A3SA Application Note – Preparation for NextGen TV Signal Signing

Version 0.9.2

12/18/2024

Notice

Copyright

Version History

Version #	Implemented By	Revision Date	Notes
v0.9	A3SA Technical Committee Eonti, Heartland Video Sys	02/25/2022	PRELIMINARY VERSION SUITABLE FOR BETA TESTING
v0.9.1	A3SA Technical Committee, Heartland Video Systems	04/04/2022	• Some signing engines automatically perform the bind step obviating section 4.2.3 • Relaxed some testing requirements • Updated test results recording questionnaire
v0.9.2	A3SA Technical Committee	12/18/2024	• Minor editorial updates

TABLE OF CONTENTS

TABLE OF FIGURES

1. Introduction

1.1. Scope

Cryptographically signing signaling tables in a NextGen TV broadcast emission is both mandatory and essential for compliance with the ATSC 3 security system. This document provides informative descriptions of how to perform the necessary steps to prepare equipment and software, conduct compliance verification, and enable ATSC 3 signal signing. It is intended to supplement rather than replace vendor-specific documents that may cover some of the same tools and topics. If there are inconsistencies between vendor documents and this application note the vendor document should be determinative.

1.2. Assumptions

This document covers only Broadcasters that are using the ROUTE protocol [2].

1.3. Overview

The following high-level steps should be followed to prepare for NextGen TV Signal Signing.

- Register (Section 2)
- Generate and submit the necessary data (Section 4)
- Install certificates (Section 4)
- Verify the installation (Section 6)
- Self-certify your results (example in Appendix A, contact A3SA for [1])

1.4. References

This document may make reference to other documents identified here.

[1] A3SA: “Broadcaster Readiness Questionnaire for NextGen TV Signal Signing,” 2/14/2022.

[2] ATSC: ATSC Standard: “Signaling, Delivery, Synchronization, and Error Protection,” Doc. A/331:2023-03, Advanced Television System Committee, Washington, DC, 28 March 2023.

[3] ATSC: ATSC Standard: “ATSC 3.0 Security and Service Protection”, Doc. A/360:2023-03, Advanced Television System Committee, Washington, D.C., 28 March 2023.

[4] ISO/IEC: “Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 1: MPEG media transport (MMT),” Doc. ISO/IEC 23008-1:2017(E), International Organization for Standardization/ International Electrotechnical Commission, Geneva Switzerland.

1.5. Definitions and Terms

"**ATSC 3**" means ATSC 3.0 and subsequent versions of the broadcasting suite of standards, as referenced by *ATSC A/300 "ATSC 3.0 System"*.

"**Certificate"** means digital file used for identity and authorization. The file identifies the broadcaster, contains the broadcaster's Public Key, identifies the Validity Period for the certificate, a unique serial number and any other identifying information the ecosystem deems relevant. Commonly Certificates are referred to as X.509 Certificates after the standard set to define the format of Certificates.

"Certification Authority" or **"CA"** means the hardware, software, processes, facilities and personnel used to issue, manage, revoke and renew Certificates.

"**CertificationData LLS Table"** or **"CDT**" means a specific signaling message that shall be cryptographically signed for compliance with ATSC 3; see [3].

"**Certificate Signing Request"** or **"CSR"** means a file that will contain the Broadcaster's Public Key (tied to the Broadcaster's Private Key) and will also include other information that should be included within the end-entity Certificate (e.g., the broadcaster name, country, etc.). You will eventually be sending this CSR file to Eonti (the PKI Registration Authority and Management Authority) to obtain the unique broadcaster end-entity Certificate.

"**Digital Certificate Subscriber Agreement**" or "**DCSA**" means the Subscriber Agreement that must be in place between the Broadcaster and Eonti prior to completing the signal signing certificate issuance process.

"Integrated Receiver Decoder" or **"IRD"** means a signal reception system that receives and converts modulated signals back into their original format suitable for presentation to an end-user device or display. IRDs typically contain a built-in decoder for unscrambling TV programming channels.

"Intermediate CA" or **"ICA"** means a Certificate chained to the Root CA Certificate that can be used to issue Signing Certificates.

"**MPEG Media Transport Protocol"** or **"MMTP**" means the protocol used to deliver broadcast services defined in [4].

"**Online Certificate Status Protocol**" or **"OCSP"** means the Internet protocol used for obtaining the revocation status of an X.509 digital certificate.

"**Private Key**" means the key of a key pair that is kept secret by the holder of the key pair, and that is used to create digital signatures and/or to decrypt electronic records or files that were encrypted with the corresponding Public Key.

"**Public Key**" means the key of a key pair that may be publicly disclosed by the holder of the corresponding Private Key and that is used by the Relying Party to verify digital signatures created with the holder's corresponding Private Key and/or to encrypt messages so that they can be decrypted only with the holder's corresponding Private Key.

“**Public Key Infrastructure” or “PKI**” means the architecture, organization, techniques, practices, and procedures that collectively support the implementation and operation of a Digital Certificate-based Public Key cryptographic system.

“**Real-Time Object Delivery over Unidirectional Transport” or “ROUTE**” means the protocol used to deliver broadcast services defined in [2].

“Root CA” means the highest CA in the PKI and is the trust point for all Certificates that are issued by the ATSC 3 PKI.

“**SignedMultiTable” or “SMT**” means a specific signaling message that shall be cryptographically signed for compliance with ATSC 3; see [2].

“**Signing Engine”** means the product, system, or platform that cryptographically signs the signaling message tables in an ATSC 3 emission.

2. Registration for A3SA Signal Signing

The process of obtaining signal signing certificates begins with the A3SA PKI Registration and Management Authority, Eonti (www.eonti.com).

The Broadcaster may register and sign their signals without first licensing with A3SA. There may be certain advantages to a broadcaster who also registers with A3SA, but it is not necessary for signal signing. Questions about A3SA licensing can be sent to admin@A3SA.com.

2.1. The Certificate Issuance Process

Contact Eonti (a3sapki@eonti.com) to request a broadcaster certificate account. Eonti will send the Broadcaster the Digital Certificate Subscriber Agreement (DCSA). They will also send:

- Certificate Naming Application(s),
- Customer Contact Information Form, and
- Fee Schedule.

The Broadcaster should execute the DCSA and return it to Eonti along with:

- Completed Certificate Naming Application(s),
- Completed Customer Contact Information,
- Certificate Signing Request (CSR) files for each certificate requested (See Section 3 for details), and
- Purchase order.

Eonti will then conduct Broadcaster Identity Verification, including

- Verifying the legal company name and address,
- Verifying the BSID,
- Verifying Facility ID,
- Checking the A3SA license status, and
- Emailing the contacts listed on the customer profile form to verify their information.

When Identity Verification is complete, Eonti invoices the broadcaster for the requested certificates.

Once payment is received as well as the necessary CSRs for the requested certificates, Eonti activates the account and generates the certificates ordered. Certificates are then securely sent to the Broadcaster.

See (Appendix B) for a flow chart overview of this process.

3. Signing Certificate Requirements

For each ATSC 3 transmitter the requirement is to buy two certificates: one for the CertificationData LLS Table (CDT) and one for the SignedMultiTable (SMT) signaling message tables. During the ATSC 3 rollout transition period, only stations that emit an ATSC 3 signal require certificates. Lighthouse (AKA Host stations) sign all broadcast services on behalf of all stations they transmit. Guest stations do not need individual signal signing certificates.

Additionally, a Lighthouse or Host transmitter is required to purchase a certificate if any Broadcast Applications (either Host's or Guest's) are being sent. Only one of these certificates is required for the transmitter to cosign any apps being distributed over the air by the station.

Lastly, any author of a Broadcast App is required to obtain a certificate to sign their own application (this is the app that will be co-signed by the Lighthouse Broadcaster Application certificate). Once an app is signed, it can be used as many times on as many different transmitters as needed (bearing in mind each transmitter needs to be certified to transmit app data).

3.1. File Naming Conventions

Since there will be numerous data files generated, exchanged, and stored as part of this configuration process, it is important to establish a naming convention for effective identification of the various data elements used in the step-by-step process. The file naming convention used in this document has the following parts:

<Station-ID><type><serial no>

Where:

- <Station-ID> should be the station call letters. In the case of Application signing, this should be company abbreviation.
- <type> should be
    - For signaling signing either CDT or SMT, or
    - Omitted for application signing
- <serial-no> is a number the broadcaster can assign if there are multiple Certificates to help distinguish one Certificate from an earlier version.

For example, the following name could signify the CSR for the certification data table for a fictitious station with call sign WZZZ, serial number 1:

```
WZZZcdt01.csr
```

In addition, files exchanged with Eonti may include additional type information, depending on the purpose of the certificate.

<Station-ID><purpose><type><serial no>

Where:

- <Station-ID> should be the station call letters. In the case of Application signing, this should be company abbreviation.

- <purpose> should be
  - Signaling Signing
  - Application Distributor
  - Application Author
- <type> should be
  - For signaling signing either CDT or SMT, or
  - Omitted for application signing
- <serial-no> is a number the broadcaster can assign if there are multiple Certificates to help distinguish one Certificate from an earlier version.

