Propuesta de un Mecanismo de Seguridad para el Intercambio de Datos de Usuario en Redes de Próxima Generación

Universidad del Cauca

Jaime Andrés Oliva Ortega
Fabio Joaquín Fuertes Montenegro

ANEXO B
GENERACIÓN DE CERTIFICADOS X.509

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán, Octubre de 2009

TABLA DE CONTENIDO

LISTA DE FIGURAS

1. Introducción

Muchas implementaciones de IPsec permiten el uso de certificados X.509, para la autenticación de los participantes en una comunicación segura, estos son los mismos certificados que se emplean para la implementación de SSL en el protocolo HTTP.

A continuación se mostrará la creación de una CA (Autoridad Certificadora) y la petición y firma de los certificados de host, requeridos por strongSwan para la autenticación entre los equipos del prototipo. Lo anterior se realizará en el host de nombre *akira,* el cual cuenta con sistema operativo Debian 5 y se utilizará la herramienta OpenSSL, además del script CA.pl instalado por defecto en dicha distribución, el cual automatiza las tareas básicas de OpenSSL.

2. Configuraciones preliminares

En primer lugar se verifica la instalación del paquete OpenSSL como lo muestra la Figura 1. .

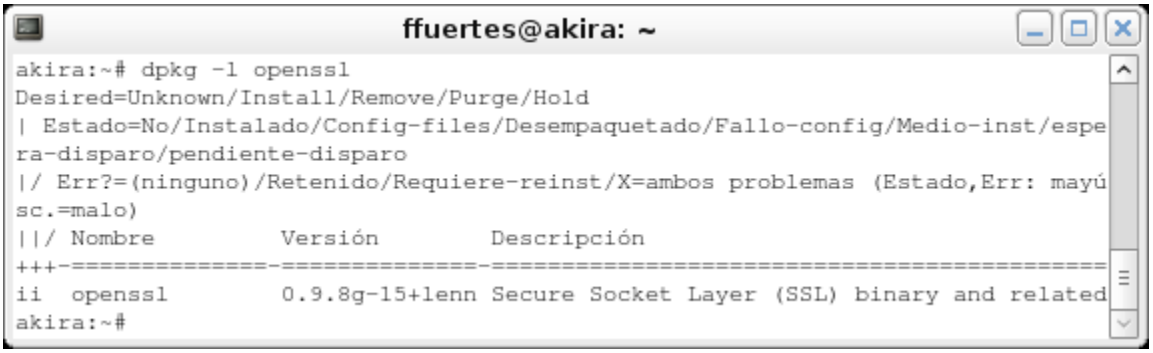

Figura 1. Verificación de la Instalación de OpenSSL

Ahora en el archivo de configuración /etc/ssl/openssl.cnf se realizan las modificaciones mostradas a continuación.
Cambiar el directorio por defecto:

```
[[ CA_default ]
dir =./akiraCA
```

Cambiar opciones de la configuración de petición de certificados:

```
[ req_distinguished_name ]
countryName = CO
countryName_default = CO
#
```

```
stateOrProvinceName = Cauca
stateOrProvinceName_default = Cauca

#
localityName_default = Popayán
#
0.organizationName = Unicauca
0.organizationName_default = Unicauca
#
organizationalUnitName = IMSeg
organizationalUnitName_default = IMSeg
#
commonName = Common Name
#
emailAddress = admin@imseg.unicauca.edu.co
```

Modificar el script CA.pl ubicado en el directorio /usr/lib/ssl/misc/ para que se adapte a la configuración de OpenSSL hecha previamente:

```
$CATOP="./akiraCA";
$CAKEY="akiraCakey.pem";
$CAREQ="akiraCareq.pem";
$CACERT="akiraCacert.pem";
```

Como paso adicional, para facilitar el llamado al script CA.pl se crea un enlace simbólico con el siguiente comando:

```
ln -s /usr/lib/ssl/misc/CA.pl /usr/bin/CA.pl
```

Por último se crea el directorio de trabajo, donde se guardarán todos los archivos de la CA incluyendo su llave pública y privada, además de las peticiones y certificados firmados de los clientes:

```
mkdir /var/openssl
cd /var/openssl
```

3. Creación de la Autoridad Certificadora

Utilizando las configuraciones previas, se realiza la petición para crear una nueva CA mediante el comando:

```
CA.pl –newca
```

Posteriormente se ingresa la contraseña de la CA, el cual es muy importante pues con él se puede acceder a la clave privada akiraCakey.pem con la cual se firman las peticiones de certificados de los clientes. De la misma manera se establece la información básica que describe a la CA, este proceso se muestra en la Figura 2. .

