Birth Certificate Provisioning Service - Technical Paper

Silas Steinhauser
Gapfruit

March 26, 2026

Abstract

This document presents our Birth Certificate Provisioning service, covering two core aspects: the strategic value of embedding IEEE 802.1AR-compliant device identity in products, and the security and operational advantages of our TPM attestation-based certificate issuance flow over conventional HSM infrastructure.

Birth Certificates (IEEE 802.1AR) establish a hardware-bound, globally unique device identity from the moment of manufacture, enabling zero-touch provisioning, supply chain integrity verification, and seamless integration with zero-trust network architectures. Our TPM attestation flow further advances security by ensuring that each device's individual private key is generated and permanently bound within the device's TPM. In addition to this security advantage, the TPM-based approach eliminates the need for dedicated signing infrastructure, reduces operational risk, and scales efficiently with production volume.

Together, these two aspects position our Birth Certificate Provisioning service as a technically superior, standards-compliant, and operationally efficient solution for device identity management across the full product lifecycle.

1 Management Summary

Our Birth Certificate Provisioning service addresses a fundamental challenge in connected device security: establishing a trusted, verifiable identity for every device from the moment of manufacture. By combining IEEE 802.1AR-compliant device identity with a TPM-based attestation flow, we offer a technically robust and operationally efficient solution for device lifecycle management.

1.1 Birth Certificates in Products

IEEE 802.1AR defines a framework for Secure Device Identity (DevID), enabling each device to carry a globally unique, cryptographically bound identity in the form of an Initial Device Identifier (IDevID) - also known as Birth Certificate. Embedding this standard into products delivers several concrete advantages:

- Devices can authenticate themselves to networks and management systems without relying on shared secrets or manually provisioned credentials, significantly reducing the attack surface during onboarding and operation.
- The IDevID anchors a chain of trust that extends from the manufacturer through to the end operator, enabling zero-touch provisioning workflows compliant with standards such as RFC 8572 [3] and RFC 8995 [4].
- Interoperability is assured across multi-vendor environments, as the identity framework integrates naturally with existing PKI infrastructure and protocols including mTLS.
- Supply chain integrity is strengthened, allowing operators to cryptographically verify device authenticity and detect tampering or counterfeiting.

The result is a device identity that is tamper-resistant, standards-compliant, and compatible with zero-trust network architectures.

1.2 TPM Attestation Flow vs. HSM-Based Infrastructure

The second key aspect of our service is the mechanism by which Birth Certificates are issued. Our TPM attestation flow offers superior security compared to conventional HSM-based certificate issuance, as well as significant operational advantages:

- Stronger security for individual device keys: This is the critical distinction. In an HSM-based model, the HSM protects only the CA signing key used to issue certificates. It provides no guarantee about where the device's own private key is generated or stored - that key may reside in software or unprotected flash memory, vulnerable to extraction. With TPM attestation, the device's private key is generated inside the TPM, is hardware-bound, and can never be exported. The TPM cryptographically proves to our CA that the key was created within a genuine, certified TPM - providing end-to-end assurance over the individual device key, not just the issuance infrastructure.
- Simplified infrastructure: An HSM-based issuance model requires dedicated hardware, secure facilities, high-availability configurations, and ongoing maintenance. Our TPM attestation flow eliminates this overhead by leveraging the TPM already present on the device as the root of trust, removing the need for centralised HSM deployments.
- Scalability: TPM-based attestation scales naturally with production volume without requiring proportional investment in HSM capacity or licensing. Certificate issuance can be integrated directly into the manufacturing line.
- Reduced operational risk: HSM infrastructure introduces operational dependencies including hardware failure, key management complexity, and personnel requirements. Our service abstracts these concerns, reducing the risk surface for certificate issuance operations.

- Faster time to market: Onboarding to our service requires no capital investment in HSM hardware or custom infrastructure, enabling teams to begin issuing compliant device identities rapidly.
- Minimal dependency: The dependency on Gapfruit only exists during physical provisioning. Once the provisioning is complete, the birth certificate can be used independently of Gapfruit.

