Evidencias Interoperabilidad XRoad Julio 2017

gob.mx

Índice

Control de versiones del documento

Versión	Fecha	Descripción de la actualización	Elaborado por
1.0	10/07/2017	Elaboración de las evidencias correspondientes a la instalación y configuración de servicios relacionados al sistema X-Road, SignServer	Leví Durán Torres

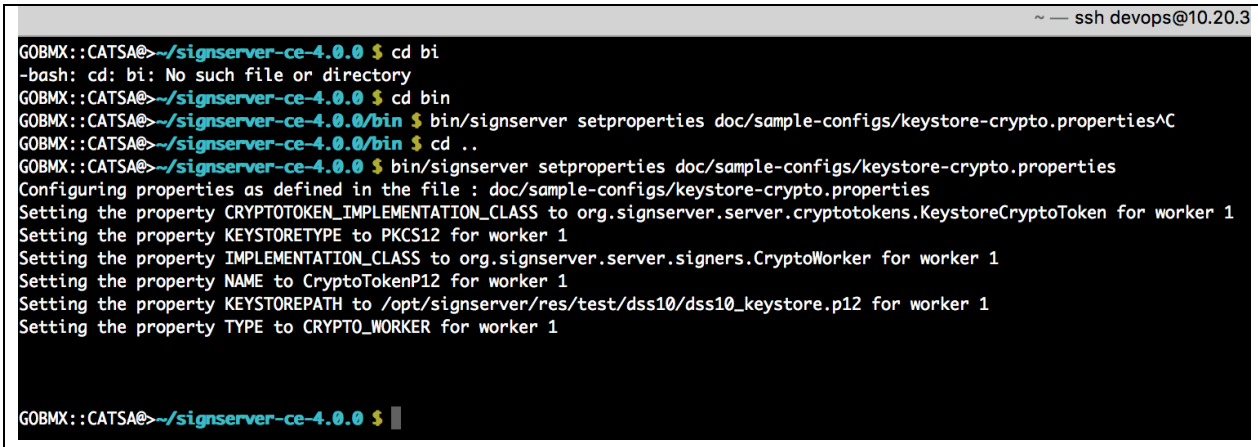
1 Objetivo.

Este documento está destinado a dar una representación gráfica de los elementos que soportan la instalación y configuración de servicios relacionados al sistema X-Road.

2 Evidencias.

1. Evidencia: Configuración de entidades finales en EJBCA

SignServer
A continuación se mencionan las instrucciones para poder configurar un timestamp signer en Signserver Configuración cryototoken para timestamp signer 1. Primero se debe cargar la configuración del crypto token con el comando setproperties. Este comando carga todas las propiedades que están dentro del archivo properties que recibe como argumento. Antes de cargar la configuración del crypto token asegurarse de que el modulo tsa esta compilado dentro de la instalación de SignServer, esto se hace verficando que la propiedad includemodulesinbuild tenga el valor de true dentro del archivo signserver_deploy.properties 2. Ejecutar el comando setproperties con el archivo keystore-crypto.properties como argumento $ bin/signserver setproperties doc/sample-configs/keystore-crypto.properties



Nota: El valor de 1 se refiere al id del worker de SignServer que son los procesos que se encargan de realizar la firma, en este caso del estampillado de tiempo

3. Ahora se debe actualizar la propiedad de KEYSTOREPATH del crypto token para que apunte a un keystore de tipo PKCS#12 que contenga contiene llaves y un certificado para la firma del estampillado de tiempo. También se debe actualizar el password y setear la llave default del crypto token