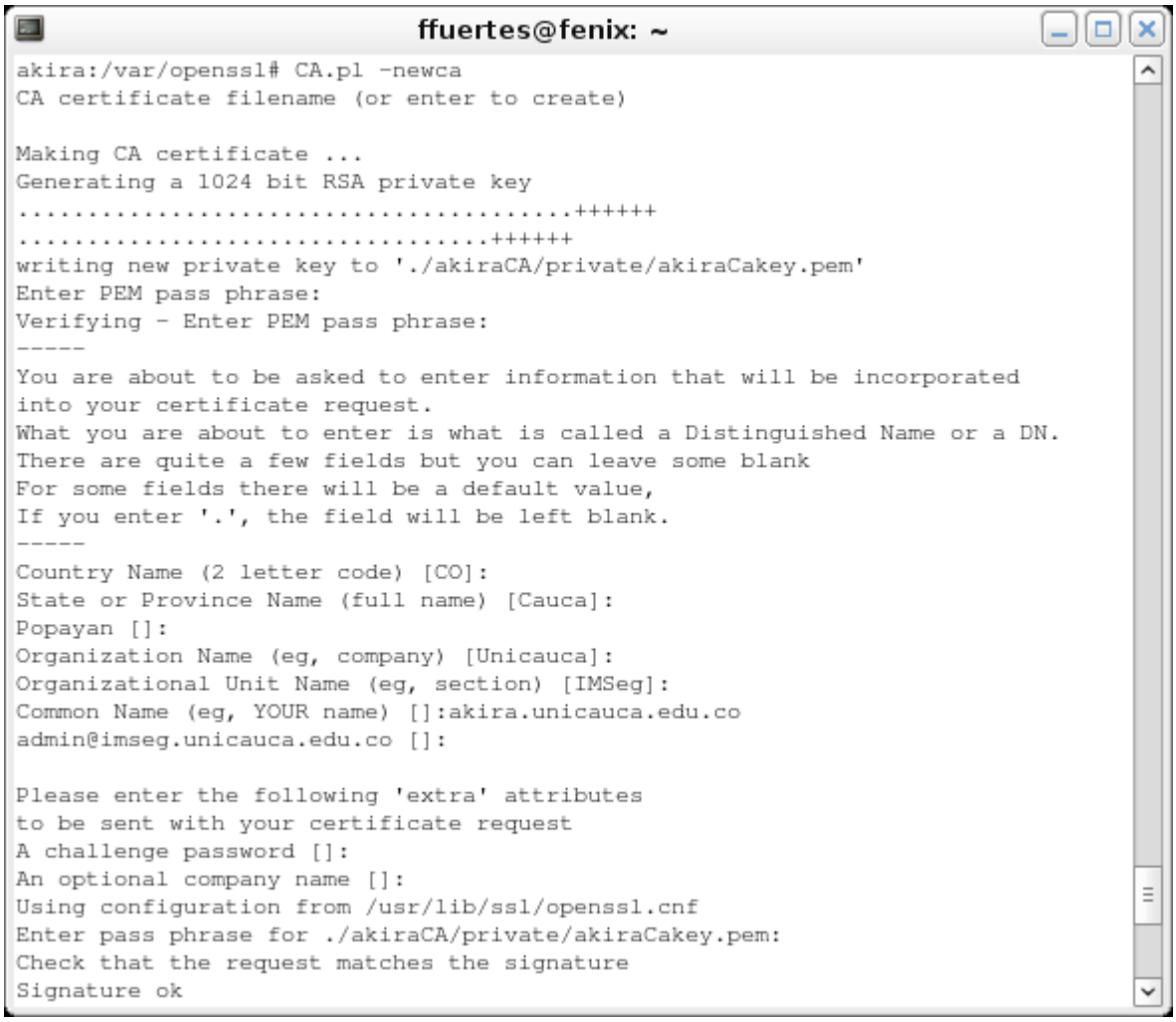

Figura 2. Creación de una Autoridad Certificadora utilizando OpenSSL

Posteriormente, la 0 muestra los detalles del certificado de la CA creado, el cual se utilizará en los hosts clientes para verificar la autenticidad de los certificados de sus pares y con esto dar vía libre al establecimiento de un túnel IPsec.