1.3 Conclusion

Our Birth Certificate Provisioning service provides a technically superior and standards-aligned approach to device identity. Unlike HSM-based alternatives - which secure only the issuance process while leaving individual device key protection unguaranteed - our TPM attestation flow ensures that every device's private key is hardware-bound from the moment of creation. Combined with IEEE 802.1AR compliance, this delivers the highest level of cryptographic assurance across the full device identity lifecycle, at scale, and without the burden of maintaining dedicated signing infrastructure.

2 Introduction

This technical paper introduces the Birth Certificate Provisioning (BCP), a process that automates the creation of TPM 2.0 backed digital identities (X.509 certificates), which are signed by a trusted root certificate authority (CA). These digital identity certificates are assigned to a single entity, are not transferable and are able to identify a device during its complete lifetime. The existing security guarantees of a TPM 2.0 are bootstrapped to the entire device by applying established provisioning techniques from the Trusted Computing Group (TCG).

After a successful execution of the BCP, the TPM of the device contains a X.509 certificate chain for a TPM-backed primary signing key. This can be used to sign further certificates or to authenticate at an endpoint, for example. Furthermore, the certificate chain will be encoded and converted into a QR code. This QR code can then be used to physically label the device.

3 Technical Prerequisites

3.1 TPM 2.0

As previously stated, a prerequisite for the service is the availability of a TPM 2.0. Furthermore, in order to bootstrap the trust, the TPM manufacturer must provision the TPM with an Endorsement Key (EK) certificate. This EK certificate is a unique identifier for the actual TPM chip. The corresponding key is recreated when needed and is not stored on the TPM.

Figure 1 shows the certificate chain of the EK certificate, which is later used to identify the TPM and its validity.

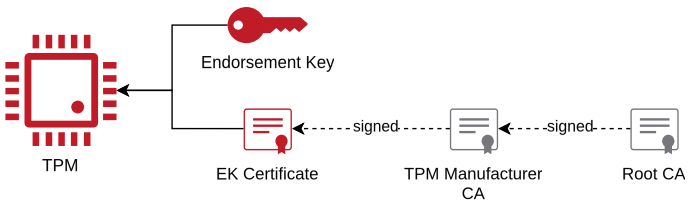

Figure 1: TPM Overview. A signed EK certificate is required.

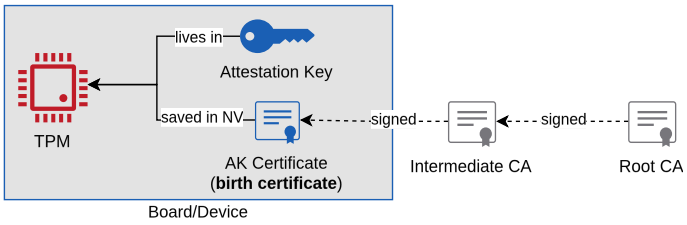

Figure 2: Desired setup for the device. The TPMs NV contains a X.509 birth certificate with a valid certificate chain.

3.2 Software Support

The BCP procedure needs to communicate with the TPM, establish a connection to the CA, and do cryptographic operations. Therefore, the Board Support Package (BSP) needs to meet the following dependencies:

- `curl` (version >= 8.0)
- `libtss2-dev` (version >= 4.0.2)
- `libtss2-tcti-tabrmd-dev` (version >= 3.0.0)
- `tpm2-openssl` (version >= 1.3.0)
- `openssl` (version >= 3.1.1)

Network Capabilities For the provisioning a functioning network setup with the right configuration is required. This includes a DNS service, correct routing as well as the correct system time.

3.3 Compatible Certificate Authority

The CA needs to support the additional requirements, which are explained in Section 5. Not all CAs are suitable. In our case, we have a partnership with Digicert[1].

4 Goal

The aforementioned EK identifies a TPM. Analogously, a second key is created that then identifies the device. The new key is known as the "Attestation Key" (AK), and its X.509 certificate is referred to as the "Birth Certificate". The objective is to create this key and its corresponding certificate. Figure 2 illustrates the desired state of the device.