When applicable, file naming should incorporate this purpose. For example, the following name could signify a privacy enhanced mail (armored format) for the signaling signing certification data table for a fictitious station with call sign WZZZ:

```
WZZZ-signaling-signing-cdt01.pem
```

Here is an example of a privacy enhanced mail for the application distributor for a fictitious broadcast company ACME, serial number 1:

```
ACME-application-distributor01.pem
```

3.2. Storing Private Keys, Certificates, and CSRs

3.2.1. USB Tokens

After a Private Key, a Certificate, or a CSR is created, it needs to be protected. There are a few methods to do this depending on cost and company policy. One way to store this important data is to use a hardened USB token. These tokens generally require a physical pin to access the token. It is recommended that the Private Key be encrypted once it is loaded onto the token. This double layer acts as extra security and adds a level of assurance to prevent, for example the Private Key from being taken or used without authorization. Storing keys in a file stored on either fixed or removable media, even where that file has been encrypted, is strongly discouraged unless the media device itself is safely stored in a location that requires multi-person access.

3.2.2. Key Backup and Redundancy

To ensure that Private Keys and Certificates can always be used and accessed, backups should be created. At a minimum, two (2) backups should be created: one onsite and the other offsite. If you lose your private key, you will be unable to install your certificates, you will need to generate a new key pair and re-issue the certificate.

One method to create the offsite backup is using a local bank to store a hardened USB token in a safety deposit box. If feasible, a geographically diverse location should be used. If your company has multiple locations, then the Private Key can be stored at a second location. It should be noted that at the second facility the Private Key should be secured, and two-party control should be maintained to prevent unauthorized access.

3.2.3. Multi-Party Control

At all times, a Private Key should only be accessible with multi-party control. Multi-party control ensures that one malicious insider or inexperienced technician is not able to act alone. It requires at least two (2) people to allow an action to take place. This prevents unauthorized use and also adds assurance that the key will not be unintentionally deleted or corrupted. Separation of duties in multi-party control is very important. An example of multi-party control with separation of duties would be if one employee has access to the USB token, another employee has access to the password for the token and a third person has access to the laptop where the key pair is created.

4. Certificate Generation

4.1. Third-party Tools

Commercial companies delivering signing engines have features to generate the necessary CSR files. The process differs depending on the vendor and tools employed. See Appendix C for examples using third-party vendors' platforms.

Many steps in this document (CSR and certificate generation, certificate validation, etc.) depend on command-line tools from the OpenSSL project (www.openssl.org). These are included with all Linux distributions and can be readily installed on Windows and/or Mac computers. See the following site for a variety of options for OpenSSL distributions:

https://wiki.openssl.org/index.php/Binaries.

The following site offers a variety of OpenSSL installers for Windows computers:

https://slproweb.com/products/Win32OpenSSL.html

In addition, when transferring sensitive files such as private keys and certificates, be sure to follow your local IT department's policies regarding such transfers. We use WinSCP, a popular file transfer client, that supports secure copying and secure FTP:

https://winscp.net/eng/download.php

4.2. Step-by-step Commands for Certificate Generation

4.2.1. Generate the CSR

Once the OpenSSL tool is installed and configured on a host computer, enter at a command-line prompt the following command to generate the CSR for the CDT table (remember to use your own <Station-ID><type><serial no> fields):

```
openssl req -new -newkey rsa:3072 -noenc
  -keyout WZZZ-signaling-signing-cdt01.key
  -out WZZZ-signaling-signing-cdt01.csr
```

Note that you will be asked to enter the following fields of information to be associated with the CSR. Answer to the best of your knowledge; they do not have to be perfect but are used for informative purposes.

- Country Name (2 letter code)

- State or Province Name (full name)
- Locality Name (e.g., city)
- Organization Name (e.g., company)
- Organizational Unit Name (e.g., dept)
- Common Name (enter <Station-ID><type><serial no>)
- Email Address

Repeat for the SMT table:

```
openssl req -new -newkey rsa:3072 -noencdes
  -keyout WZZZ-signaling-signing-smt01.key
  -out WZZZ-signaling-signing-smt01.csr
```

Keep the `.key` files handy as they will be needed later in the process. At the end, store them securely along with the certificates and verification results as per Section 0.

4.2.2. Exchange Files with Eonti

The `.csr` files generated above should be transmitted to Eonti. Email is adequate for this transfer since no Private Keys are contained in the CSR. After Eonti uses the CSRs to create the Public Key Certificates, they will return three Certificates for each request. For example, for the CDT request:

- `ATSC3_Root-CA_root.crt`
- `A3SA Signing CA intermediate.crt`
- `A3SA Signaling Signing cdt01 primary.crt`

The primary certificate (aka the end entity certificate) requires binding to its private key as is described in the next section. The Root CA and ICA installation is done without the binding process.

4.2.3. Bind Broadcaster Private Keys to the Certificates

If you used a signing engine to generate the CSR, then that Signing Engine might automatically perform the binding process described in this section as the primary Certificates are imported into the system. For these implementations, this section should be skipped. If the Signing Engine that is going to be used does not perform the binding to the previously generated Private Key, the process described in this section must be followed for successful installation of the primary Certificate.

When you receive the primary certificates from Eonti, they do not contain the broadcaster Private Key. However, in order for the signal signing tools to construct signing tables properly, you need to bind the broadcaster Private Key with the Certificate.

At a command-line prompt enter the following command to insert the broadcaster Private Key into the CDT signing certificate.

```
openssl pkcs12 -export -out A3saWZZZcdt01.p12
     -inkey A3SA-signaling-signing-cdt01.key
     -in "A3SA Signaling Signing cdt01 primary.crt"
```


Repeat for the SMT certificate:

```
openssl pkcs12 -export -out A3saWZZZsmt01.p12
     -inkey A3SA-signaling-signing-smt01.key
     -in "A3SA Signaling Signing smt01 primary.crt"
```

5. Certificate Installation

A number of companies delivering tools and technology to the broadcast industry for NextGen TV have products that provide application and signal signing features suitable for A3SA compliance. The process to install the certificates on these Signing Engines differs depending on the vendor and tools employed by the broadcaster. Therefore, the process is described separately for each vendor in the following sections.

5.1. Certificate Installation Using Triveni

Here we describe the certificate installation process for the Triveni GuideBuilder XM (GBXM) Signing Engine. For more information see: http://www.trivenidigital.com/products/guidebuilder-xm-atsc3-metadata-system.php.

1. If you used the GBXM to generate the CSR, the GBXM automatically binds the primary Certificate to the Private Key during the Certificate import to the system, so the binding process described in Section 4.2.3 should not be followed.
2. Put all four certificates in a local folder on the GBXM. Use WinSCP or similar program allowing secure transfers. In our example this would be:
   - `ATSC3_Root-CA_root.crt`
   - `A3SA Signing CA intermediate.crt`
   - `A3SA Signaling Signing cdt01 primary.crt`
   - `A3SA Signaling Signing smt01 primary.crt`
3. Go to the Config App/Certificates section of GBXM. Click Add and select the four files, see below:

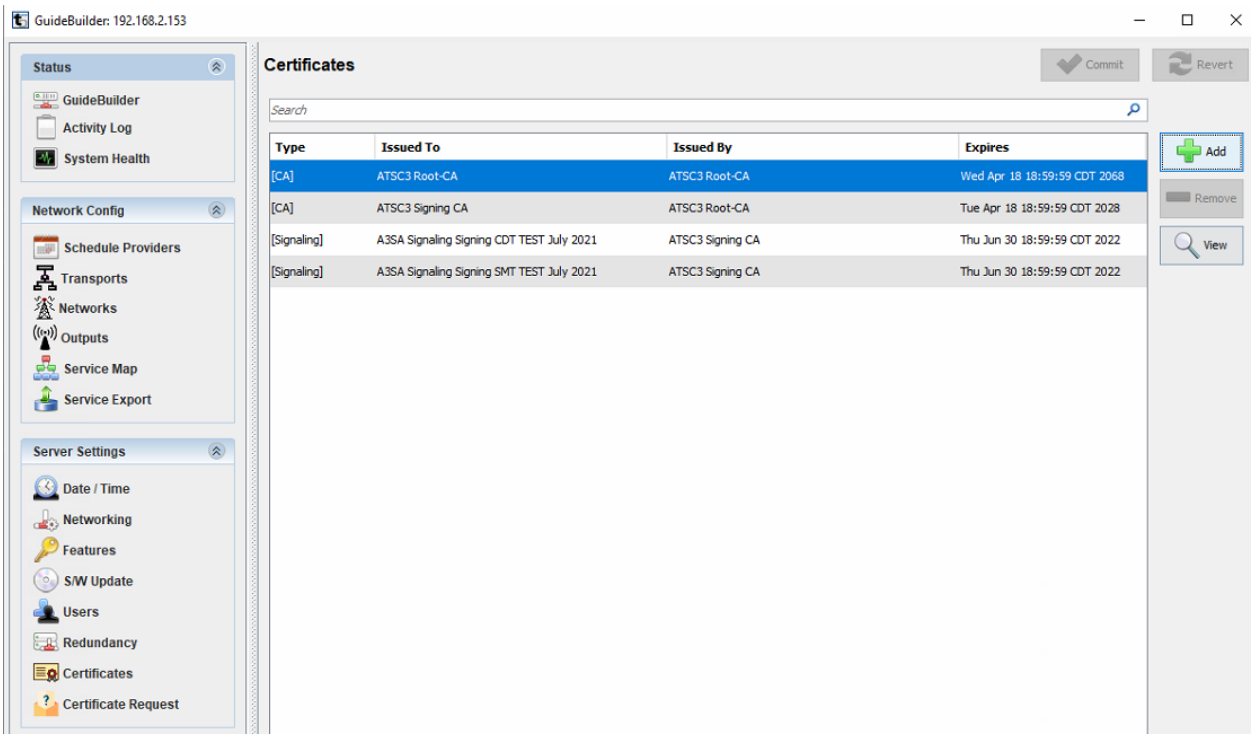


Figure 1. Add Certificates to GBXM

4. Click Commit after they are added.
5. Now click on the ConfigApp/Transports tab.

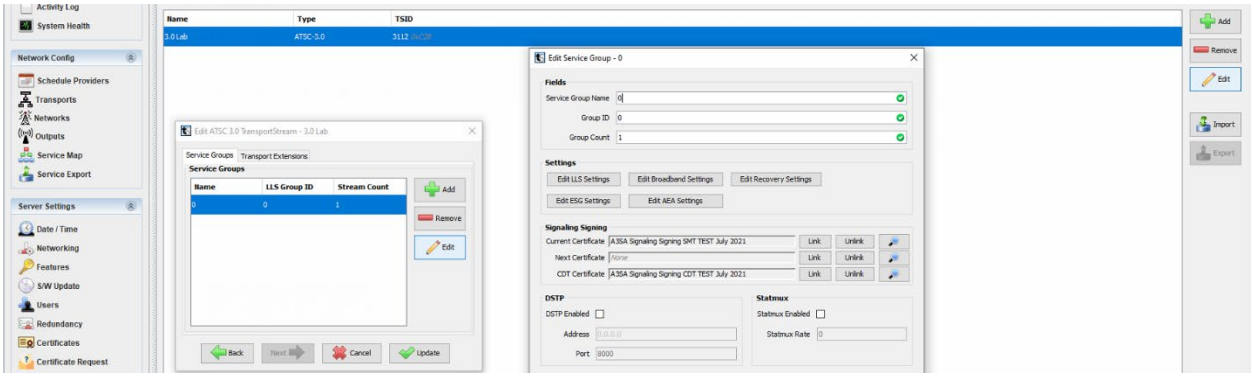


Figure 2. Link Certificates to Services on GBXM

6. Select your Transport and click Edit. From the pop-up window, select the actual Service Group and click edit again.
7. In the Signal Signing area, you will link the SMT using the Current Certificate link and the CDT using the CDT Certificate link. For initial setup, you will not need to link a certificate to Next Certificate. Click OK.
8. Be sure to click Update, then click Commit.
9. Note: The Root and Intermediate ICAs don't need to be assigned, they just need to be imported in the previous steps.

5.2. Certificate Installation Using DigiCAP

The following steps should be followed to install the signing certificates on the DigiCAP DigiCaster signing engine.

1. Go to the Configuration/Security menu to import the CRT files you generated in Section 4.2. You must import them in the following order:
   a. Root (e.g., `ATSC3_Root-CA_root.crt`)
   b. Signing Intermediate (e.g., `A3SA Signing CA intermediate.crt`)
   c. Primary for CDT (e.g., `A3saWZZZcdt01.p12`)
   d. Primary for SMT (e.g., `A3saWZZZsmt01.p12`)
2. Turn on Signing Security and choose SLT and STT in the Signing LLS List.

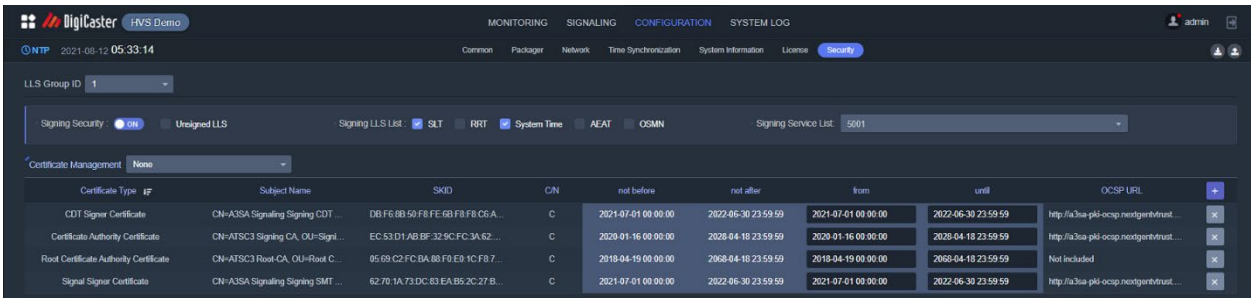


Figure 3. Enabling Signing on DigiCaster

3. Now go to Monitoring and then edit the services you want to have signed. Turn on the Signal Signing under the Edit Service section.

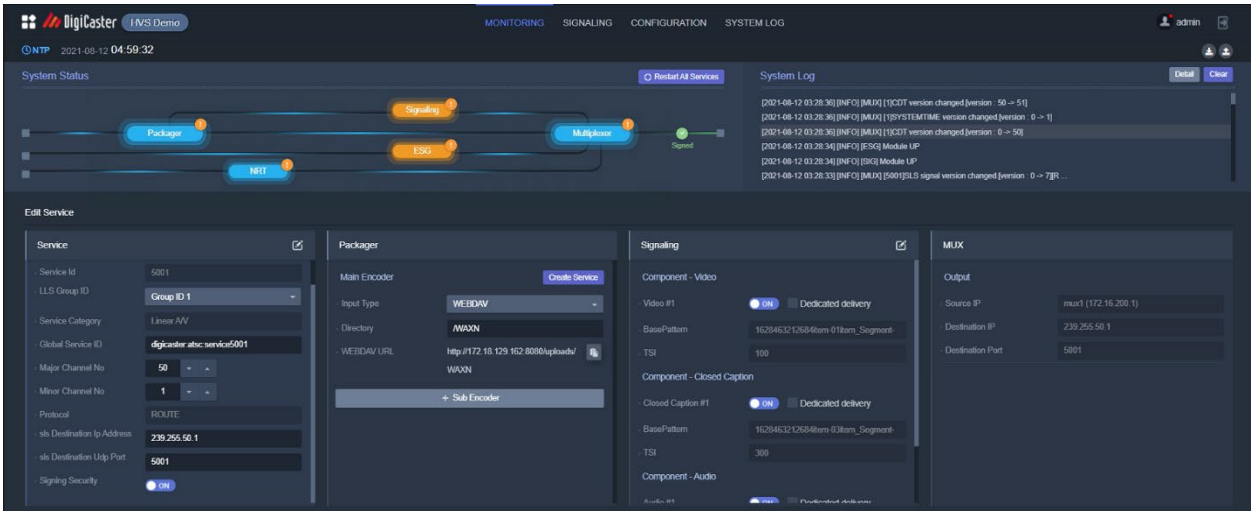


Figure 4. Link Certificates to Services on DigiCaster

5.3. Certificate Installation Using Enensys

The following steps should be followed to install the necessary certificates on the Enensys MediaCast signing engine.

1. Click on Certificate tab.
2. Click Import PKCS#12 and select the CDT with Chain .p12 file. The certificate should now be loaded. Do the same with the SMT .p12 file.

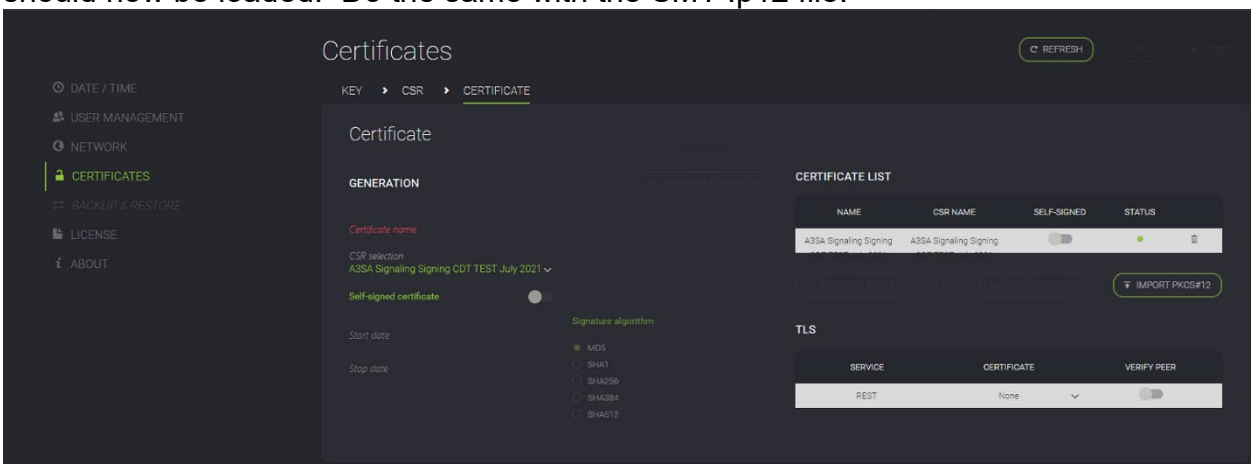


Figure 5. Add Certificates to MediaCast

3. Now go to Signaling and down to the Certificates area.
4. Click Add New + and choose the CDT certificate you loaded earlier. Click Use for CDT signing and hit apply. Do the same for the SMT.

