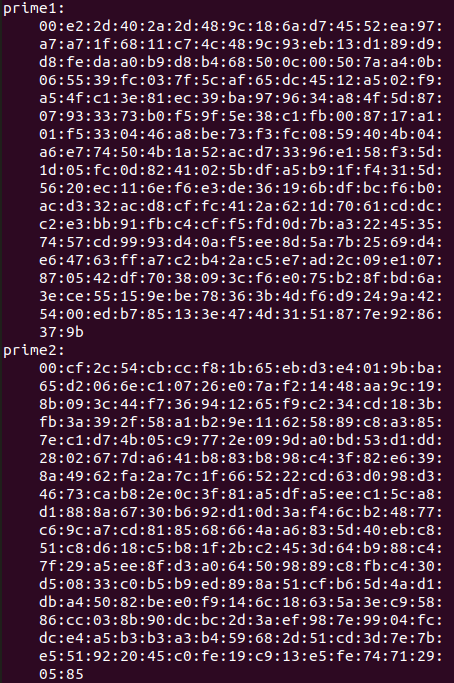
## Archivos creados

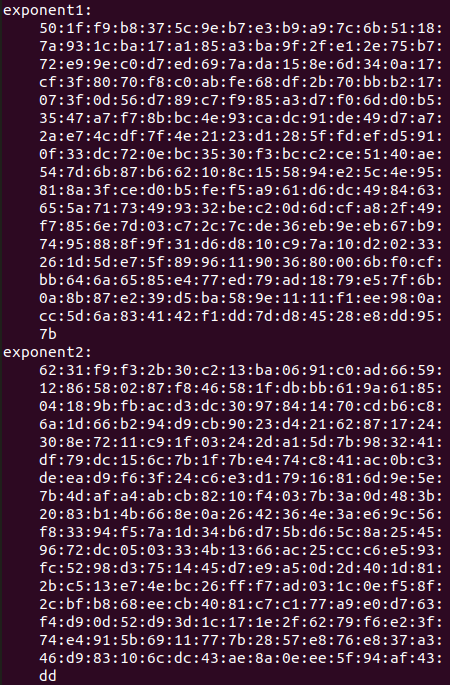
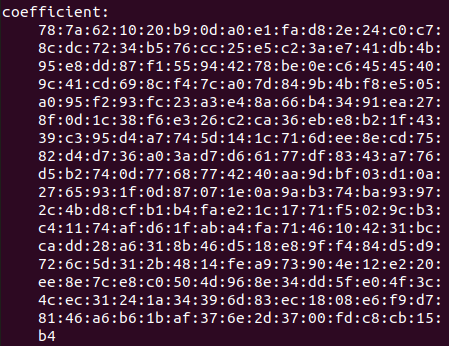


Se puede ver como a parte de la estructura de archivos creado en el punto 1 de esta sección, se ha creado una clave privada de la CA subordinada, así como una CSR en la carpeta csr de la CA subordinada y finalmente el certificado firmado por la CA raíz en la carpeta certs de la CA subordinada.