In a conventional setting, a certificate signing request (CSR) is created and sent to a CA, which then creates a certificate from it. In this situation, however, the CA is unaware of whether the key that signed the CSR is stored in a TPM. Therefore, it is necessary to carry out further checks. The TCG outlines a set of procedures for creating and enrolling an AK and certificate. These procedures ensure that the CA can trust that the key in the CSR originates from the TPM.

5 Provisioning Flow

The provisioning flow follows the procedure described in the TCG specification [2] in Section “6.1 OEM Creation of an IAK Certificate” using the “Single Round-Trip Variation”.

There are many different values and other data that require careful handling. Thus, the specification mentions the RFC 7030 “Enrollment Over Secure Transport”[5] as a way to enable correct

handling of the data for the procedure. The use of a "/fullcmc" endpoint is necessary. To communicate with such an endpoint, "Full PKI Request" and "Full PKI Response" are required. These concepts are described in yet another RFC 5272[6]. This RFC specification profiles certificate enrollment for clients using Certificate Management over CMS (CMC) messages over TLS. It clearly defines how the request needs to be structured.

Figure 3 shows a detailed overview of the complete enrollment flow.

6 Results

The TPM will be in the following state after a successful execution of the BCP. Note that this is in compliance with the TCG specification section "7.3 TPM Persistent Objects"[2]:

- The AK is saved on the TPM at the NV-index `0x81020001`.
- The birth certificate and the other certificates required to create a valid certificate chain are stored on the TPM as X.509 compliant certificates. They are DER encoded and saved at NV-index `0x01C90100` and following, if it does not fit in a single index. The root certificate is not saved in order to preserve NV space.

Note: No TPM access policies for the AK and certificate have been set. It is the customer's responsibility to ensure that the product is handled correctly and to specify suitable policies or passwords. It should be noted that this means that the AK and its certificate can be removed from NV, for example when clearing the TPM. While the AK can be recreated, this is not possible for the certificate. In this instance, it will be necessary to reprovision the device.

An optional QR code is generated which can be used to physically label the device. The value of the QR code can simply be the serial number of the device, or it can contain the full certificate chain. For the second option, the certificate chain on the QR code is first DER encoded, and to ensure it fits the limited space, it is further C509 encoded[7]. Note that the C509 standard is still being developed by the IETF and needs to be considered "work in progress". Nonetheless, different groups, such as the Security Working Group of the LoRa-Alliance[8], are currently considering this standard for the QR code of the devices.