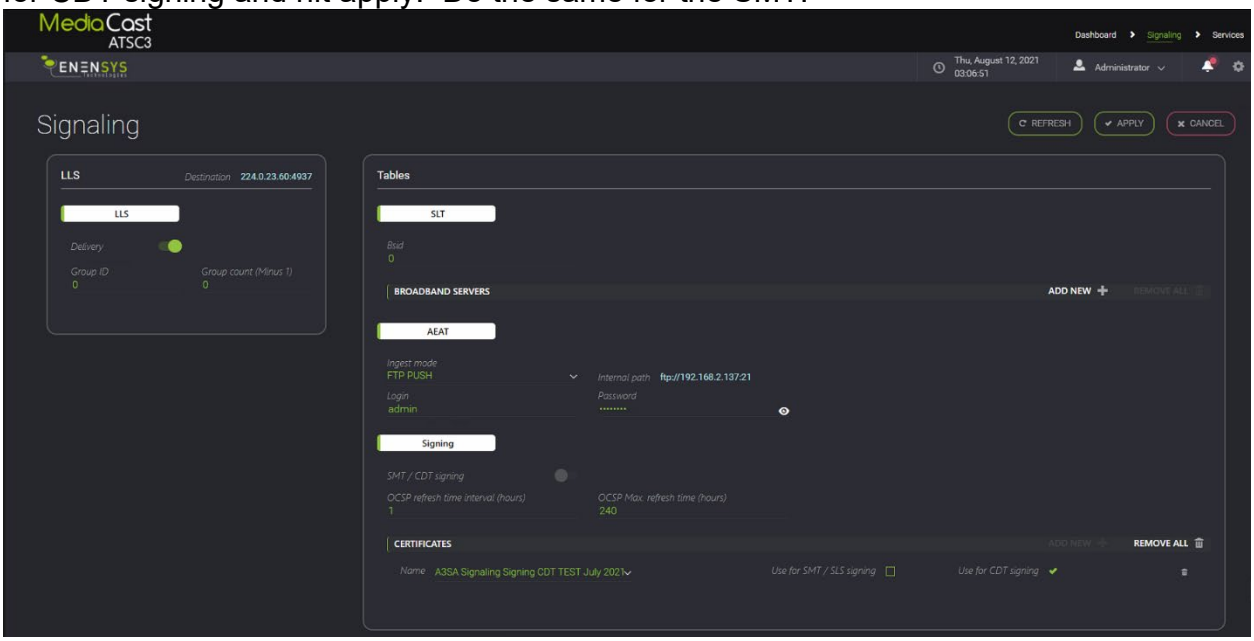


Figure 6. Assigning Certificates to Services with MediaCast

5. Make sure to turn on SMT/CDT signing by using the slide button.
6. Go to the Services tab and make sure to turn on the signing slide button for the actual services you want signed too.

6. Verification of Signal Signing Installation

The verification process consists of the following high-level steps, described in the next sections:

1. Verify that all installed Certificates are good using the integrated Signing Engine OCSP request/response, if available,
2. Verify your installation using an IRD and its built-in validity checking,
3. Perform manual certificate extraction and validation, and
4. Testing using ATSC 3 TVs for the consumer market.

You will need to record the results of your testing described in Sections 6.1, 6.2, and 6.3. You must record the results in the fillable form that is available in Appendix A [1], or upon request by emailing compliance@a3sa.com.

6.1. Verification of Certificates Using Signing Engine OCSP Features

Some Signing Engine implementations offer a user interface feature to verify that each installed Certificate is good by querying the OCSP database. If this is available, perform this step first. The instructions for performing this step will vary depending on which Signing Engine you have deployed. Refer to the vendor specific user documentation.

6.2. Verification Using IRD Validity Checking

A number of companies delivering tools and technology to the broadcast industry for NextGen TV provide IRDs with features suitable for verification of application and signal signing setup compliance with A3SA. For example, the following products can each be used in varying capacity to analyze and record ATSC 3 broadcast streams received from RF, Ethernet, and PCAP file inputs:

- Sencore ARD3100
  https://www.sencore.com/product/atsc-3-0-receiver-decoder-ard-3100-3400/

- DS Broadcast BGD4100
  http://dsbroadcast.com/ds/product/?A=2&uid=5

- Triveni StreamScope XM
  http://www.trivenidigital.com/products/streamscope-atsc3.0-broadcast-stream-analyzers.php

In each case, the exact procedure and IRD setup varies. For example, using the StreamScope XM, click the “System Tables” button in the left panel to check the certificate status for a specific service. Otherwise see the vendor and product-specific documentation to perform basic verification that the ATSC 3 stream with signal signing configured is operating without issue.

1. Before enabling service signal signing, confirm that both the CDT and SMT show a Signature status as "unsigned."
2. After enabling signal signing, confirm that both the CDT and SMT Signature status shows “Valid” and *not* “unsigned,” "revoked," or “invalid.”

6.3. Certificate Extraction and Validation

Certificate extraction and validation is a strongly recommended series of steps to complete self-certification for signal signing. Please follow these steps in order to verify that the necessary Certificates have been correctly generated and installed. The steps in this section will also be used in the event of an audit.

A number of companies delivering tools and technology to the broadcast industry for NextGen TV have IRDs that provide certificate extraction features suitable for verification of signing setup suitable for A3SA and ATSC 3 compliance. The process to extract and verify the necessary Certificate generation and installation differs depending on the vendor and tools employed by the broadcaster. Therefore, the process is described separately for each vendor in the following sections.

However, for each vendor, the process follows these high-level steps:

1. Extract the certificates from the signed signaling message tables emanating from the newly installed broadcaster Signing Engine,
2. Perform a basic decode and manual inspection of the information in the certificates,
3. Parse the certificates using OpenSSL, and
4. Confirm the validity status of the certificates using the Online Certificate Status Protocol (OCSP).

Please note that this last step requires the computer being used to perform the validation to be connected to the Internet.

6.3.1. Sencore ARD3100

TBD

6.3.2. DS Broadcast BGD4100

TBD

6.3.3. Triveni StreamScope XM

6.3.3.1. Certificate Extraction Using StreamScope

1. Provide as input to StreamScope XM a signed signal. This can be done directly via RF, via ethernet/IP, or by providing a PCAP recording of the signal.

2. Click System Tables in the left side panel.
3. Then click on the CDT.
4. Click XML in the right pane, and then click Download. Move the file to your working computer where you have access to text editors, OpenSSL, and the Internet.

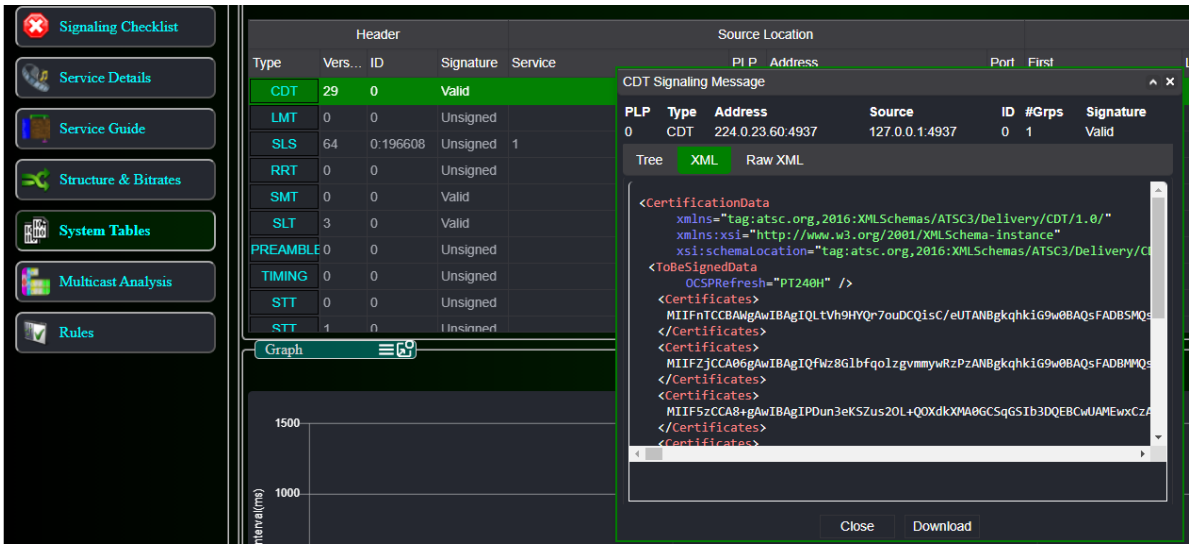


Figure 7. StreamScope XM System Tables for Certificate Export

5. Open the downloaded XML file. Many applications can open and read XML; for example, you can use a text editor or your default browser. The example below uses Notepad++ (https://notepad-plus-plus.org/) on Windows OS.