   $bin/signserver setproperty 1 KEYSTOREPATH $SIGNSERVER_HOME/res/test/dss10/dss10_tssigner1.p12
   $ bin/signserver setproperty 1 KEYSTOREPASSWORD foo123
   $ bin/signserver setproperty 1 DEFAULTKEY "TS Signer 1"
   $ bin/signserver reload 1

```
GOBMX::CATSA@>~/signserver-ce-4.0.0 $ bin/signserver getstatus complete all
Current version of server is : SignServer CE 4.0.0


The Global Configuration of Properties are :

  No properties exists in global configuration

  The global configuration is in sync with the database.



Status of CryptoWorker with id 1 (CryptoTokenP12) is:
   Worker status : Offline
   Token status  : Offline

   Errors:
      - Error Crypto Token is disconnected

   Worker properties:
      KEYSTORETYPE=PKCS12

      DEFAULTKEY=TS Signer 1

      KEYSTOREPATH=/home/devops/signserver-ce-4.0.0/res/test/dss10/dss10_tssigner1.p12

      KEYSTOREPASSWORD=letme1n1234

      NAME=CryptoTokenP12

      CRYPTOTOKEN_IMPLEMENTATION_CLASS=org.signserver.server.cryptotokens.KeystoreCryptoToken

      IMPLEMENTATION_CLASS=org.signserver.server.signers.CryptoWorker

      TYPE=CRYPTO_WORKER


   Authorized clients (serial number, issuer DN):


Status of Signer with id 2 (TimeStampSigner) is:
   Worker status : Offline
   Token status  : Offline
   Signings      : 0

   Errors:
      - No signer certificate available
      - Certificate chain not available
      - No signer certificate available
      - Error Crypto Token is disconnected

   Worker properties:
      CRYPTOTOKEN=CryptoTokenP12
```

Nota: El password para el keystore debe ser tal cual foo123

4. Se procede a cargar la configuración de referencia del signer de timestamp (estampillado de tiempo), esto hará que se genere otro worrker con id de 2

   $ bin/signserver setproperties doc/sample-configs/timestamp.properties

```
GOBMX::CATSA@>~/signserver-ce-4.0.0 $ bin/signserver setproperties doc/sample-configs/timestamp.properties
Configuring properties as defined in the file : doc/sample-configs/timestamp.properties
Setting the property CRYPTOTOKEN to CryptoTokenP12 for worker 2
Setting the property IMPLEMENTATION_CLASS to org.signserver.module.tsa.TimeStampSigner for worker 2
Setting the property DEFAULTKEY to ts00003 for worker 2
Setting the property NAME to TimeStampSigner for worker 2
Setting the property ACCEPTANYPOLICY to true for worker 2
Setting the property DEFAULTTSAPOLICYOID to 1.3.6.1.4.1.22408.1.2.3.45 for worker 2
Setting the property AUTHTYPE to NOAUTH for worker 2
Setting the property TYPE to PROCESSABLE for worker 2


GOBMX::CATSA@>~/signserver-ce-4.0.0 $
```

5. Setear la propiedad de TS_DEFAULT_KEY del worker 2

   $ bin/signserver setproperty 2 DEFAULTKEY "TS Signer 1"

```
GOBMX::CATSA@>~/signserver-ce-4.0.0 $ bin/signserver setproperty 2 DEFAULTKEY "TS Signer 1"
Setting the property DEFAULTKEY to TS Signer 1 for worker 2

See current configuration with the getconfig command, activate it with the reload command
```

6. Se actualiza la configuración del worker 2
   $ bin/signserver reload 2

```
GOBMX::CATSA@>~/signserver-ce-4.0.0 $ bin/signserver reload 2
SignServer reloaded successfully

Current configuration is now activated.

GOBMX::CATSA@>~/signserver-ce-4.0.0 $
```

7. Se procede a crear CSR para el signer de timestamp

```
GOBMX::CATSA@>~/signserver-ce-4.0.0 $ bin/signserver generatecertreq 2 "C=MX,CN=TS Signer" SHA256WithRSA tscsr.req
PKCS10 Request successfully written to file tscsr.req
GOBMX::CATSA@>~/signserver-ce-4.0.0 $
```