## Valor de las claves generadas

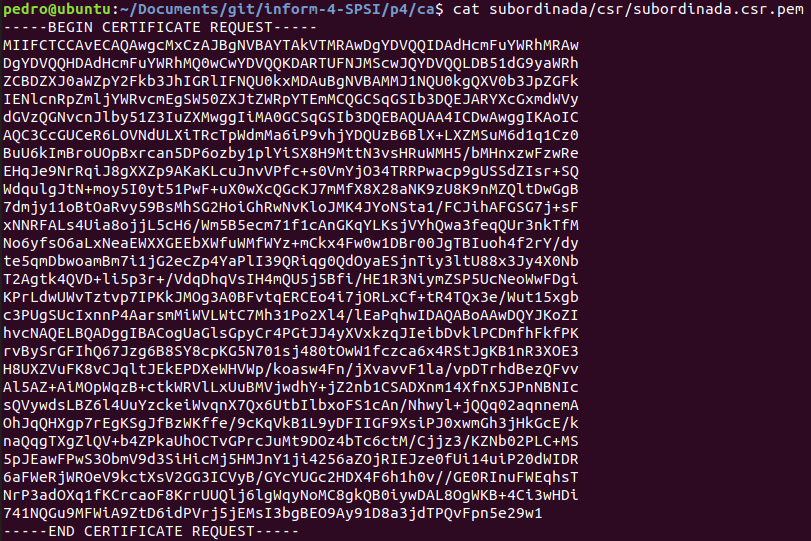
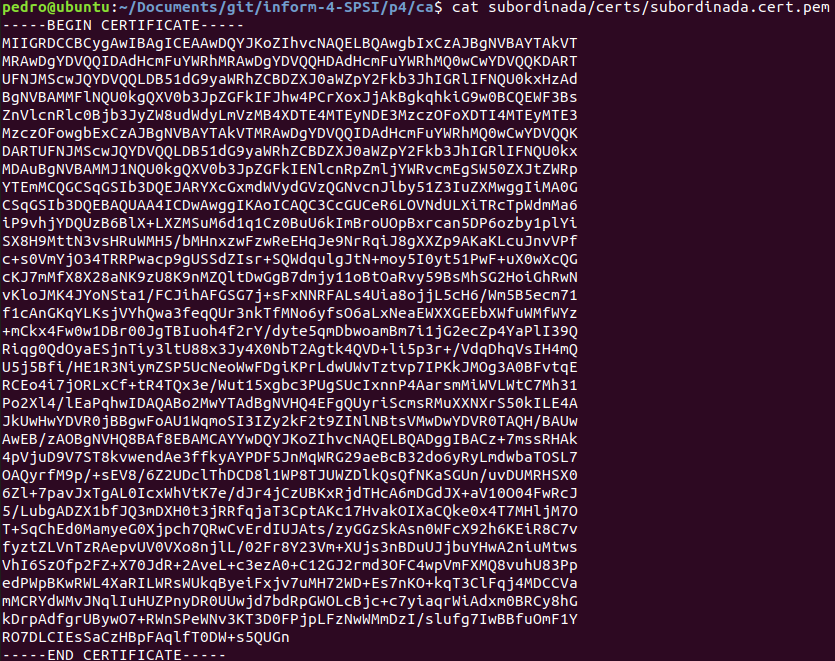




## Valor de los certificados generados

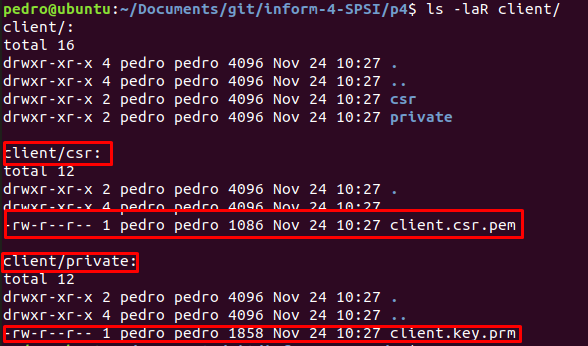
# Crear una solicitud de certificado que incluya la generación de claves en la misma

## Creación de la solicitud



* Openssl req: para indicar que voy a hacer una solicitud
* -newkey: crea una nueva clave privada para esta solicitud, en este caso rsa de 2048 bits.
* -keyout: indica dónde se va a guardar la clave privada que se acaba de generar.
* -out: indica dónde se va a guardar la solicitud
* - config: indica que configuración se va a coger, en este caso la de la CA raíz.