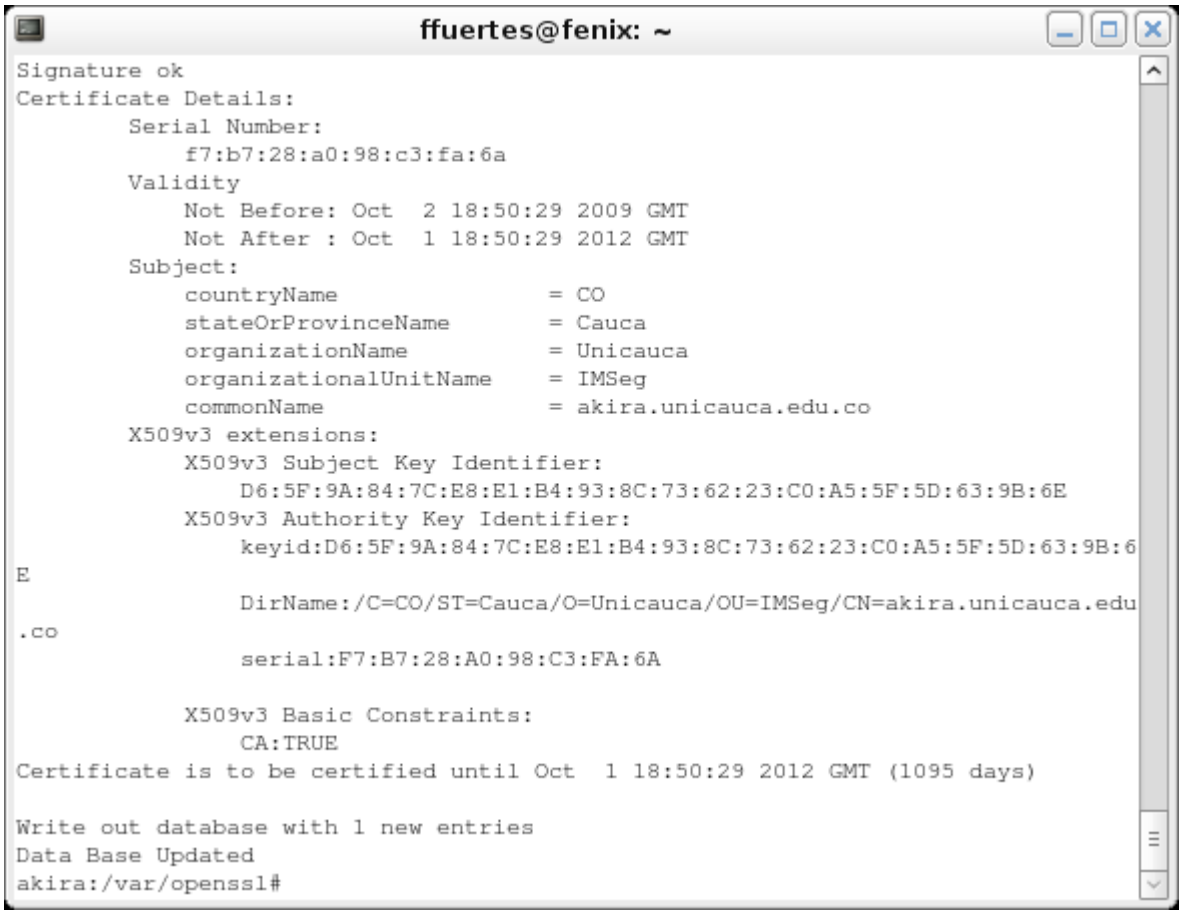

Figura 3. Datos del certificado de la CA recientemente creada

4. Petición de certificado de cliente

Para esta petición se tomará como ejemplo el host *fenix;* el comando utilizado es:

```
CA.pl -newreq
```

Como se puede observar en la Figura 4. , los datos ingresados para la petición son similares a los de la creación de la CA (ver Figura 2.); sin embargo el atributo mas importante en este paso es el *Common Name*, pues se trata del identificador único del equipo al cual se le va a generar el certificado; este debe ser distinto para cada certificado de host, en el caso del ejemplo en cuestión se ha elegido el nombre distintivo del equipo con el dominio de la Universidad: *fenix.unicauca.edu.co.* Igualmente es necesario ingresar una contraseña para la clave privada del host.