8. Se debe

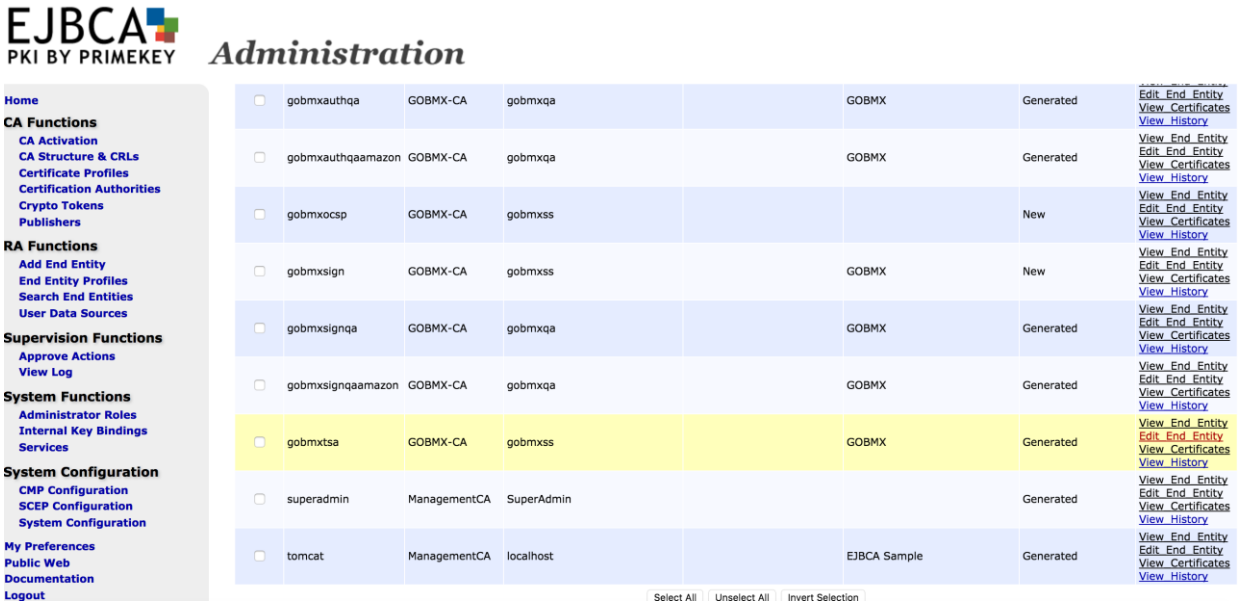
2. Evidencia: Generación de certificado TSA desde EJBCA

EJBCA

A continuación se mencionan las instrucciones para poder generar un certificado firmado dentro la consola pública de EJBCA con un CSR creado desde SignServer

1. Verificar que ya existe una entidad final dentro de la consola de admin de EJBCA con el propósito de generar certificados firmados de tipo timestamp

   https://10.20.37.221:8443/ejbca/adminweb/



Edit End Entity

End Entity Profile	GOBMX-TSA-EP	Required
Status	Generated ⇕ Save	
Username	**gobmxtsa**	☑
Password (or Enrollment Code)		☑
Confirm Password		
Maximum number of failed login attempts	○ ◉ Unlimited	
Remaining login attempts	☐ Reset login attempts	
E-mail address	@	☐
Subject DN		
CN, Common name	gobmxss	☑
C, Country (ISO 3166)	MX	☐
O, Organization	GOBMX	☐
serialNumber, Serial number (in DN)	gobmx-dev/coress/CORE	☐
Main certificate data		
Certificate Profile	GOBMX-TSA ⇕	☑
CA	GOBMX-CA ⇕	☑
Token	User Generated ⇕	☑
	Save Close	

Nota: También se puede comprobar que la entidad final de gobmxtsa tiene llave con un uso de tipo timestamp viendo las propiedades del perfil de certificado al que está referenciado para GOBMX TSA (opción Certificate Profiles)

EJBCA PKI BY PRIMEKEY *Administration*

Home
CA Functions
CA Activation
CA Structure & CRLs
Certificate Profiles
Certification Authorities
Crypto Tokens
Publishers
RA Functions
Add End Entity
End Entity Profiles
Search End Entities
User Data Sources
Supervision Functions
Approve Actions
View Log
System Functions
Administrator Roles
Internal Key Bindings
Services
System Configuration
CMP Configuration
SCEP Configuration
System Configuration
My Preferences