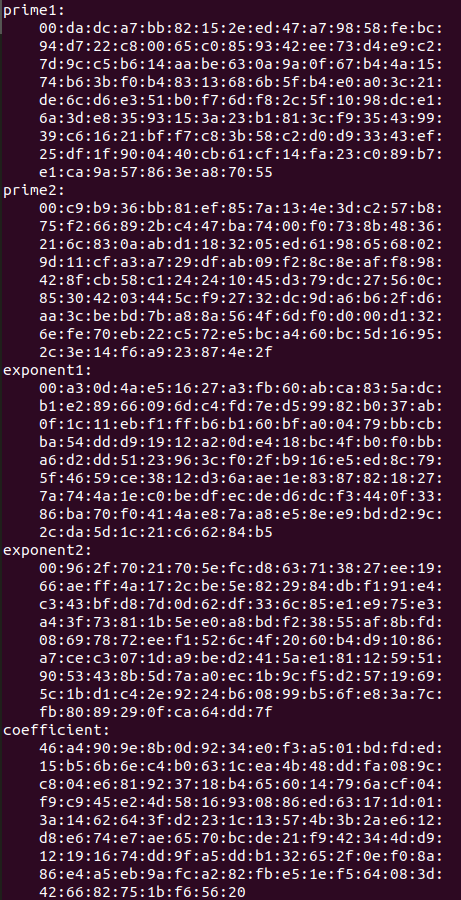
## Archivos generados

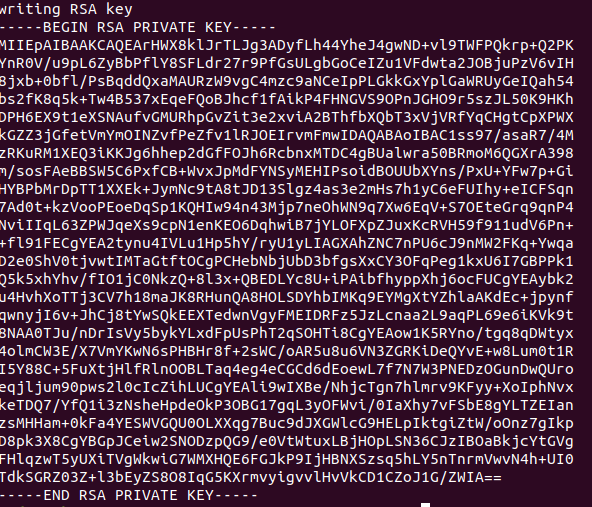


Como se puede ver, se han generado dos archivos, el de la solicitud del certificado en csr y la clave privada en private.