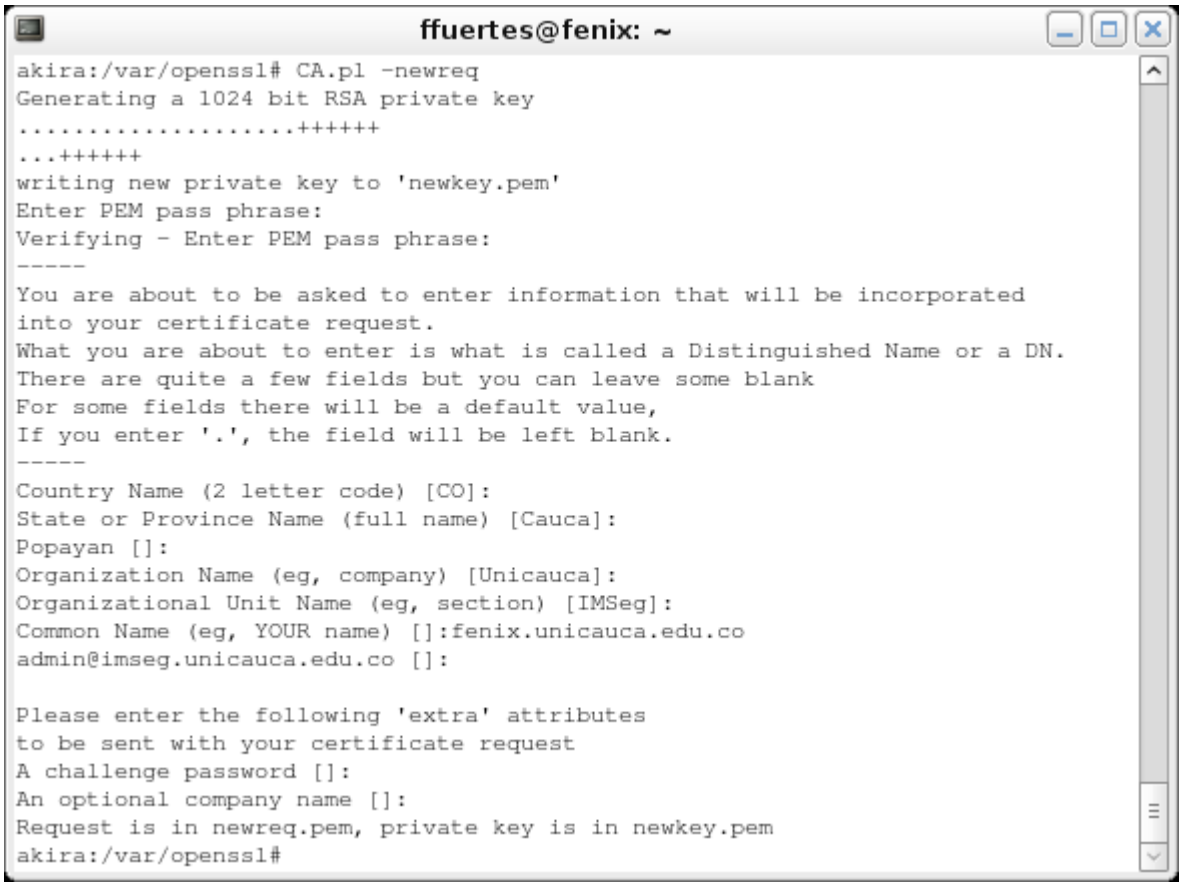

Figura 4. Petición de certificado de cliente

Como resultado de la petición hecha previamente se crean los archivos *newkey.pem* y *newreq.pem* (ver Figura 5.) los cuales corresponden a la clave privada del host y al archivo con los datos de petición de certificado respectivamente.

Figura 5. Archivos de petición de certificado

5. Firma del certificado de cliente

Como elemento final en el proceso, se solicita a la CA que firme la petición del certificado creado en el paso anterior, esto se hace utilizando el siguiente comando:

```
CA.pl -sign
```

En este paso se solicita la clave de la CA establecida anteriormente y si esta es correcta se firma la petición contenida en *newreq.pem* y se muestran los detalles del certificado, como puede apreciarse en la Figura 6. .

