

Evidencias Interoperabilidad XRoad Julio 2017

gob.mx

A large, dark gray triangle that originates from the left edge of the page and extends diagonally towards the bottom right corner, covering the lower half of the page.

Índice

1	Objetivo.	3
2	Evidencias.	3

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 cryotoken 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 verificando 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
```

```

~ — ssh devops@10.20.3
GOBMX::CATSA@~/signserver-ce-4.0.0 $ cd bi
-bash: cd: bi: No such file or directory
GOBMX::CATSA@~/signserver-ce-4.0.0 $ cd bin
GOBMX::CATSA@~/signserver-ce-4.0.0/bin $ bin/signserver setproperties doc/sample-configs/keystore-crypto.properties^C
GOBMX::CATSA@~/signserver-ce-4.0.0/bin $ cd ..
GOBMX::CATSA@~/signserver-ce-4.0.0 $ bin/signserver setproperties doc/sample-configs/keystore-crypto.properties
Configuring properties as defined in the file : doc/sample-configs/keystore-crypto.properties
Setting the property CRYPTOTOKEN_IMPLEMENTATION_CLASS to org.signserver.server.cryptotokens.KeystoreCryptoToken for worker 1
Setting the property KEYSTORETYPE to PKCS12 for worker 1
Setting the property IMPLEMENTATION_CLASS to org.signserver.server.signers.CryptoWorker for worker 1
Setting the property NAME to CryptoTokenP12 for worker 1
Setting the property KEYSTOREPATH to /opt/signserver/res/test/dss10/dss10_keystore.p12 for worker 1
Setting the property TYPE to CRYPTO_WORKER for worker 1