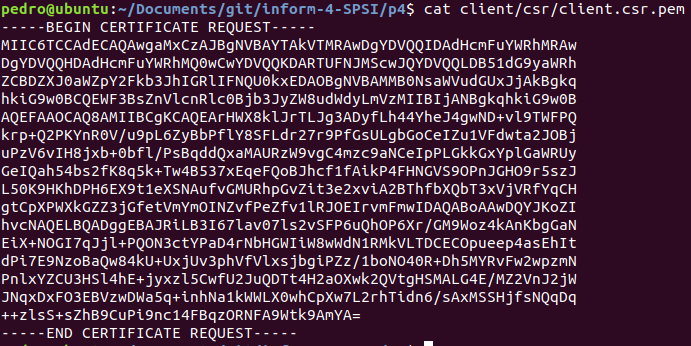
## Valor de la clave privada





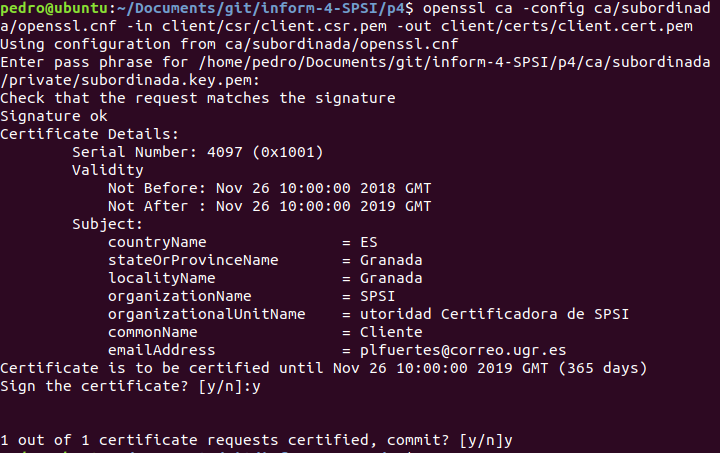


## Valor del certificado



# Cread un certificado para la solicitud anterior empleando la CA subordinada.

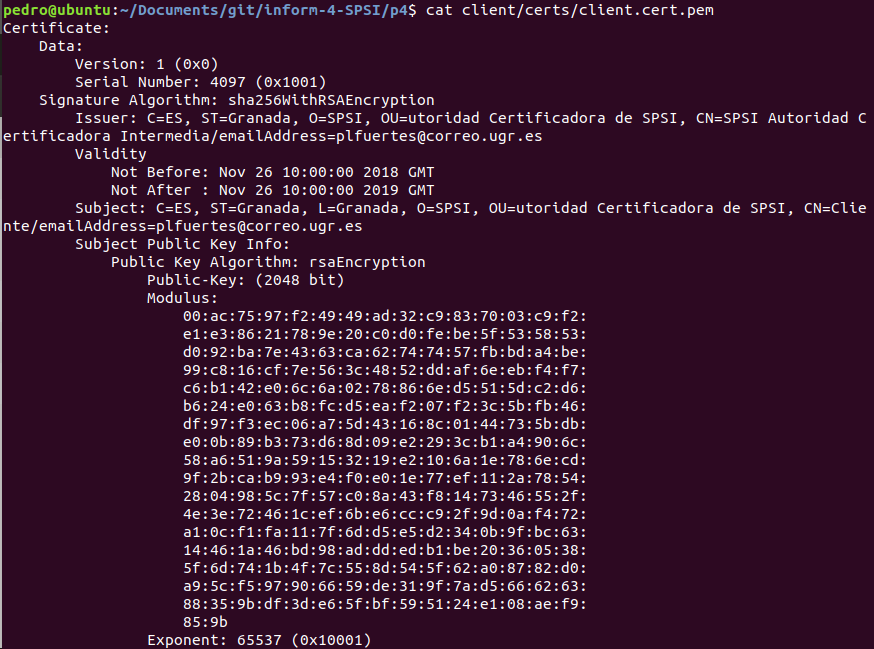
## Firmamos la petición

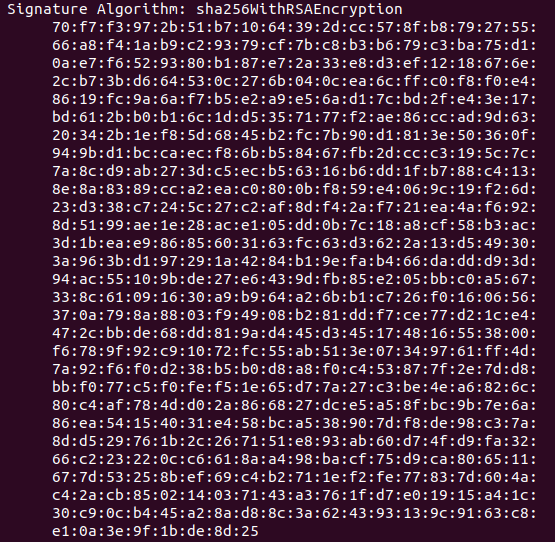


* -config: indica la configuración que se usará.
* -in: indica dónde está la solicitud a firmar
* -out: indica dónde se guardará el certificado

\* Como se puede ver, no se especifica dónde está la privada de la CA subordinada. Esto es debido que se obtiene del archivo de configuración openssl.cnf