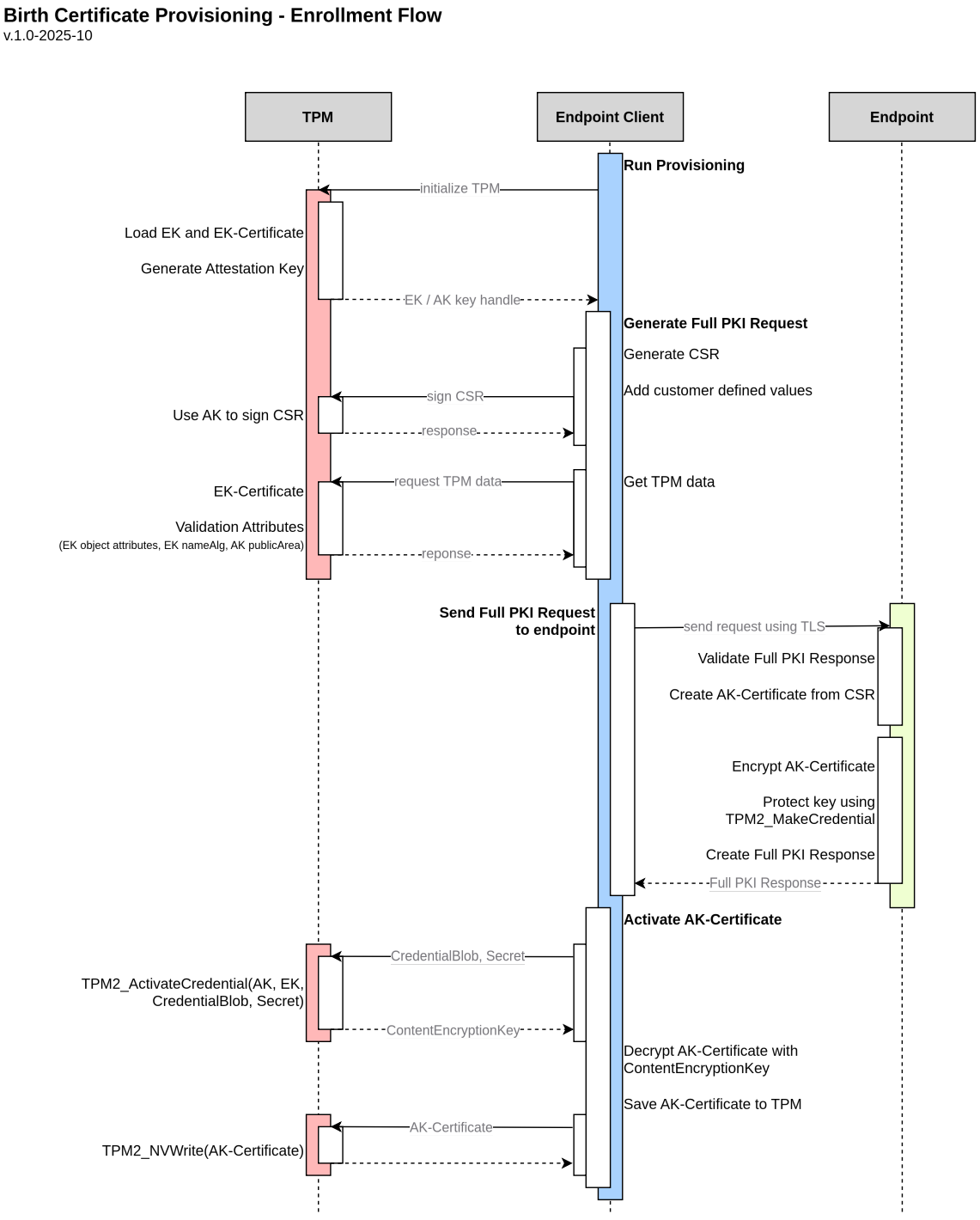

Figure 3: Flow-diagram of the complete provisioning flow.

7 Business Interface

In this section, additional, nontechnical requirements are discussed. These are required for the provisioning to function correctly. Other configuration options also exists, for which secure defaults are provided. The Figure 4 gives a complete overview.

Boot Image

The BSP recipes for the boot image (e.g. Yocto or Buildroot) needs to fulfill all technical requirements as mentioned in the Section 3. It enables the BCP binary to be executed on the device. Further the root certificate from gapfruit is added to the boot image. This is essential to verify the correctness of the BC.

Endpoint Credentials

Requests to the "/fullcmc" endpoint are only accepted when proper credentials are used. Using such credentials provides two advantages. Firstly, only authorized devices can use the endpoint to create a BC. Secondly, it is used to enforce that only a predefined number of devices can be provisioned. So before using the service, a credential needs to be acquired through Gapfruit and a maximum number of times that this credential can be used is defined.

BCP binary

The actual program that completes the flow described in Section 5.

Device Serial Number

Even though this point is optional, the customer is encouraged to use serial number to identify each of their devices. This serial number will be part of the certificate.

OpenSSL CSR Configuration

OpenSSL, the provider for cryptographic operations, defines a configuration standard to specify, among other values, the distinguished name[9]. The specified values here will then be present on the birth certificate. Even though only the common-name attribute is required, it is good practice to provide more values. The distinguished name includes values about the device and its organization. Values include:

- Common Name: e.g. "imx8mp"
- Country Name: e.g. "US"
- Locality Name: e.g. "New York"
- Province Name: e.g. "New York"
- Street Name: e.g. "Company-Street 123"
- Organization Name: e.g. "Company Name"
- Organizational Unit: e.g. "Engineering"
- Serial Number: see requirement above