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

```

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

- 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

- 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/>



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**
- Public Web**
- Documentation**
- Logout**

<input type="checkbox"/>	gobmxauthqa	GOBMX-CA	gobmxqa		GOBMX	Generated	Edit_End_Entity View_Certificates View_History
<input type="checkbox"/>	gobmxauthqaamazon	GOBMX-CA	gobmxqa		GOBMX	Generated	View_End_Entity Edit_End_Entity View_Certificates View_History
<input type="checkbox"/>	gobmxocsp	GOBMX-CA	gobmxss			New	View_End_Entity Edit_End_Entity View_Certificates View_History
<input type="checkbox"/>	gobmxsign	GOBMX-CA	gobmxss		GOBMX	New	View_End_Entity Edit_End_Entity View_Certificates View_History
<input type="checkbox"/>	gobmxsignqa	GOBMX-CA	gobmxqa		GOBMX	Generated	View_End_Entity Edit_End_Entity View_Certificates View_History
<input type="checkbox"/>	gobmxsignqaamazon	GOBMX-CA	gobmxqa		GOBMX	Generated	View_End_Entity Edit_End_Entity View_Certificates View_History
<input type="checkbox"/>	gobmxtsa	GOBMX-CA	gobmxss		GOBMX	Generated	View_End_Entity Edit_End_Entity View_Certificates View_History
<input type="checkbox"/>	superadmin	ManagementCA	SuperAdmin			Generated	View_End_Entity Edit_End_Entity View_Certificates View_History
<input type="checkbox"/>	tomcat	ManagementCA	localhost		EJBCA Sample	Generated	View_End_Entity Edit_End_Entity View_Certificates View_History

Select All | Unselect All | Invert Selection

Edit End Entity

End Entity Profile	GOBMX-TSA-EP	Required
Status	Generated <input type="button" value="Save"/>	
Username	gobmxtsa	<input checked="" type="checkbox"/>
Password (or Enrollment Code)	<input type="password"/>	<input checked="" type="checkbox"/>
Confirm Password	<input type="password"/>	
Maximum number of failed login attempts	<input type="radio"/> <input type="text"/> <input checked="" type="radio"/> Unlimited	
Remaining login attempts	<input type="text"/> <input type="checkbox"/> Reset login attempts	
E-mail address	<input type="text"/> @ <input type="text"/>	<input type="checkbox"/>
Subject DN		
CN, Common name	gobmxss	<input checked="" type="checkbox"/>
C, Country (ISO 3166)	MX	<input type="checkbox"/>
O, Organization	GOBMX	<input type="checkbox"/>
serialNumber, Serial number (in DN)	gobmx-dev/coress/CORE	<input type="checkbox"/>
Main certificate data		
Certificate Profile	GOBMX-TSA <input type="button" value="Save"/> <input type="button" value="Close"/>	<input checked="" type="checkbox"/>
CA	GOBMX-CA	<input checked="" type="checkbox"/>
Token	User Generated	<input checked="" type="checkbox"/>

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)

Edit

Certificate Profile: GOBMX-TSA

Back to Certificate Profiles	
Certificate Profile Id	169504415
Type	End Entity <input checked="" type="checkbox"/> Sub CA <input type="checkbox"/> Root CA <input type="checkbox"/>
Available bit lengths	<input type="checkbox"/> 0 bits <input type="checkbox"/> 192 bits <input type="checkbox"/> 239 bits <input type="checkbox"/> 256 bits <input type="checkbox"/> 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 [?]	<input type="checkbox"/> Allow
Allow extension override [?]	<input type="checkbox"/> Allow
Allow certificate serial number override [?]	<input type="checkbox"/> Allow No unique index for (issuerDN,serialNumber) on database table 'CertificateData'. "Allow certificate serialnumber override" not allowed.
Allow subject DN override [?]	<input type="checkbox"/> Allow
Allow Key Usage Override	<input type="checkbox"/> Allow
Allow back dated revocation [?]	<input type="checkbox"/> Allow

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

Public Web
Documentation
Logout

X.509v3 extensions	
Basic Constraints	<input checked="" type="checkbox"/> Use... <input checked="" type="checkbox"/> Critical
Authority Key ID	<input checked="" type="checkbox"/> Use
Subject Key ID	<input checked="" type="checkbox"/> Use
Key Usage	<input checked="" type="checkbox"/> Use... <input checked="" type="checkbox"/> Critical
Key Usage:	
<input checked="" type="checkbox"/> Digital Signature	<input type="checkbox"/> Data encipherment <input type="checkbox"/> CRL sign
<input checked="" type="checkbox"/> Non-repudiation	<input type="checkbox"/> Key agreement <input type="checkbox"/> Encipher only
<input checked="" type="checkbox"/> Key encipherment	<input type="checkbox"/> Key certificate sign <input type="checkbox"/> Decipher only
Extended Key Usage [?]	<input checked="" type="checkbox"/> Use... <input checked="" type="checkbox"/> Critical
Any Extended Key Usage	
Server Authentication	
Client Authentication	
Code Signing	
Email Protection	
1.3.6.1.5.5.7.3.5	
1.3.6.1.5.5.7.3.6	
1.3.6.1.5.5.7.3.7	
Time Stamping	
OCSP Signer	
Subject Alternative Name	<input checked="" type="checkbox"/> Use... <input type="checkbox"/> Critical
Issuer Alternative Name [?]	<input checked="" type="checkbox"/> Use... <input type="checkbox"/> Critical
Subject Directory Attributes	<input type="checkbox"/> Use
Name Constraints [?]	<input type="checkbox"/> Use... <input type="checkbox"/> Critical
Certificate Revocation List Distribution	

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

<http://10.20.37.221:8080/ejbca/>

Enroll

Create Browser Certificate
Create Certificate from CSR
Create Keystore
Create CV certificate

Register

Request Registration

Retrieve

Fetch CA Certificates
Fetch CA CRLs
List User's Certificates
Fetch User's Latest Certificate

Inspect

Inspect certificate/CSR
Check Certificate Status

Miscellaneous

As of 2017-03-14

Welcome to the public EJBCA pages

Enroll

- Create Browser Certificate - Install a certificate in your web browser. This certificate may be exportable depending on browser and browser settings.
- Create Certificate from CSR - Send a PKCS#10 certificate request generated by your server, and receive a certificate that can be installed on the server. Consult your server documentation.
- Create Keystore - Create a server generated keystore in PEM, PKCS#12 or JKS format and save to your disc. This keystore can be installed in a server, browser or in other applications.
- Create CV Certificate - Used for EU EAC ePassport PKI. Send a CVC certificate request generated by an Inspection System, and receive a CV certificate. Note: this can not be used for regular certificates, CV certificates are completely different.

Retrieve

- Fetch CA Certificates - Browse and download CA certificates.
- Fetch CA CRLs - Download Certificate Revocation Lists.
- Fetch User's Latest Certificate - Download the last issued certificate for a user for whom you know the certificate Distinguished Name.

Inspect

- Inspect certificate/CSR - Inspect a dump of a CSR or a certificate. This gives an output of a CVC or ASN.1 dump, suitable for technical inspection and debugging.

- 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

PKI BY PRIMEKEY

Enroll

- Create Browser Certificate
- Create Certificate from CSR
- Create Keystore
- Create CV certificate

Register

- Request Registration

Retrieve

- Fetch CA Certificates
- Fetch CA CRLs
- List User's Certificates
- Fetch User's Latest Certificate

Inspect

- Inspect certificate/CSR
- Check Certificate Status

Miscellaneous

- Administration
- Documentation

Certificate enrollment from a CSR

Please give your username and enrollment code, select a PEM- or DER-formatted certification request file (CSR) for upload, or paste a PEM-formatted request into the field below and click OK to fetch your certificate.

A PEM-formatted request is a BASE64 encoded certificate request starting with
-----BEGIN CERTIFICATE REQUEST-----
and ending with
-----END CERTIFICATE REQUEST-----

Enroll

Username

Enrollment code

Request file tscsr.req
or pasted request

Result type

- 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>

Leví Durán Torres, Arquitecto de Sistemas B, 30 de Junio de 2017 al 14 de Julio de 2017,03

Página 2 de 5

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)
ACGOBMX::CATSA@~/signserver-ce-4.0.0 $
```

