escrypt
SECURITY. TRUST. SUCCESS.

Whitepaper

How CycurHSM increases the security level of certificate handling.

Implementing chains of trust with CycurHSM

From version 2.5.6 onwards, CycurHSM supports X.509 certificate security use cases. Users may implement chains of trust based on certificates that are protected by the HSM. With the functionality provided it is possible to build secure authentication or authorization services.

It is vital to understand how an HSM can increase the security level of certificate handling, since certificates are by definition public elements: A set of certificates – called parent or "root" certificates – are part of the HSM code flash. Therefore, they are part of the protected space and become part of the HSM root of trust.

The basic operation of secure certificate handling is the injection of a certificate. In this process, the HSM copies the certificate into its secure RAM and checks the authenticity of the certificate against parent and injected certificates (digital signature verification). It is possible to build a secure ECU certificate store using this feature. That is, the trust level of the HSM firmware is transferred to the certificates. It is possible to perform basic parsing operations on injected certificates. CycurHSM supports reading out the issuer, subject and serial number fields. X.509 custom extensions not known to CycurHSM are not supported or ignored.

Contents

Building a secure certificate storage

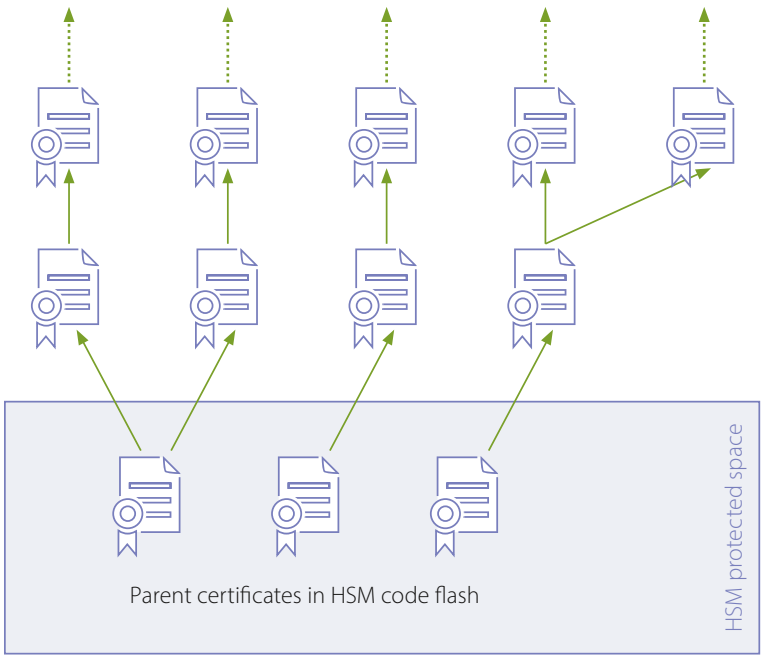
It is possible to secure a large set of certificates with CycurHSM. However, the user must be aware that the available code flash of the HSM is typically very limited. A typical setup is to offer three slots for X.509 v3 certificate with a size of 1536 bytes. The actual value may vary and is listed in the delivery notes shipped with your CycurHSM variant.

In order to secure a larger set of certificates they are stored on the application side, preferred within a consecutive memory block. Optionally, the CycurHSM trusted boot feature may be used to ensure the authenticity of this memory block. The user defines a trusted boot software part covering the certificates. The HSM can check this block during boot or in the background during runtime (RTMD).

It is possible to configure CycurHSM to reject access to keys and certificates if a manipulation is detected. An attacker is no longer able to replay broken certificates or to insert a manipulated chain of trust.

This approach already establishes a trusted certificate store even without the X.509 certificate feature. If this approach is not feasible, e.g. because the OEM software update specifications do not allow for correct handling of the certificate store within in the ECU bootloader, the X.509 certificate feature can be used. The trust level is established if certificates on the application side are part of certificate chains linked to the parent certificates within the HSM protected space as depicted below.