## Mostramos los valores

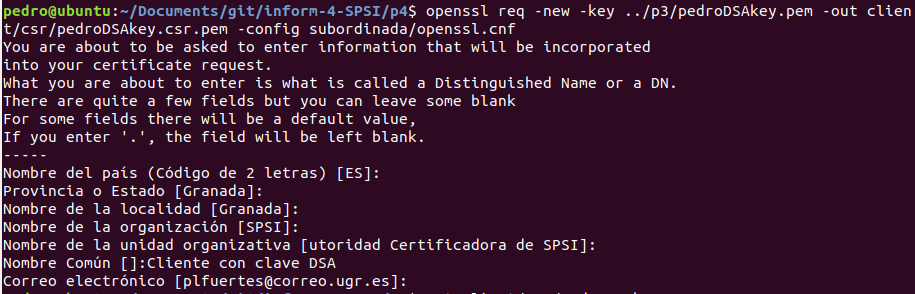






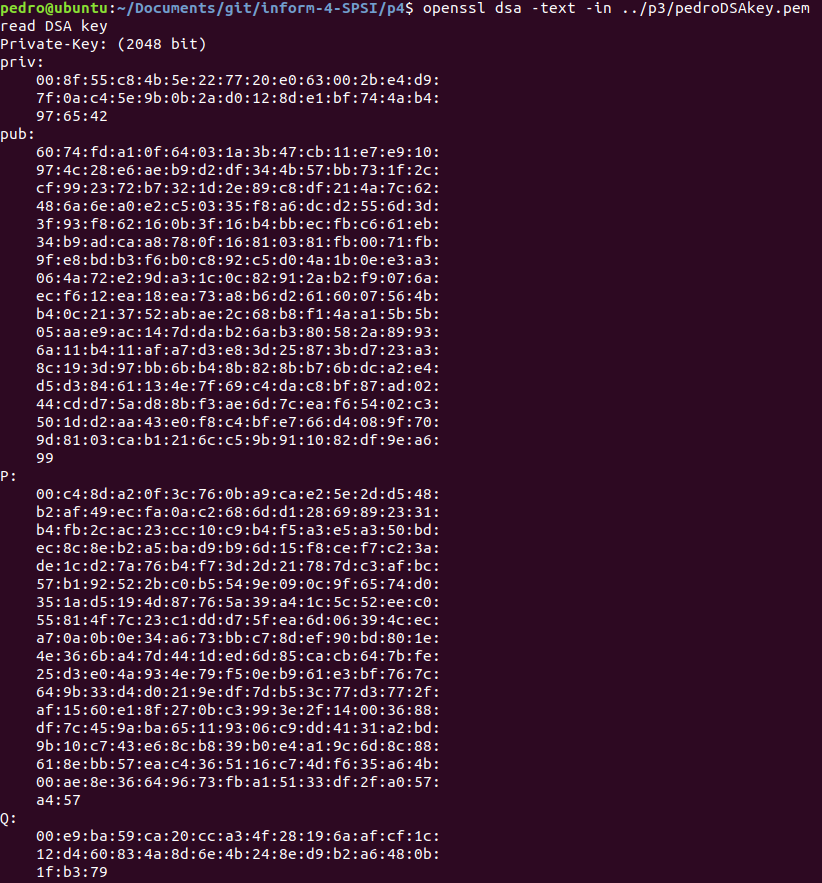
# Crear una solicitud de certificado para otra clave de las generadas en prácticas anteriores