Edit

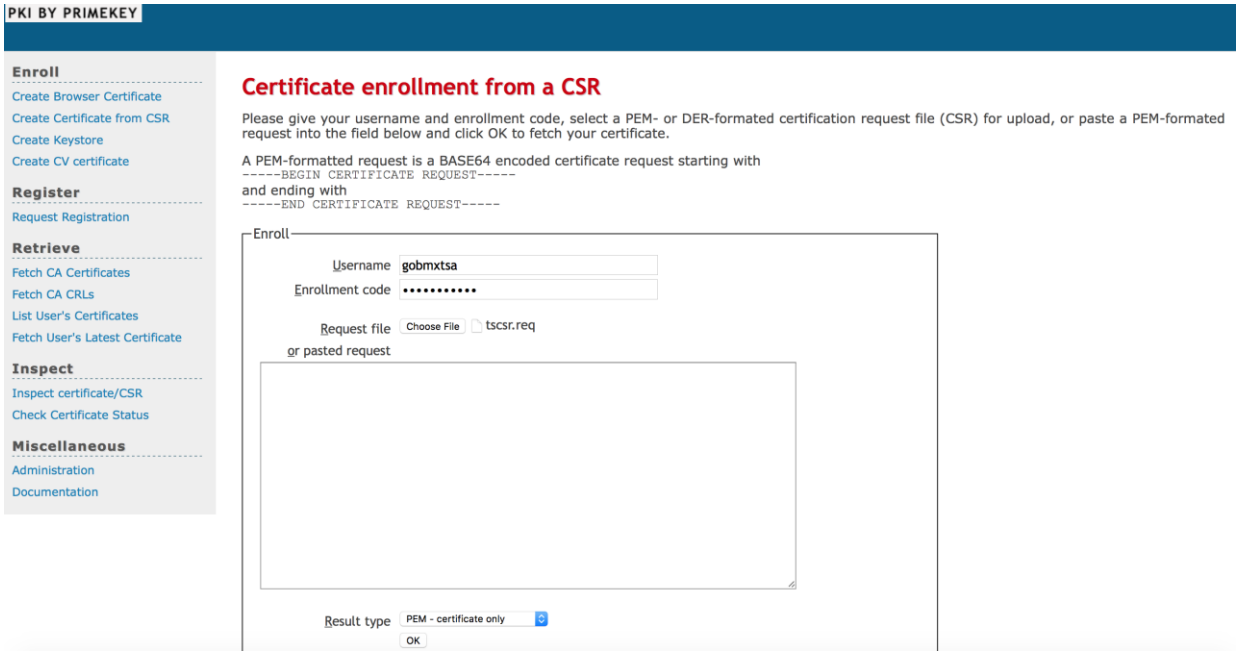
Certificate Profile: GOBMX-TSA

	Back to Certificate Profiles
Certificate Profile Id	169504415
Type	End Entity ✓Sub CA Root CA
Available bit lengths	0 bits 192 bits 239 bits 256 bits 384 bits
Signature Algorithm	Inherit from issuing CA ⇕
Validity(*y *mo *d) or end date of the certificate [?]	730d ISO 8601 date:[yyyy-MM-dd HH:mm:ssZZ]: '2017-07-13 05:15:37+00:00'
Permissions	
Allow validity override [?]	☐ Allow
Allow extension override [?]	☐ Allow
Allow certificate serial number override [?]	☐ Allow No unique index for (issuerDN,serialNumber) on database table 'CertificateData'. "Allow certificate serialnumber override" not allowed.
Allow subject DN override [?]	☐ Allow
Allow Key Usage Override	☐ Allow
Allow back dated revocation [?]	☐ Allow

2. Ahora se debe proceder a la liga de la consola pública de EJBCA

   http://10.20.37.221:8080/ejbca/

3. Elegir la opción de Create certificate from CSR y utilizar los datos de la entidad final de gobmxtsa cargando también el CSR creado en SignServer en la sección anterior