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

X-ROAD-VUN-CS
CENTRAL SERVER

CONFIGURATION
Members
Security Servers
Groups
Central Services
Certification Services
Time Stamping Services
MANAGEMENT
Management Requests
Configuration Management
System Settings
Back Up and Restore
HELP
Version

TIME STAMPING SERVICES (1)

EDIT ADD DELETE ubuntu

Search

Name	Valid From	Valid To
/CN=gobmxss/serialNumber=gobmx-dev/coress/CORE/O=GOBMX/C=MX	2017-06-13 22:12:26	2019-06-13 22:12:26

Add Timestamping Service

URL

http://api.gob.mx/qa/tsa

Certificate

UPLOAD

CANCEL

OK

0

X-ROAD-VUN-CS

CENTRAL SERVER

CONFIGURATION

Members

Security Servers

Groups

Central Services

Certification Services

Time Stamping Services

MANAGEMENT

Management Requests

Configuration Management

System Settings

Back Up and Restore

HELP

Version

TIME STAMPING SERVICES (1)

EDIT ADD DELETE ubuntu

Search

Name

/CN

Valid To

-13 22:12:26 2019-06-13 22:12:26

Certificate Details

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 1895678668007865114 (0x1a4ecdb46ee4fb1a)

Signature Algorithm: sha256WithRSAEncryption

Issuer: CN = GOBMX-CA, O = GOBMX, C = MX

Validity

Not Before: Jun 13 22:12:26 2017 GMT

Not After : Jun 13 22:12:26 2019 GMT

Subject: CN = gobmxss, serialNumber = gobmx-dev/coreess/CORE, O = GOBMX, C = MX

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (2048 bit)

Modulus:

00:9d:3d:fc:18:6f:22:fd:b1:a7:b8:53:30:9c:e7:

60:f9:c6:87:32:47:4f:05:a7:11:74:16:88:82:0c:

0c:3d:f1:39:d1:23:99:d9:32:eb:0c:41:c9:a2:dc:

58:75:ae:c7:4b:4f:ad:5f:97:48:71:a9:66:11:80:

a5:b3:fa:6d:1f:33:b9:4d:0a:6a:75:14:ee:4e:ac:

69:e5:c3:00:df:bf:72:84:3d:0e:a9:35:4d:02:61:

eb:bc:f8:fd:23:2b:64:b6:80:2d:07:fc:5c:8f:09:

11:4c:76:a0:68:29:a0:e5:25:8d:70:b2:b2:55:47:

2d:ee:76:de:44:0e:72:0d:22:50:b0:20:04:18:73:

59:6c:9e:74:bc:04:e0:d4:34:04:c8:a4:fc:ec:62:

9f:48:1b:1c:94:0d:16:20:82:07:32:df:3f:49:18:

69:b1:46:71:11:26:b2:53:a6:00:e0:a0:b1:56:d7:

b1:17:9c:7e:08:e2:c1:e8:2c:f3:95:11:bd:24:0f:

16:a5:7f:72:2a:02:0c:b0:c0:c0:b1:cb:09:2e:22:

CLOSE

Ligas relacionadas:

<https://github.com/GOBMX/xroad-source>

<https://www.signserver.org>