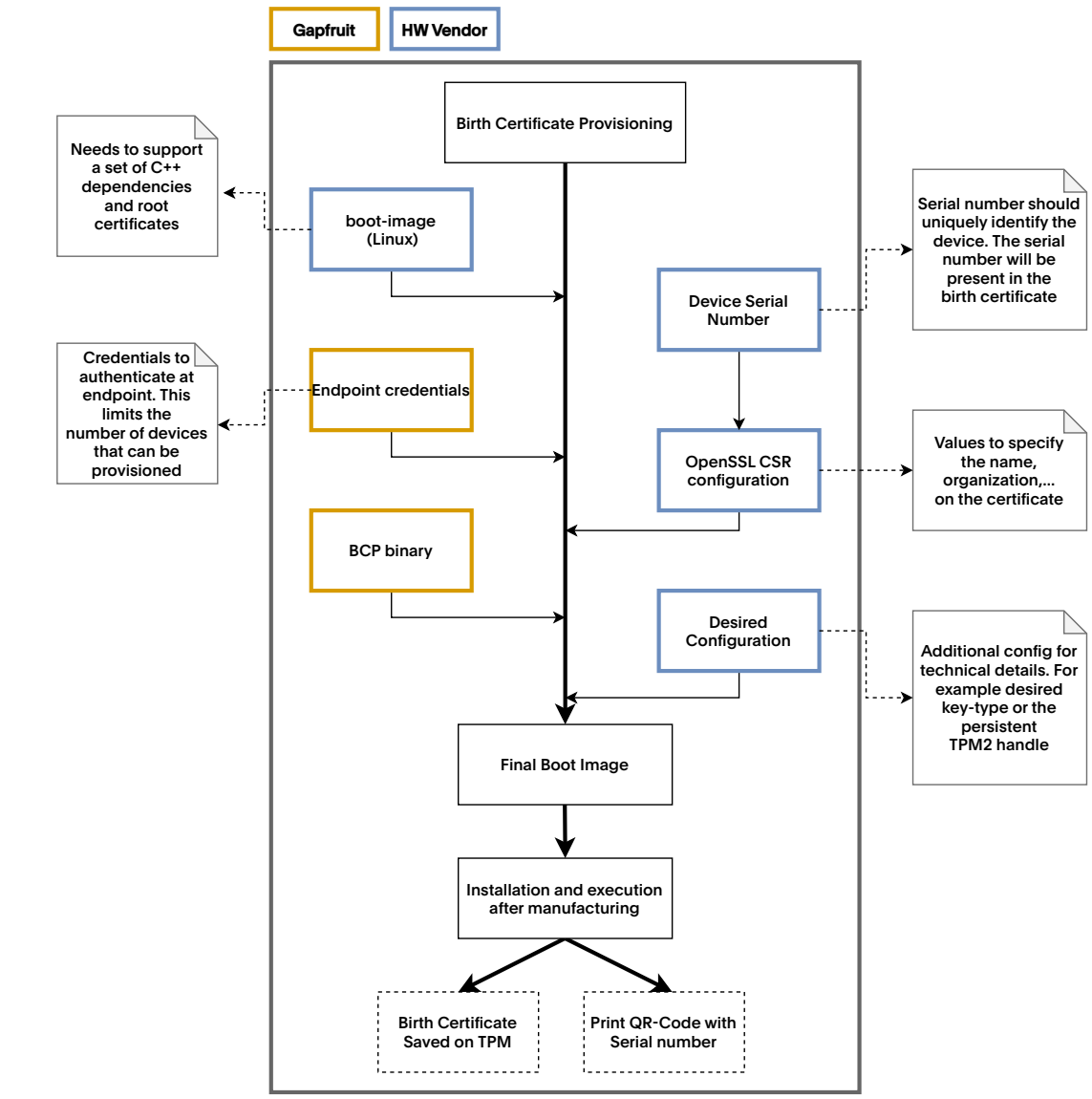

Figure 4: Diagram of the Business Interface, showing all required inputs for the service to work.