4. Al dar click en OK se generará el certificado firmado en formato PEM.

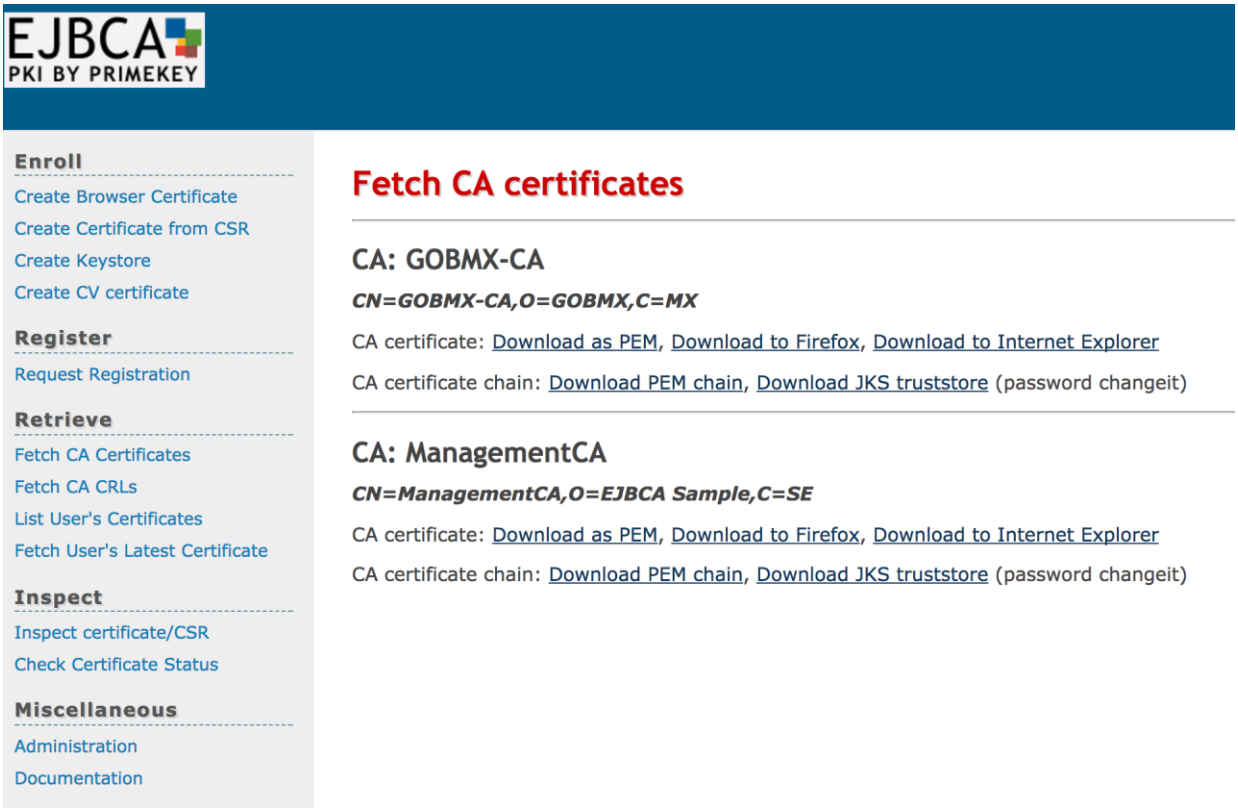
Ligas relacionadas:

https://github.com/GOBMX/xroad-source
https://www.signserver.org

3. Evidencia: Carga del certificado firmado dentro de la configuración del worker de SignServer

SignServer/EJBCA/X-Road

1. Regresar a la línea de comandos de SignServer y crear la cadena de certificados para que esta sea cargada al worker de timestamp. La cadena de certificados se compone de dos archivos. El primero es certificado de la CA como tal el cual se obtiene dando click en la opción de Fetch CA certificates dentro de la consola publica de EJBCA (en este caso es la CA de GOBMX-CA)



El segundo archivo es el que se generó en la sección anterior desde la consola pública de EJBCA. Se deben concatenar ambos archivos para poder crear la cadena de certificados

```
GOBMX::CATSA@>~/signserver-ce-4.0.0 $ cat /home/devops/gobmxtsa.pem /home/devops/GOBMXCA.pem > /home/devops/certchain.pem
```

2. Después se carga la cadena de certificados en la configuración del timestamp signer

```
GOBMX::CATSA@>~/signserver-ce-4.0.0 $ bin/signserver uploadsignercertificate 2 glob /home/devops/gobmxtsa.pem
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Uploading the following signer certificate  :

            Subject DN:     CN=gobmxss,SN=gobmx-dev/coress/CORE,O=GOBMX,C=MX
            Serial number:  100f3bce3ee6b072
            Issuer DN:      CN=GOBMX-CA,O=GOBMX,C=MX
            Valid from:     2017-04-11 20:54:33 UTC
            Valid until:    2018-03-16 19:26:21 UTC
GOBMX::CATSA@>~/signserver-ce-4.0.0 $ bin/signserver uploadsignercertificatechain 2 glob /home/devops/certchain.pem
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Uploading the following signer certificates  :

            Subject DN:     CN=gobmxss,SN=gobmx-dev/coress/CORE,O=GOBMX,C=MX
            Serial number:  100f3bce3ee6b072
            Issuer DN:      CN=GOBMX-CA,O=GOBMX,C=MX
            Valid from:     2017-04-11 20:54:33 UTC
            Valid until:    2018-03-16 19:26:21 UTC


            Subject DN:     CN=GOBMX-CA,O=GOBMX,C=MX
            Serial number:  d63a34ad152991b
            Issuer DN:      CN=GOBMX-CA,O=GOBMX,C=MX
            Valid from:     2017-03-16 19:26:21 UTC
            Valid until:    2018-03-16 19:26:21 UTC


GOBMX::CATSA@>~/signserver-ce-4.0.0 $
```