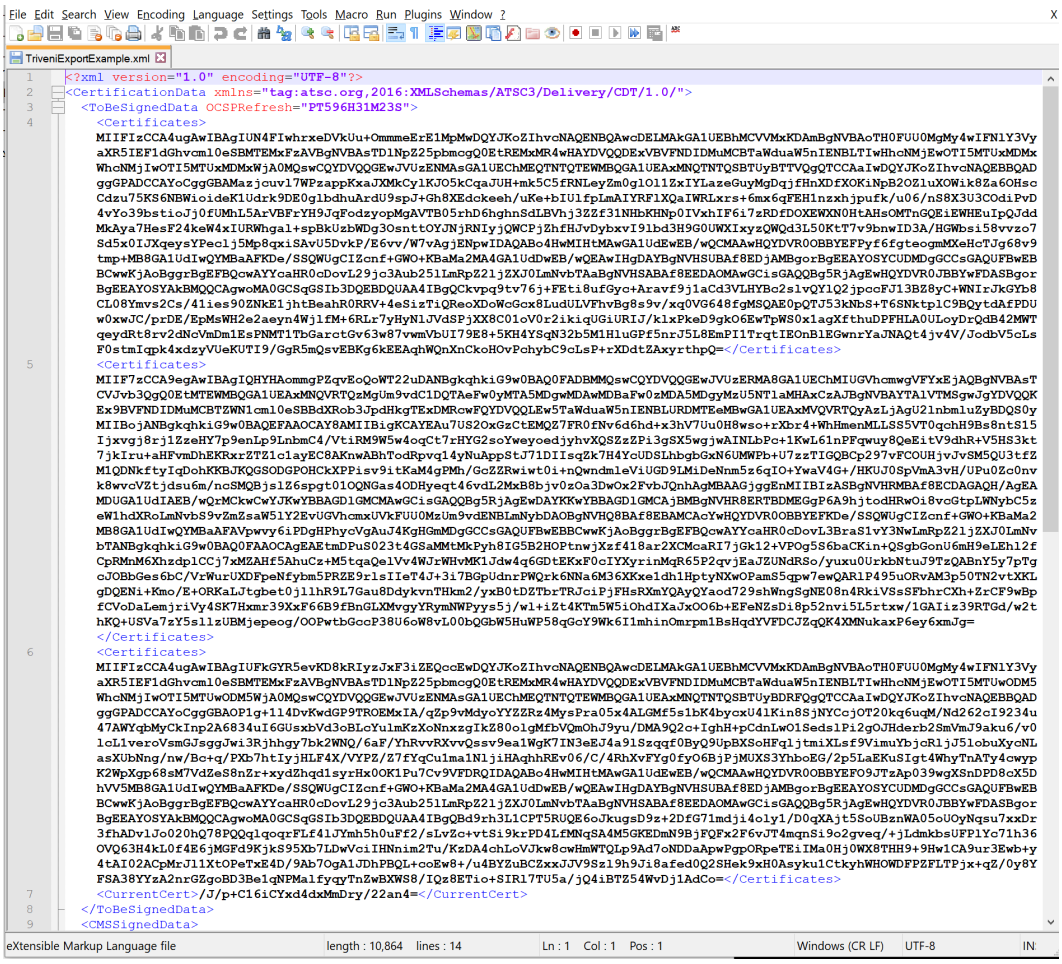


Figure 8. Text Editor View of Triveni GBXM Certificates in XML format

6. There should be three lines that start with the XML tag <Certificates> and end with </Certificates>. These contain the bodies of the three Certificates of interest: the primary CDT, the primary SMT, and the ICA.
    a. The exact order of Certificates in the XML file is vendor workflow specific.
    b. At this point, which XML string comprises which Certificate is unknown.

6.3.3.2. Certificate Inspection Using ASN.1 JavaScript Decoder

Repeat the following numbered steps for each of the three lines containing:

```
<Certificates> <cert body data> </Certificates>
```

1. Copy to your clipboard all the text between the <Certificates> tags. Make sure to copy any "=" characters at the end of the string too.
2. Bring up the ASN.1 JavaScript decoder tool at: https://lapo.it/asn1js/. This tool can either be used via the Internet website, or it can be installed locally.
3. Paste the data from your clipboard into the blank field and then click “decode”.
4. Confirm that the string parses without error messages.
5. You may be able to find and recognize some PrintableStrings that should align with the data you entered during certificate generation. In particular, find the

"OBJECT IDENTIFIER 2.5.4.3" line and under it the PrintableString. This is the Subject Common Name; check it for consistency with your station and signing table. For example, this screen shot shows the decoding output of the XML example in the last section:

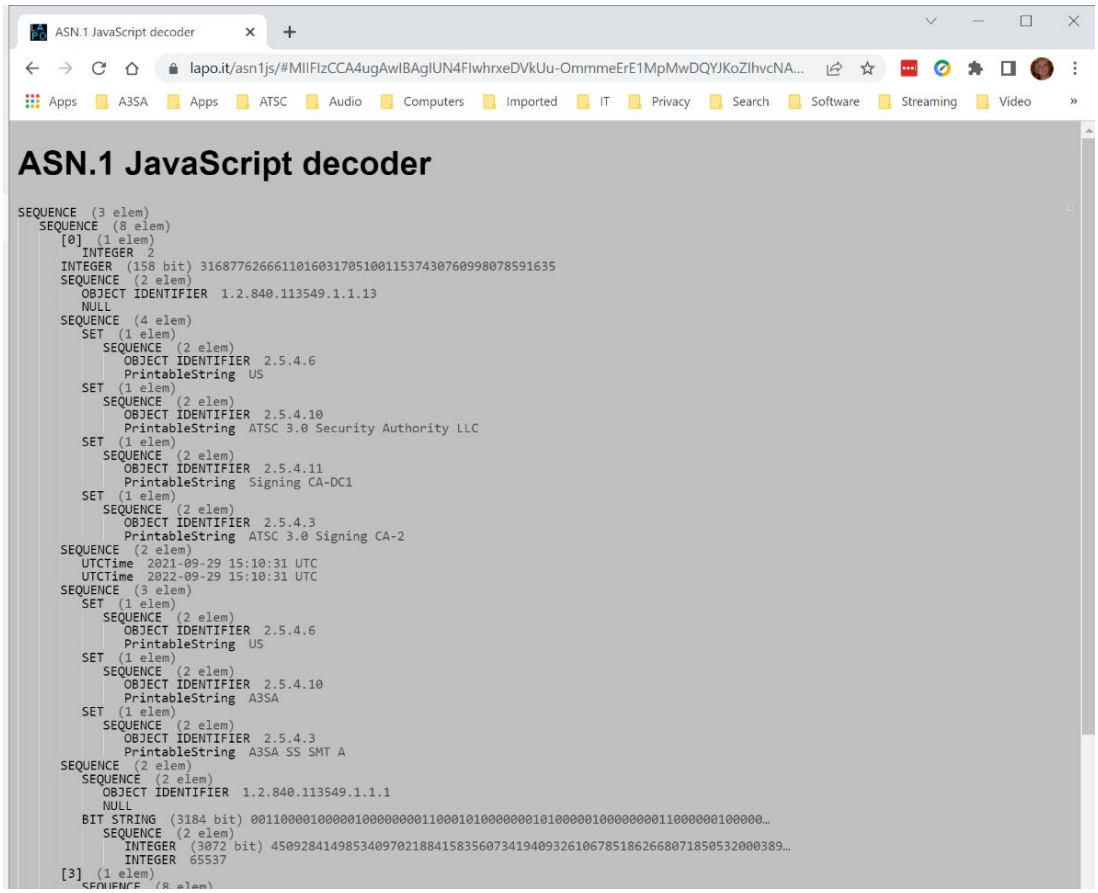


Figure 9. PrintableString of "A3SA SS SMT A": the SMT Primary Certificate

6.3.3.3. Parse the X.509 Certificate Using OpenSSL

Once you have checked that the body of the extracted text decodes without error, the next step is to create a new text file for each Certificate. For each of the three lines with

```
<Certificates> <cert body data> </Certificates>
```

Repeat the following steps:

1. Create a new text file using a familiar text editor. On Windows, Notepad++ is a possible choice ((https://notepad-plus-plus.org/).
2. Copy to your clipboard all the text between the <Certificates> tags in the XML export download. Make sure to copy any "=" characters at the end of the string.
3. The text files need to be limited to 64 characters per line. A variety of methods can be used to do this; for example, you can paste the clipboard to the new text file and break the lines after 64-characters manually using the text editor.

   a. Alternatively you can use the following online tool: https://onlinetexttools.com/split-text
   b. Paste the Certificate body data into the web page "input text" field.
   c. Under "text splitter options" click the radio button next to “Use Length for Splitting” and set the length to 64.
   d. In the webpage "pieces" section, click "Copy to clipboard"
4. Once the Certificate body data has been split into lines no longer than 64 characters and pasted into the new text file, add a header and a footer at the beginning and end of the file:

```
-----BEGIN CERTIFICATE-----
<pasted certificate data>
-----END CERTIFICATE-----
```

5. Save the file. You can use a temporary name because at this point you may not know which certificate it is; e.g.,

```
TempCert.txt
```

   You will rename the certificate file in a subsequent step.