Desired Configuration

Gapfruit uses secure default settings that comply with current standards. This configuration is sufficient for most use cases. However, there may be situations with special requirements. Should these arise, contact us to find a solution.

For example, by default the AK key type will be "ECC NIST P256". The advantage include its small key-size and fast creation time. This is especially valuable in constrained environments such as a TPM. But some system or endpoints might only support "RSA-2048" keys. In such a case the configuration can be changed to use other key types.

8 Quality Assurance

Several measures are taken to ensure secure and proper execution of the provisioning. The measures are divided into four categories: development, pre-provisioning, provisioning, and post-provisioning. In general, if a quality assurance check fails, the system tries to recover. If this is not possible, the execution stops and provides the reason.

8.1 Development

Some measures are taken, even before the binary of the BPC is created. Best practices for coding are used. Namely, as is custom, code changes on how the provisioning works require to pass a pipeline. This pipeline runs many tests, for which a local endpoint is used, to ensure that the provisioning works as intended and no breaking changes are introduced. Further, a weekly integration test is executed that creates a birth certificate using Digicerts endpoint.

8.2 Pre-Provisioning

Once the binary is executed, the following points are checked:

- The NV memory of the TPM is read. No provisioning will be done if it contains a certificate signed by a gapfruit-root-certificate . An overwrite flag can reset the NV if some other data is present to overwrite its content.
- A simple network check is done, to ensure that a secure connection to the endpoint is possible.
- Check the NV memory of the TPM for EK certificates.

8.3 Provisioning

As is evident, more checks are carried out than are listed here. It is merely a subset of the most relevant ones:

- One of the first steps of the provisioning is to create a certificate signing request (CSR). We double check that the CSR is for the AK of this device.
- The HTTP headers of the response from the end point is checked to ensure consistency with RFC 5272[6].
- We verify that the HTTP data can be accurately decoded and parsed into a structure that is compliant with RFC 5272: a *full PKI Response*.
- The signature of the *full PKI Response* is verified using the gapfruit-root-certificate.
- *Full PKI Response* handling is carried out in accordance with RFC 5272. For instance, message objects are only processed if the control sequences of the *full PKI Response* are correct.
- Following the extraction of the BC from the *full PKI Response*, it is verified that the AK matches the birth certificate.
- The certificate chain from the BC to the gapfruit-root-certificate is validated.

8.4 Post-Provisioning

Following the provisioning stage, it is essential to ensure that the device is in the desired state:

- Load the BC from NV memory of the TPM at the predefined index.
- Verify the certificate chain from the BC to the gapfruit-root-certificate.

If all of the aforementioned checks are successful, the provisioning process is complete. An additional script is provided that utilizes the AK and the BC. This guarantees that the TPM object functions as designed and serves as an example for utilising them.

It is also recommended, to add protection to the BC using a policy or locking down the NV.

References

[1] Digicert - Global provider of high-assurance TLS/SSL, PKI, IoT and signing solutions. `https://www.digicert.com/`

[2] TCG - TPM 2.0 Keys for Device Identity and Attestation `https://trustedcomputinggroup.org/wp-content/uploads/TPM-2.0-Keys-for-Device-Identity-and-Attestation-v1.10r9_pub.pdf`

[3] RFC 8572 - Secure Zero Touch Provisioning (SZTP) `https://www.rfc-editor.org/rfc/rfc8572`

[4] RFC 8995 - Bootstrapping Remote Secure Key Infrastructure (BRSKI) `https://www.rfc-editor.org/rfc/rfc8995`

[5] RFC 7030 - Enrollment over Secure Transport `https://www.rfc-editor.org/rfc/rfc7030`

[6] RFC 5272 - Certificate Management over CMS (CMC) `https://www.rfc-editor.org/rfc/rfc5272`

[7] CBOR Encoded X.509 Certificates `https://datatracker.ietf.org/doc/draft-ietf-cose-cbor-encoded-cert/`

[8] LoRa Alliance - Security Working Group. Login required to access resource `https://members.lora-alliance.org/wg/Security-wg`

[9] OpenSSL Configuration `https://docs.openssl.org/3.1/man5/config/`