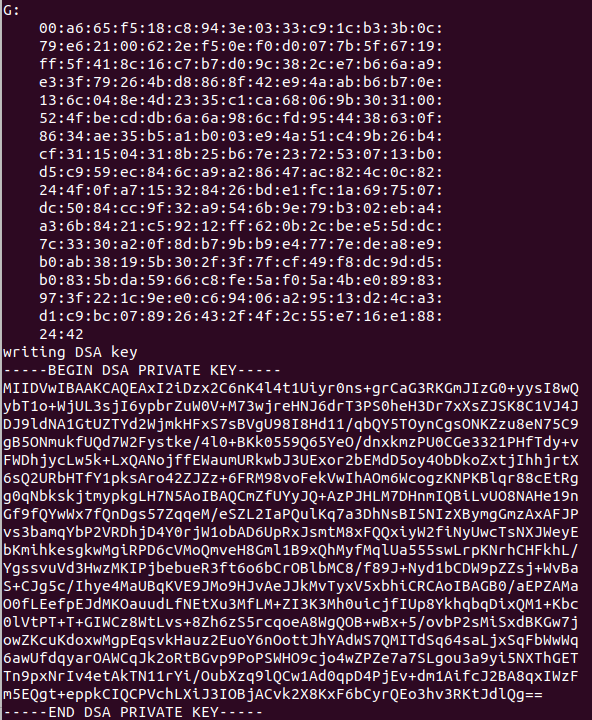
## Creación de la solicitud



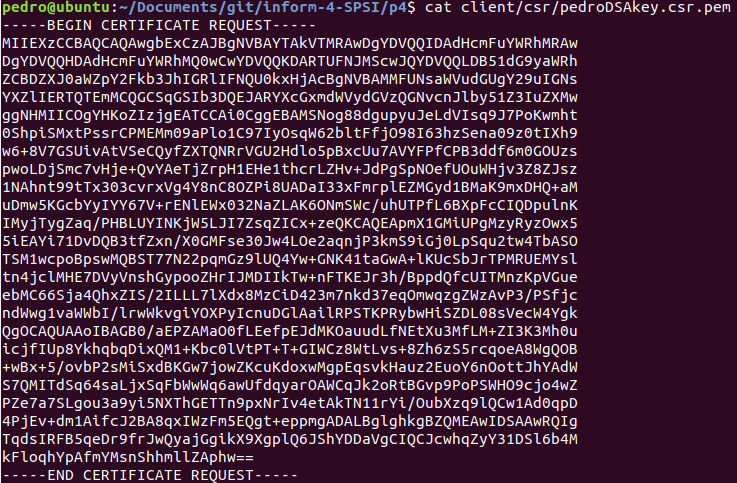
* Openssl req: Indica que vamos a trabajar con solicitudes de certificados
* -new: Que es una nueva solicitud
* -key: indica dónde se encuentra la clave para el certificado
* -out: dónde se va a guardar la certificación
* -config: la configuración para hacer la solicitud