6. From a host running OpenSSL, parse the certificate data you just created. Enter at the command prompt:

```
openssl x509 -text -noout -in TempCert.txt
```

7. Make sure no errors are the returned.
8. The following fields should be searched for and checked for consistency with your stations' certificate application:
   a. Check the validity period (not before, not after) and ensure the dates are consistent with your intended usage.
   b. Check the Issuer Country (C = US) and Issuer Organization (e.g., O = A3SA in these examples, but more likely your organization name).
   c. Check the Subject Common Name (CN) field in particular (in the screenshot below, see `Subject: … CN =`). This should be consistent with the data entered during certificate generation. Confirm that it matches your expectations. From this you should be able to identify which Certificate this is (CDT, SMT, or ICA).
9. Rename the certificate file based on the common name identification from the prior step. For example, where "SS" is shorthand for "signal signing,"

```
ren TempCert.txt A3saPrimarySScdt01.crt
```

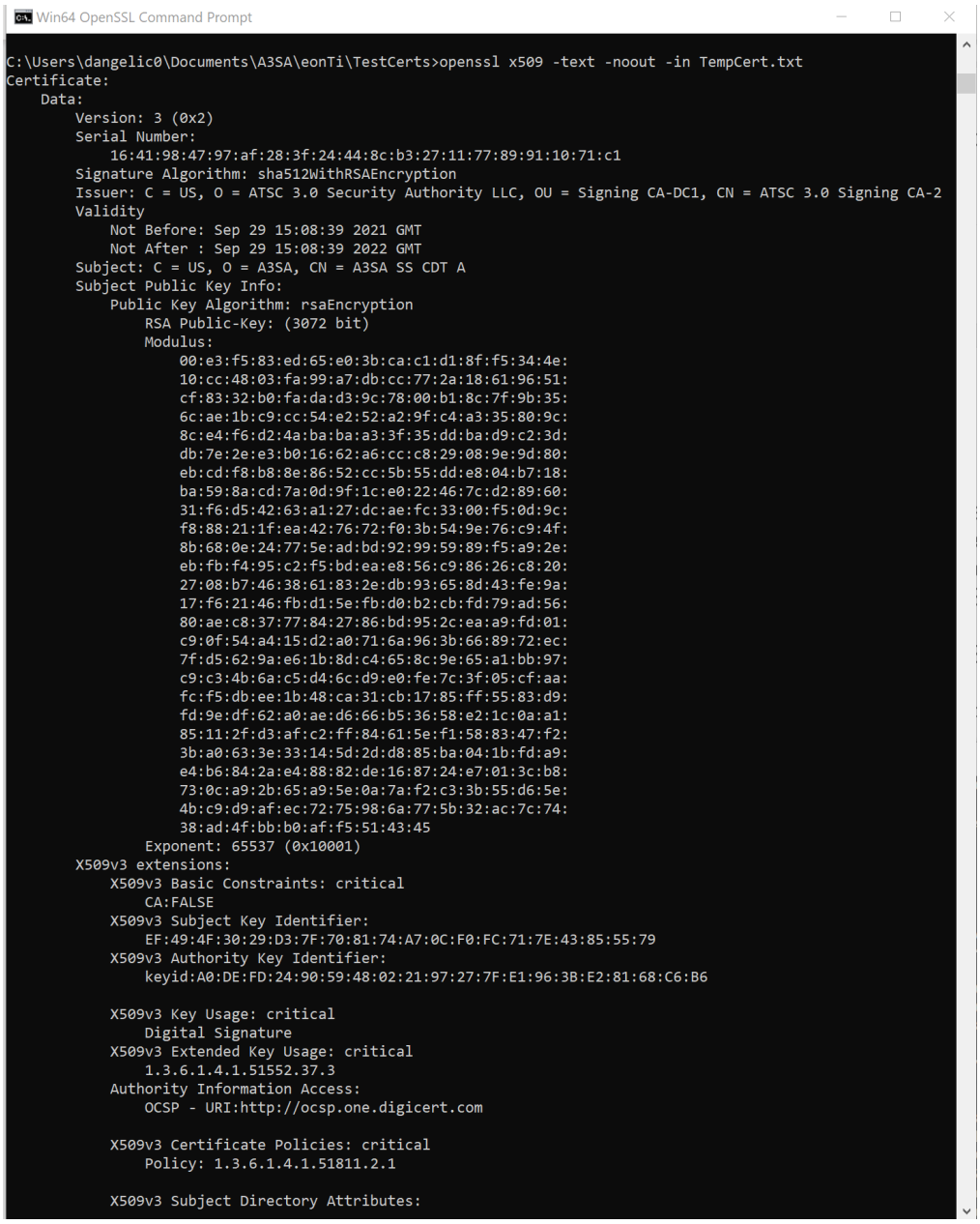


Figure 10. Screenshot with CN = "A3SA SS CDT A" the primary cert for the CDT

6.3.3.4. Check Certificate Status Using OCSP

At this point you should have three Certificate files: the CDT, the SMT, and the ICA that was used to sign the CDT and SMT Certificates. Next you should check that the online status of these Certificates is good using OCSP.

Now check their status with the A3SA OCSP responder. First find the OCSP URL and hostname. This is found within the Certificate itself. For example, in Figure 10, the "Authority Information Access:" block provides the OCSP - URI of http://ocsp.one.digicert.com. Therefore, use that URL in commands used to check the OCSP status of that Certificate.

Please note that in each of the following numbered steps:

- The example names used here are representative and use A3SA as the sample organization name. You may have a slightly different naming convention, using call letters and/or different organization names. Be sure to use filenames that are indicative of the type (primary, intermediate), purpose (signal signing, application author, application distributor), call letters, etc. (See Section 3.1).
- The exact syntax of the OpenSSL commands varies by platform OS, especially in the command entry for the Host option. Check your local documentation.

1. Check that the OCSP status of the CDT primary certificate for signal signing is "good":

```
openssl ocsp -issuer A3SA_Signing_CA_intermediate.crt -cert
   A3saPrimarySScdt01.crt -noverify -url
   http://ocsp.one.digicert.com -header
   Host=ocsp.one.digicert.com
```

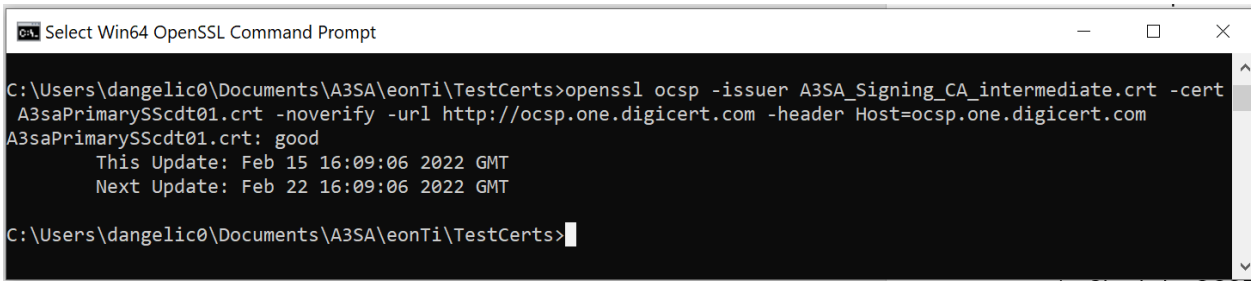


Figure 11. Example "good" Status from OCSP on CDT

2. Check that the OCSP status of the SMT primary certificate for signing is "good":

```
openssl ocsp -issuer A3SA_Signing_CA_intermediate.crt -cert
   A3saPrimarySSsmt01.crt -noverify -url
   http://ocsp.one.digicert.com -header
   Host=ocsp.one.digicert.com
```

Figure 12. Example "good" Status from OCSP on SMT

3. Check that the OCSP status of the signing intermediate CA certificate itself is also "good". In this case, you will need to provide as issuer the root CA you received from Eonti (see Section 4.2.2).

```
openssl ocsp -issuer ATSC3_Root-CA_root.crt -cert
   A3SA_Signing_CA_intermediate.crt -noverify -url
   http://pki-ocsp.digicert.com -header Host=pki-
   ocsp.digicert.com
```


Figure 13. Example "good" Status from OCSP on ICA

4. Repeat the above three OCSP commands but this time store the output in a text file:

```
openssl ocsp -issuer A3SA_Signing_CA_intermediate.crt -cert
   A3saPrimarySScdt01.crt -noverify -url
   http://ocsp.one.digicert.com -header
   Host=ocsp.one.digicert.com > A3saPrimarySScdt01_Status.txt

openssl ocsp -issuer A3SA_Signing_CA_intermediate.crt -cert
   A3saPrimarySSsmt01.crt -noverify -url
   http://ocsp.one.digicert.com -header
   Host=ocsp.one.digicert.com > A3saPrimarySSsmt01_Status.txt

openssl ocsp -issuer ATSC3_Root-CA_root.crt -cert
   A3SA_Signing_CA_intermediate.crt -noverify -url http://pki-
   ocsp.digicert.com -header Host=pki-ocsp.digicert.com >
   A3SA_intermediate_Status.txt
```

   These files should be stored for your record keeping with the certificates.