3. Se procede a actualizar la configuración de todos los workers

$bin/signserver reload all

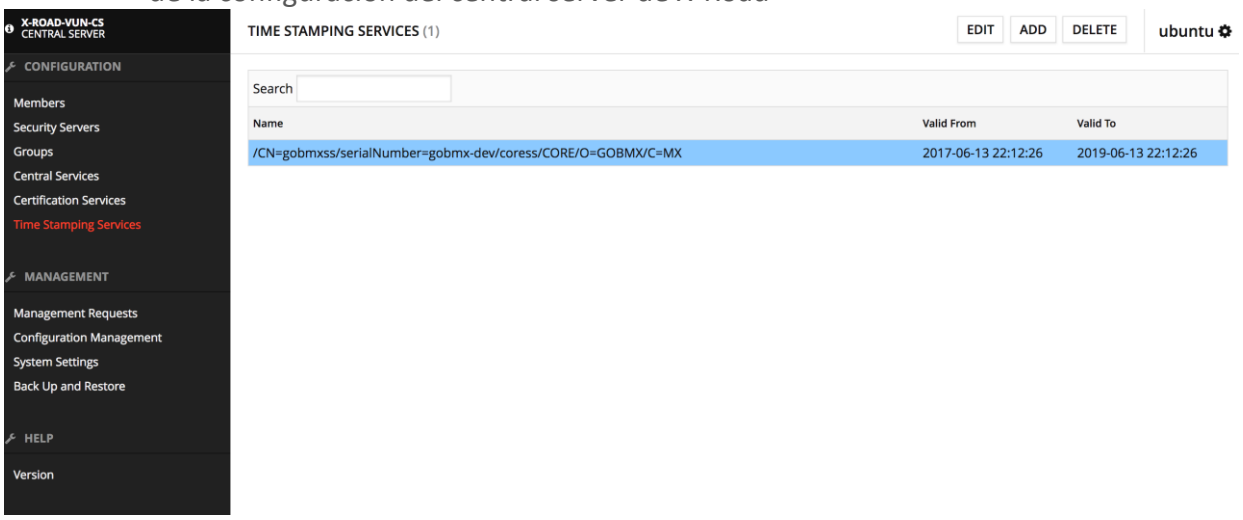
```
GOBMX::CATSA@>~/signserver-ce-4.0.0 $ bin/signserver reload all
SignServer reloaded successfully

Current configuration is now activated.

GOBMX::CATSA@>~/signserver-ce-4.0.0 $
```

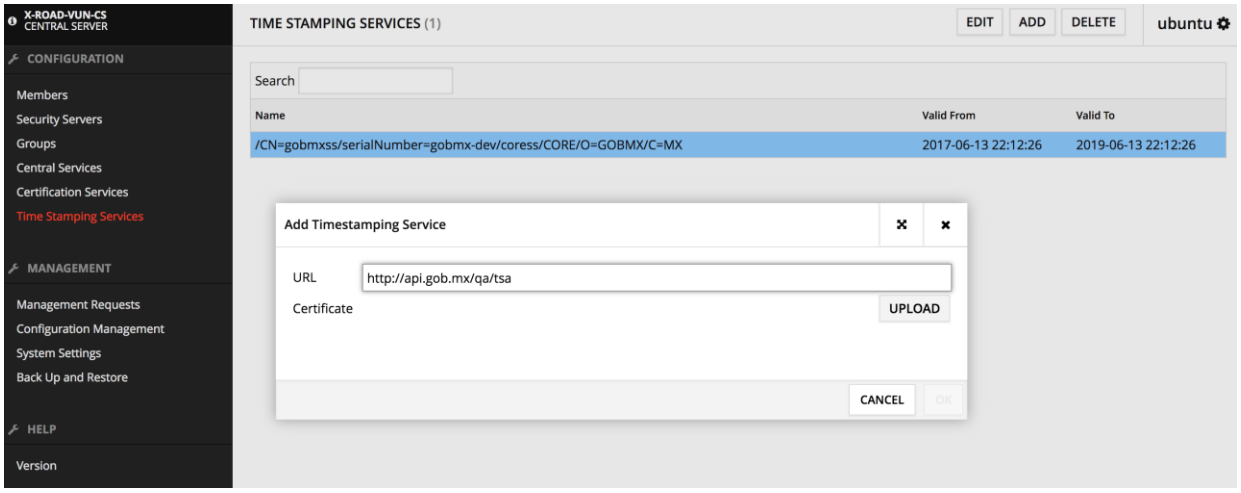
4. Se prueba de forma local que el timestamp signer esté funcionando correctamente con el comando signclient