Certificates stored on application side



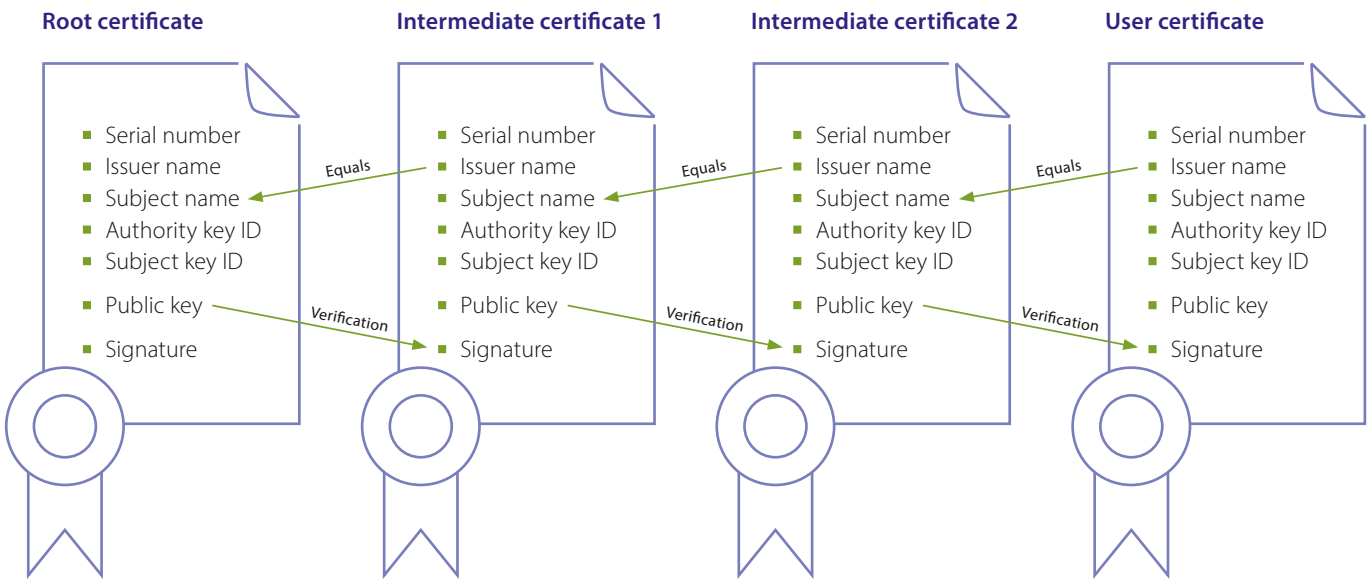
Processing certificate chains

When processing certificate chains, the typical hardware limitations of the HSM must be considered: The available secure RAM is limited for most HSM systems, with a typical value of 40 kB. The space reserved for injecting keys or certificates is typically limited to 5kB. The actual value may vary and is listed in the delivery notes shipped with your CycurHSM variant as well.

To accommodate for this limitation the processing of a chain uses a element-by-element approach. Starting from a certificate that is signed by a certificate in the HSM, the user can search and extract the next certificate in the chain on the application side. CycurHSM will consider the previously injected certificate in the "root certificate search". After injecting this second certificate, it is recommended to release the previous one to free up RAM. By following this scheme, the user may process certificate chains with arbitrary length.

User code (Pseudo code)

1. Find intermediate certificate N in certificate store
2. Inject intermediate certificate N into HSM
3. Find intermediate certificate N+1 in certificate store
4. Inject intermediate certificate N+1 into HSM
5. Release intermediate certificate N from HSM (free up RAM)
6. Repeat steps 3-5 until desired user certificate is reached
7. Perform desired operation on the certificate
   - Using CycurHSM certificate parser (extract public key or basic information)
   - Using user code / application sided certificate parsing lib, e.g. CycurLib



Certificate signing requests

It is possible to establish certificate based ECU authentication using CycurHSM. The system security design typically incorporates certificate signing requests (CSR). The CSR is created during ECU production and / or in the field, if a certificate needs to be renewed.

The security critical operation for creating a CSR is the generation of a asymmetric key pair. CycurHSM supports the generation of RSA and several ECC based key types. The user cannot extract the generated private key from the HSM, while it is possible to export the public key to the application side.

The typical steps for generating a CSR thus are:

1. Request generation of a key pair from CycurHSM
2. Extract generated public key
3. Create CSR data structure
4. Request signature of CSR structure using the generated private key
5. Complete CSR with the signature

Note that the format of the CSR is customer-specific and not security critical. Therefore, the responsibility for creating the structure and respective content is with the user.

There are two possibilities to store the CSR, if needed.

Storage of CSR using the X.509 feature

After creating the CSR, the user communicates it to the respective PKI where an identity certificate is created. This certificate is communicated to the ECU. The user now injects this certificate into the HSM. This is possible because the HSM includes a parent certificate of the PKI that has the correct public key for verifying the signature of the identity certificate. The user persists the injected certificate to one of the configured X.509 certificate slots. This method does not store the CSR itself, but the requested certificate.

Storage of CSR using the generic secure data store

If the CSR itself should be stored (in addition), the generic secure data store can be used. The prerequisite is that the CycurHSM variant has a sufficiently large data block configured. After creating the CSR, the user will persist the CSR without further processing to this data block. It is possible to configure the data block as “write-once”. With this, the CSR may be stored e.g. during production and cannot be changed afterwards. The user may read the CSR at any time to request a certificate. It is also possible to store the requested certificate in such a generic block. However, with this approach it is not possible to build chains of trust as described in section 2 of this document. ■

Summary

CycurHSM offers a strong approach to realize OEM certificate service requirements. Using CycurHSM's X.509 certificate processing offers the following key benefits:

- Creating a strong root of trust based on certificates for e.g. firmware authentication
- Manipulation of trust / certificate stores is prevented
- Toolbox approach gives the user a high degree of freedom for customization

Contact

ESCRYPT GmbH
Headquarters
Wittener Str. 45
44789 Bochum, Germany
Phone: +49-234-43870-200
Fax: +49-234-43870-211
www.escrypt.com

Subsidiaries and Technical support
For details of your local sales office as well as your local technical support team, take a look at the ESCRYPT website:
Local office: **www.escrypt.com/en/contact**
Technical support: **embedded@escrypt.com**

V2.6.7 R01 EN – 2020-01