6.4. Verification Using ATSC 3 TVs for the Consumer Market

For this verification step, you will need access to one or more consumer ATSC 3.0 televisions. Performing this step with TVs from multiple manufacturers, some with signaling signing validity checking features and some without, is recommended.

6.4.1. Models

This step is not intended to describe a complete interoperability test, of course, but you should test your newly installed and configured system on multiple consumer market TVs from different manufacturers. At the time of this writing, the following manufacturers have produced consumer market televisions with signaling signing validity checking features:

- LG Electronics
- Samsung
- Sony
- Hisense

6.4.2. Setup

The mechanism to enable application or signaling signing validity checking varies from manufacturer to manufacturer. Please refer to the vendor-specific documentation for the ATSC 3 televisions that you are using to verify broadcaster setup of signal signing.

6.4.3. Procedure

The process of verifying your broadcast setup of signal signing follows these high-level steps:

1. Enable signal signing in the broadcast and confirm using a TV with signature checking capability enabled that it shows no consumer warning.
2. If available, use an older TV that does not have the signature checking capability at all, and check that playback remains normal after signal signing has been enabled in the broadcast.

It is recommended to repeat these steps for TVs from multiple manufacturers.

Complete the Broadcaster Readiness Questionnaire and submit to A3SA at compliance@a3sa.com as soon as possible.

Appendix A – Compliance Certification

This Appendix is representative of the Broadcaster Readiness Questionnaire for NextGen TV Signal Signing [1]. To obtain the actual questionnaire as an editable fillable form email compliance@a3sa.com.

This form is used to record verification test results and to self-certify that (1) broadcaster installation and configuration of signal signing, and (2) all broadcasts that will be signed, are both is in conformance with the ATSC 3.0 technical specifications/standards. When complete, sign the form and submit by emailing this to compliance@a3sa.com.

PLEASE COMPLETE ALL FIELDS ACCURATELY

1. Station Information

Organization:	Facility ID:
Phone:	E-mail:
Address 1:	Address 2:
City, State:	Zip:
Call Signs (List all that apply):	

2. Tester Information

Name:	Title:
Phone:	E-mail:
Address 1:	Address 2:
City, State:	Zip:
Org Name (if different than above):	

3. Verification Results Using Signing Engine OCSP (Section 6.1)

Signing Engine Make:	Model:	Software Version:
☐ Yes – this signing engine has certificate status checking using OCSP and it showed ***"good"*** for both CDT and SMT certificates.		
☐ No – describe:		

4. Verification Results Using IRD Validity Checking (Section 6.2)

IRD Make:	Model:	Software Version:
☐ Yes – this IRD has signature validity checking and it showed ***"unsigned"*** for both CDT and SMT ***before*** enabling service signal signing.		
☐ No – describe:		
☐ Yes – this IRD has signature validity checking and it showed ***"Valid"*** for both CDT and SMT ***after*** enabling service signal signing.		
☐ No – describe:		

5. Verification of Certificate Extraction and Validation (Section 6.3)

Enter IRD tool used for Certificate Extraction: IRD Make: Model: Software Version:
☐ Yes – the following certificates were all ***extracted*** without issue: Root-CA, Signing CA intermediate, Signaling Signing primary for CDT, and Signaling Signing primary for SMT ☐ No – describe:
☐ Yes – all extracted certificates (Root-CA, Signing CA intermediate, primary for CDT, and primary for SMT) ***decoded*** without issue, and upon inspection those data elements listed in Procedure 6.3.3.2 (Step 5) were consistent with expectations. ☐ No – describe:
☐ Yes – all extracted certificates (Root-CA, Signing CA intermediate, primary for CDT, and primary for SMT) ***parsed*** without issue as per (e.g.) Procedure 6.3.3.3. ***All*** data elements listed in Procedure 6.3.3.3 (Step 8) were consistent with expectations. ☐ No – describe:
☐ Yes – all extracted certificates (Root-CA, Signing CA intermediate, primary for CDT, and primary for SMT) show OCSP status of ***"Good"*** per (e.g.) Procedure 6.3.3.4. The status logs ***have been saved*** as have the certificates and private keys as per Procedure 3.2. ☐ No – describe:

6. Verification Using Consumer Market ATSC 3 TVs (Section 6.4.3)

TV Make: Model: Firmware Version:
☐ Yes – this TV ***has*** signature checking capability and with it ***enabled*** it showed ***no*** consumer warning ***after*** enabling service signal signing ☐ No – describe: ☐ N/A
☐ Yes – this TV ***does not have*** signature checking capability and it showed ***no*** consumer warning ***after*** enabling service signal signing. ☐ No – describe: ☐ N/A

TV Make: Model: Firmware Version:
☐ Yes – this TV ***has*** signature checking capability and with it ***enabled*** it showed ***no*** consumer warning ***after*** enabling service signal signing ☐ No – describe: ☐ N/A
☐ Yes – this TV ***does not have*** signature checking capability and it showed ***no*** consumer warning ***after*** enabling service signal signing. ☐ No – describe: ☐ N/A

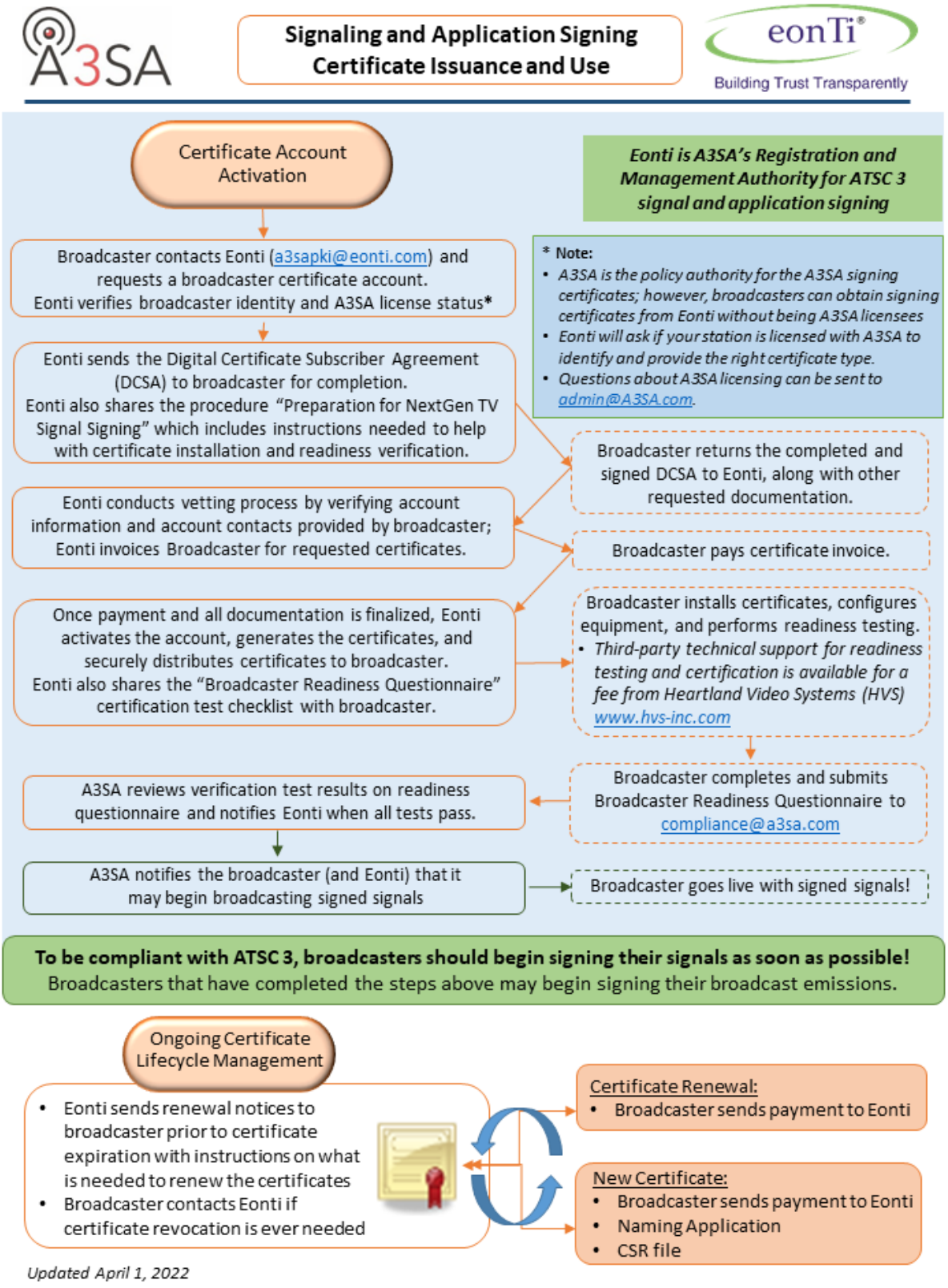
TV Make: Model: Firmware Version:
☐ Yes – this TV ***has*** signature checking capability and with it ***enabled*** it showed ***no*** consumer warning ***after*** enabling service signal signing ☐ No – describe: ☐ N/A
☐ Yes – this TV ***does not have*** signature checking capability and it showed ***no*** consumer warning ***after*** enabling service signal signing. ☐ No – describe: ☐ N/A

Self-Certification

By signing here, I certify that (1) this verification test results report is true, accurate and complete; and (2) any signal that will be signed with ATSC 3.0 Signaling Signing Certificates will be fully compliant with the applicable ATSC 3.0 technical specifications/standards:

Signature:	Date:
Name:	Title:

Appendix B – Certificate Issuance Process



Appendix C – Certificate Generation Using Third-party Tools

7. Certificate Generation Using Third-party Tools

7.1. Certificate Generation Using Triveni Products

Here we describe the certificate generation process using the Triveni GuideBuilder XM. For more information see: http://www.trivenidigital.com/products/guidebuilder-xm-atsc3-metadata-system.php.