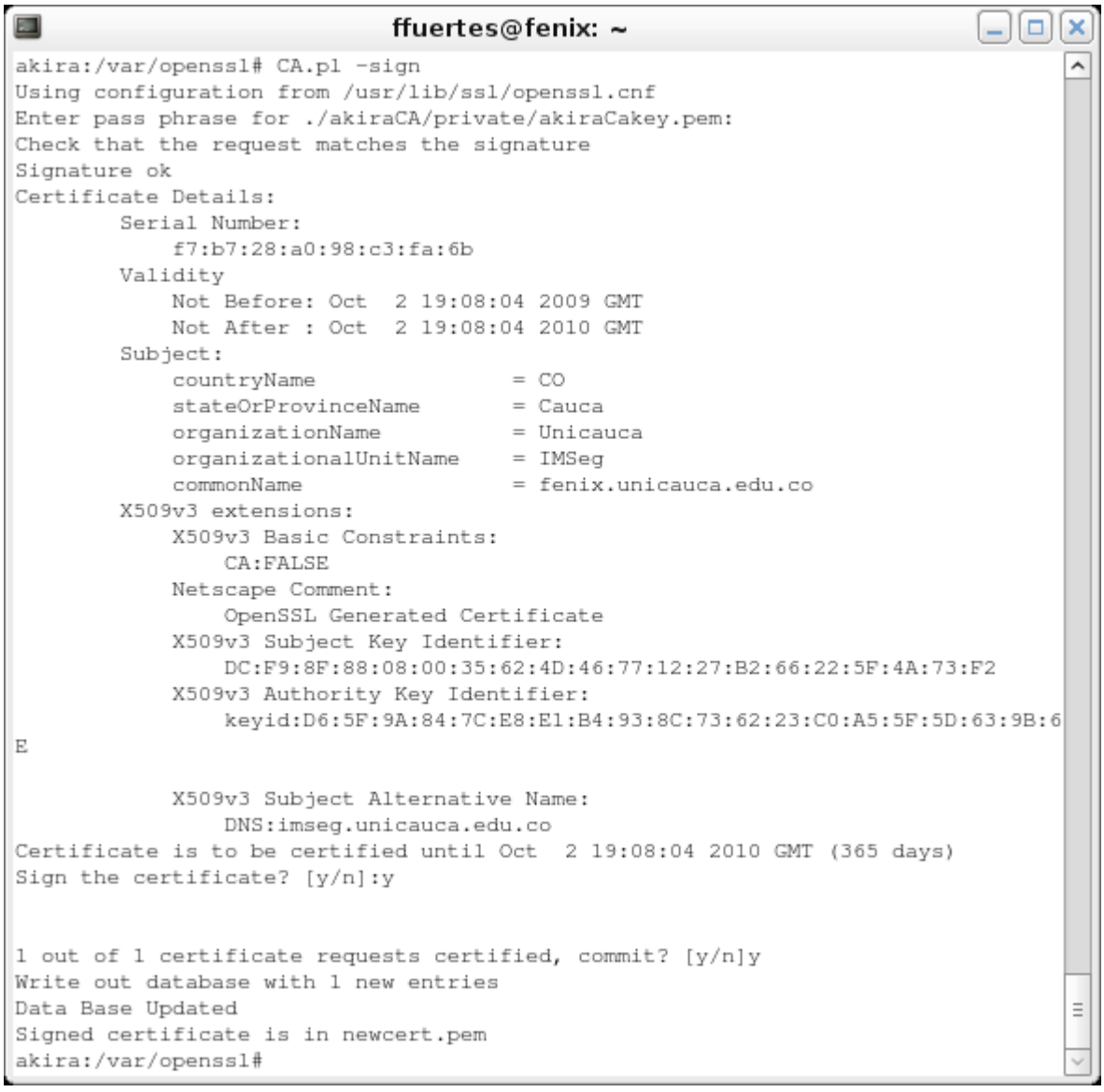

Figura 6. Firma del certificado del cliente

Ahora, se listan los archivos del directorio de trabajo (ver Figura 7.), mostrando que se ha creado el archivo *newcert.pem,* el cual corresponde al certificado del host *fenix* firmado por la CA *akira.*

Figura 7. Verificación de la creación del certificado newcert.pem

Por practicidad, se debe cambiar el nombre tanto del certificado como de la llave privada del cliente para que no se confundan con nuevos certificados y sean descriptivos del equipo que representan. El procedimiento de petición y firma de certificados se debe siguió para todos los hosts del prototipo. La Figura 8. muestra el listado de los certificados de los cuatro equipos con su respectiva llave privada.

Figura 8. Certificados de los equipos del prototipo

Por último, estos certificados se envían por medio seguro hacia cada uno de los hosts para realizar la configuración de strongSwan y establecer las comunicaciones IPsec.