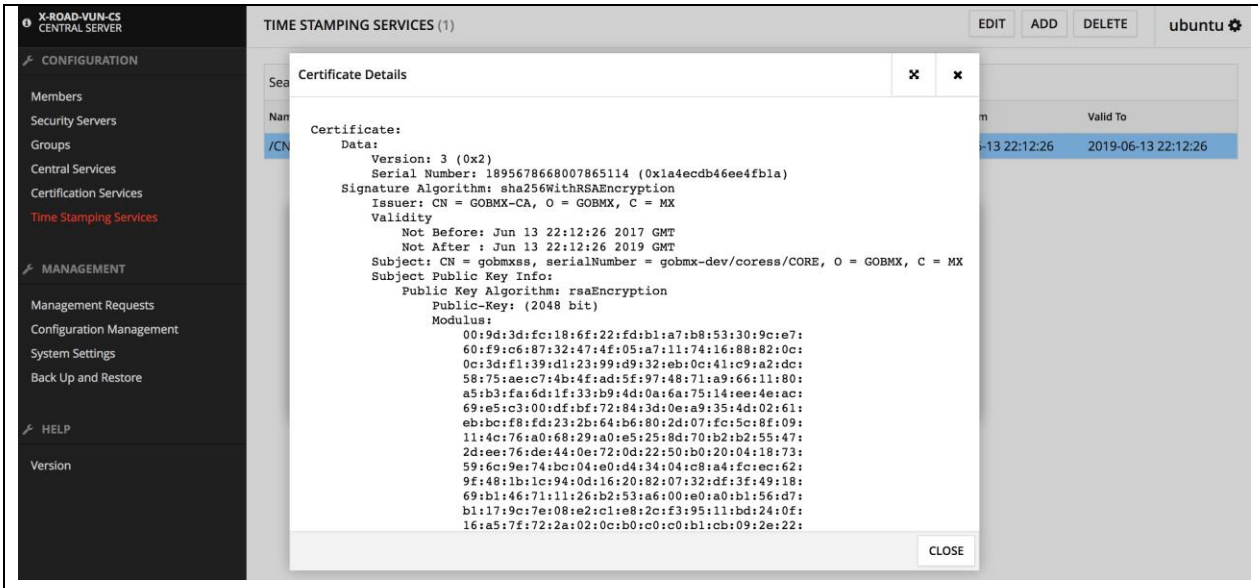
$ bin/signclient timestamp http://localhost:8080/signserver/process?workerName=TimeStampSigner

```
GOBMX::CATSA@>~/signserver-ce-4.0.0 $ bin/signclient timestamp http://localhost:8080/signserver/process?workerName=TimeStampSigner
2017-04-11 21:19:36,584 INFO  [TimeStampCommand] Got reply after 163 ms
2017-04-11 21:19:36,668 INFO  [TimeStampCommand] TimeStampRequest validated with status code: 0 (Operation Okay)
2017-04-11 21:19:37,726 INFO  [TimeStampCommand] Got reply after 57 ms
2017-04-11 21:19:37,728 INFO  [TimeStampCommand] TimeStampRequest validated with status code: 0 (Operation Okay)
2017-04-11 21:19:38,779 INFO  [TimeStampCommand] Got reply after 50 ms
2017-04-11 21:19:38,782 INFO  [TimeStampCommand] TimeStampRequest validated with status code: 0 (Operation Okay)
^CGOBMX::CATSA@>~/signserver-ce-4.0.0 $
```

5. Por último se procede a cargar el certificado pem creado en la sección anterior dentro de la configuración del central server de X-Road







Ligas relacionadas:

https://github.com/GOBMX/xroad-source
https://www.signserver.org