## Valor de la clave usada



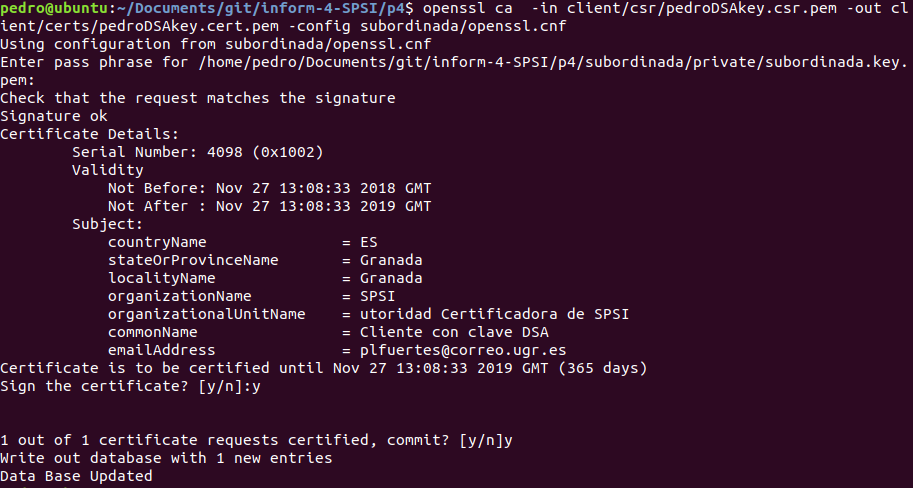


## Valor del certificado

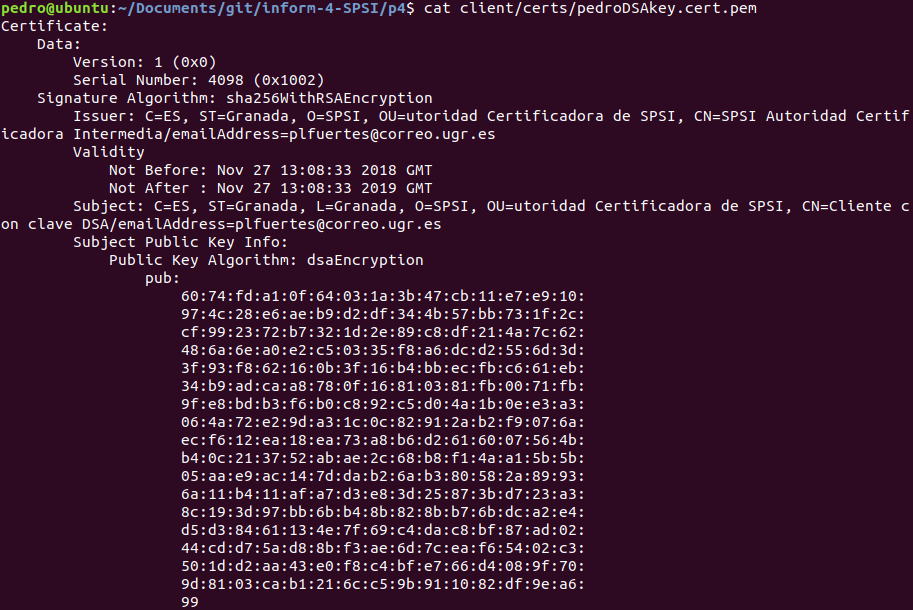


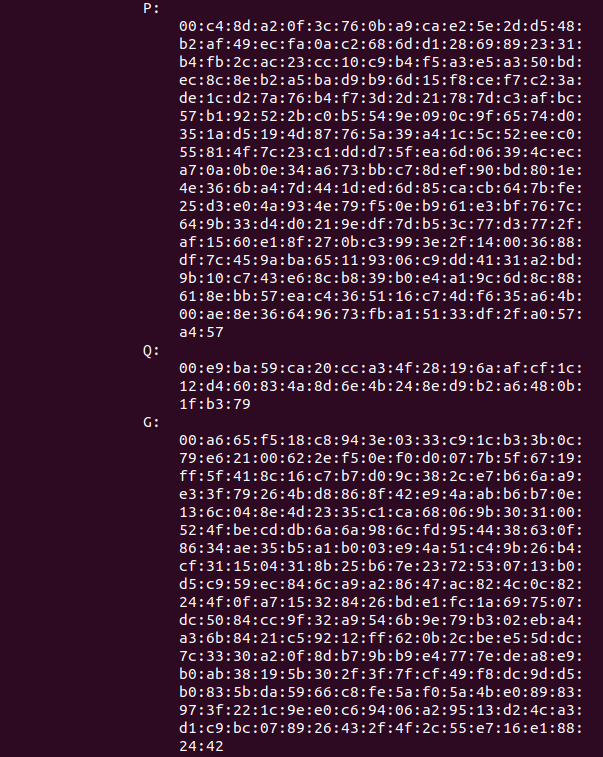
# Firmamos la petición anterior con la subordinada

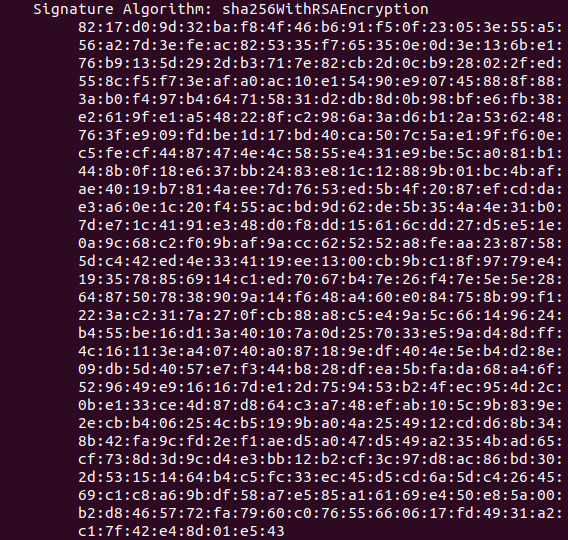
## Firma de la solicitud



## Valor del certificado





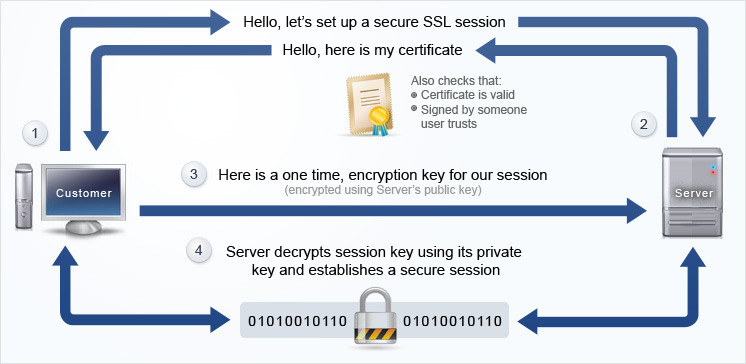




## Conclusiones

Se puede ver como al mostrar el valor del certificado aparecen los valores públicos de la clave (DSA en este caso) usada. Esto se utiliza para poder enviar una clave de sesión cifrada con la pública del servidor y que sólo ese servidor pueda descifrarla con la pública.

El proceso es más o menos este:



\* Imagen obtenida de [aquí](https://panel.rootway.com/knowledgebase.php?action=displayarticle&id=2).