The following steps should be followed to generate the necessary CSRs on the Triveni GuideBuilder XM. [(nearby lab pc, web gui to GBXM host)]

1. From the Transports menu in the GuideBuilder Config App, make sure to load the correct BSID for your station and services. Note that some versions of the Config App call this field the TSID, but you should enter the station BSID. This is typically the TSID minus one.Use the Licensing and Management System (LMS) on the FCC website if you need to find it. https://www.fcc.gov/media/radio/lms-help-center.
2. Open the Config App and go to Certificate Request. Click Add.
3. Give the CSR a name. See Section 3.1 on naming conventions. Then choose RSA 3072.
4. Fill in the rest of the fields (e.g., Country, state, city, Org, etc.) The Common name should also follow the naming conventions in this document as it pertains to certificate type, purpose, call letters, and signaling signing. Also, you should use different names for CDT and SMT, so including the table name short-hand in the name is a best practice.
    a. When all the data is entered, click OK.
5. Finally be sure to click Commit.

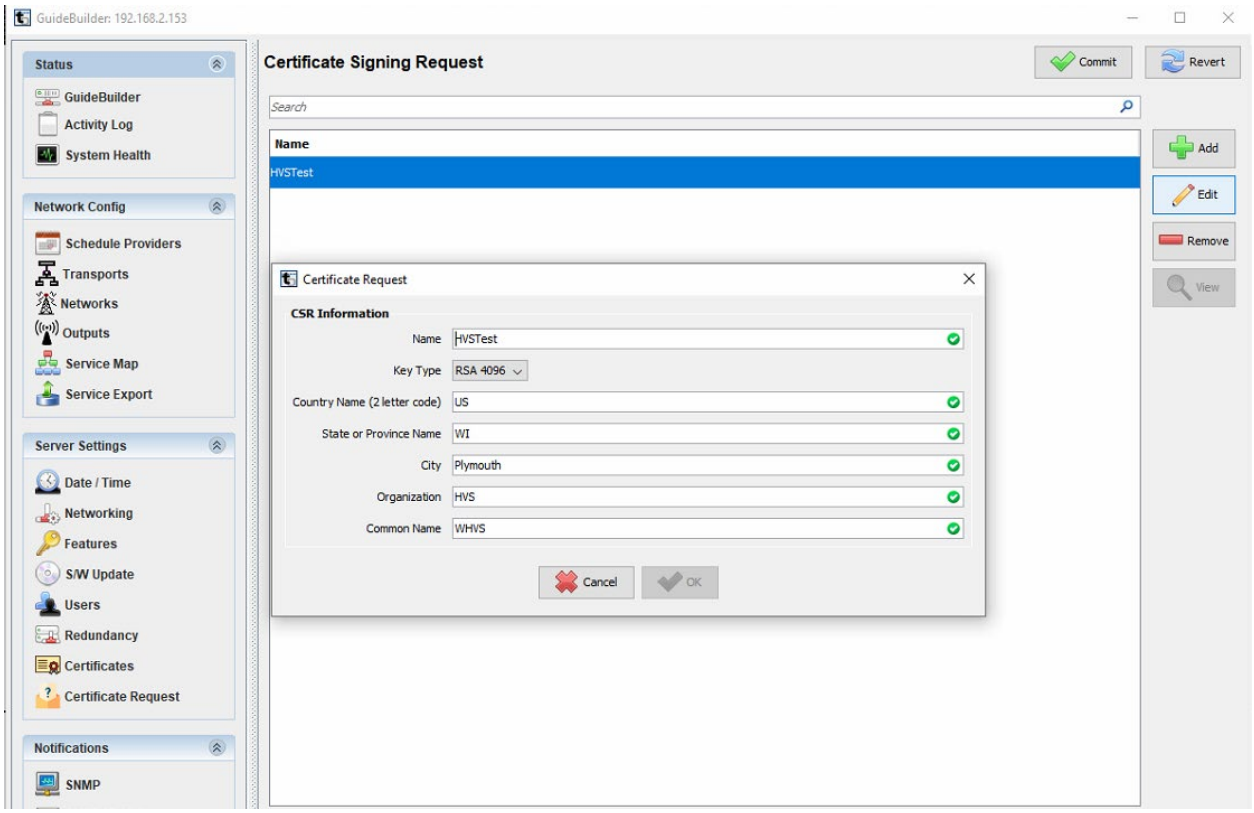


Figure 14. GBXM Certificate Signing Request

6. To export the CSR, left click the CSR you just made to highlight it and then click View. When the window comes up showing the CSR, click Download (at the bottom) and save it locally. This is the file you will send to Eonti.
7. To get the private key for your records, you need to download the database. The key is embedded in there. See GBXM user documentation for DB backup procedures.

7.2. Certificate Generation Using DigiCAP Products

DigiCAP does not have specific application support for signal signing certificate generation on their DigiCaster product. However the product is hosted on a Linux platform and they do offer a function to enter commands at a Linux command line console. Therefore you should follow the certificate generation instructions using OpenSSL described in Section 4.2.

For more information see: http://www.digicaps.com/solution/uhd-broadcasting-solution/?lang=en.

7.3. Certificate Generation Using Enensys Products

Here we describe the signal signing certificate generation process for the Enensys MediaCast ATSC. For more information see: https://www.enensys.com/products/mediacast-atsc/.

The following steps should be followed to generate the necessary CSRs on the Enensys MediaCast.

1. From the XXX menu, make sure to load the correct BSID and GSID for your station and services.
2. Go to the settings menu by clicking the Gear Icon. On the left side panel, click the Certificates link. Then click on the Key tab.
3. Enter a Key Name [ACCORDING TO SOME CONVENTION, FOR CDT, SMT]. Under Key algorithm make sure the RSA-3072 radio button is selected.

[NEED UPDATED SCREEN SHOT WITH 3072 SHOWN]

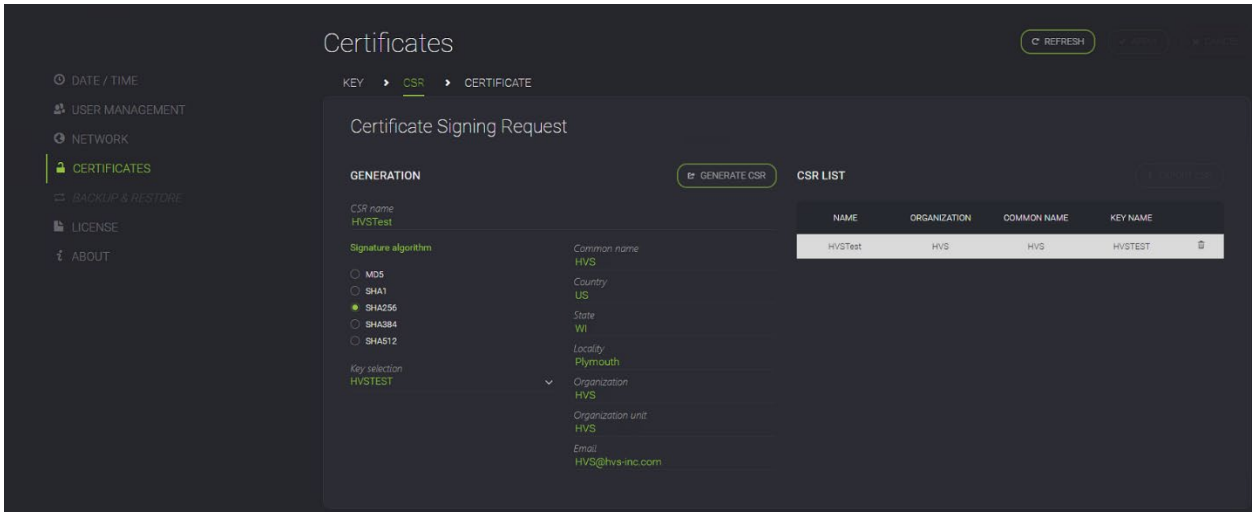
4. Click Generate Key Pair.

[NEED NEW SCREEN SHOT SHOWING RESULT OF GENERATE]

5. Click on Export Public Key and keep that for your records. See Section 3.1 for recommended practices surrounding key storage.

[NEED NEW SCREEN SHOT SHOWING RESULT OF EXPORT]

6. Now click on the CSR near the top of the middle pane



[CAPTION]

7. Choose a CSR name. Use the common name of the certificate because this will always be unique.
8. Select the algorithm: SHA256. Enter the required information indicated by red.
9. Click Generate CSR. Then click Export CSR. Store securely.

[NEED NEW SCREEN SHOT SHOWING RESULT OF THIS CLICK]

10. Send this (and other generated CSRs) to Eonti.