

TPM 2.0 PLATFORM CERTIFICATE VERIFICATION TOOLS

This file describes typical use cases for this TPM 2.0 version of the Platform Certificate Tools package.

The first three use cases indicate how the tools may be used together in a manufacturing setting to obtain the EK Certificate from the platform, create the Platform Certificate, and then verify the binding between the two certificates. It will also generate XML files containing the information in the platform certificate to more easily access information about the certificates.

The fourth use case shows how the tools may be used out in the field to verify that the Endorsement Key (EK) Certificate and the Platform Certificate match.

1. GET EK CERTIFICATE, VERIFY EK AND VERIFY EK CERTIFICATE SIGNATURE

This tool will get the EK Certificate and public key (from TPM or input file), verify against the CA certificate and make sure the EK matches the EK in the TPM

- Inputs:
 - EK Certificate CA Certificate
 - (OPTIONAL) EK Certificate
 - (OPTIONAL) EK Cert Index Indicator (1=RSA, 2=ECC, **Default 1**)
 - (OPTIONAL) EK Verification Method (1= Make/Activate Credential, 2= Sorted session, **Default 2**)
 - (OPTIONAL) TPM owner password (**Default is NULL**)
 - (OPTIONAL) TPM endorsement password (**Default is NULL**)
- Outputs
 - Notification of EK cert signature verification success or failure
 - Notification of TPM EK is valid
 - (OPTIONAL) Notification of input EK matching TPM EK (If EK cert was input)
 - (OPTIONAL) EK Certificate PEM formatted file
 - (OPTIONAL) EK Certificate XML representation file
- Actions
 - If no input EK Cert: Get the EK Cert from TPM NV
 - If input EK Cert: Load EK Cert (PEM)
 - Verify the EK Cert against the input CA Certificate Chain
 - Verify EK (Sorted session, or Make/Activate Credential)
 - If input EK Cert: Compare the EK in the Cert with the EK in the TPM
 - If output EK: Output the EK in PEM and XML formats

```
verifyEKinTPM.sh -ekcca <filename> [-ekc <filename>] [-ekcout <filename>] [-ekcxmout <filename>] [-ekindex <1  
| 2>] [-ekmethod <1 | 2>] [-ownerpw <password>] [-endpw <password>] [-v]
```

-ekcca <filename>	where the file contains a list of certificate files included in the EK certificate signing chain
-ekc <filename>	where the file contains the EK certificate
-ekcout <filename>	where filename is the name of the output EK Cert PEM file
-ekcxmout <filename>	where filename is the name of the output EK Cert XML-formatted file
-ekindex <1 2>	The built-in EK certificate “index” indicating which EK certificate in the NV to use, RSA, or ECC. 1 for RSA and 2 for ECC. This is not a required option. If not included on the command line, the code will attempt to use RSA and if not found will use ECC.
-ekmethod <1 2>	Indicates which method will be used for TPM validation. In method 1 a make credential and activate credential are performed, while in method 2 a salted session is used. This is not a required option. If not included on the command line, method 2 will be used.
-ownerpw <password>	TPM owner auth (password). This is not a required option. If not included on the command line, null will be used.
-endpw <password>	TPM endorsement auth (password). This is not a required option. If not included on the command line, null will be used.
-v	verbose mode

Example:

```
./verifyEKinTPM.sh -ekcca ./sampleFiles/CAchainRSA.txt -ekc ./sampleFiles/sample_ekcert.pem -
ekcxmout ./temp_EK_cert.xml
```

2. GENERATE PLATFORM CERTIFICATE FROM XML FILES

Use human readable/editable files to generate a valid platform certificate. The platform certificate will be signed with an input private key. The input XML files are intended to be combination of files output by the tools run in other scenarios and user edited files.

- Inputs:
 - (OPTIONAL) XML file containing EK Certificate information – issuer and issuer serial number
 - (OPTIONAL) XML file containing fields of the certificate common across multiple certificates
 - (OPTIONAL) XML file containing fields of the certificate specific to an individual device or platform (e.g. platform serial #, subjAltName)
 - Private key (for signing) in PEM (default) or DER format
 - NOTE: While each XML file input is optional, **at least one XML file must be present**

- NOTE: Together the XML files must contain the minimum information to create a valid Attribute Certificate (Should we require the minimum Platform Certificate information??)
- Outputs
 - Platform certificate (signed) in PEM (default) or DER format
- Actions
 - Parse and combine the XML files
 - Produce and sign the certificate with the combined input information
 - Output the certificate in the requested format

`platformCertFromXml.sh [-xcommon <filename>] [-xek <filename>] [-xplat <filename>] [-privkey <filename>] [-v]`

<code>-xcommon <filename></code>	XML-formatted file contains the info common across multiple certificates
<code>-xek <filename></code>	where the XML-formatted file contains the issuer and issuer serial number
<code>-xplat <filename></code>	where the XML-formatted file contains info specific to an individual platform
<code>-privkey <filename></code>	where filename is the name of the private signing key
<code>-out <filename></code>	where filename is the name of the output platform certificate
<code>-der</code>	if present the output file will be formatted as a DER, otherwise it will be PEM
<code>-v</code>	verbose mode




Example:

```
./platformCertFromXml.sh -out ./temp_plat_cert.pem -xek ./sampleFiles/sample_plat_cert_ek.xml -
xcommon ./sampleFiles/sample_plat_cert_common.xml -xplat ./sampleFiles/sample_plat_cert_platform.xml
```

3. VERIFY EK CERTIFICATE MATCHES PLATFORM CERTIFICATE HOLDER – WITHOUT TPM INTERACTION

This tool, intended to be used during the manufacturing process, will verify that a Platform Certificate and an EK Certificate match. More specifically, that the information in the EK matches the information in the Holder field of the Platform Certificate. Additionally, this tool will verify the signature on the Platform Certificate against the input CA certificate as well as ensuring the input CA certificate has not been revoked by checking against a CRL downloaded from the input CRL URL.

Note that this tool...

-  Use files output by the tools run in other scenarios.
-  Runs without any TPM interaction
-  Could find a matching platform certificate as explicit input or in a folder of platform certificates

- Inputs:
 - EK Certificate CA Certificate
 - EK Certificate
 - Platform Certificate Signing CA Certificate
 - One of...
 - (OPTIONAL) Platform Certificate or
 - (OPTIONAL) Directory of Platform Certificates
 - CRL URL
- Outputs
 - Notification of verification success or failure
 - (OPTIONAL) Platform Certificate XML representation file
 - (OPTIONAL) EK Certificate XML representation file
- Actions
 - Verify the EK Cert against the input EK CA Public Key Chain
 - Verify the Platform Certificate Signing Certificate has not been revoked
 - If Platform Cert directory, repeat following steps until a match is found or checked all files in the directory. Otherwise, check against input Platform Certificate
 - Compare the EK Cert serial number and Platform Cert Holder serial number
 - Verify the Platform Cert against the input Platform Cert Signing Public Key Chain

`verifyEKandPlatCertsNoTPM.sh -ekcca<filename> -ekc <filename> [-ekcxmlout <filename>] -pcca <filename> [-pc <filename>] [-pcd <directory>] -crlurl <URL> [-pcxmlout <filename>] [-v]`

<code>-ekcca <filename></code>	where the file contains a list of certificate files included in the EK certificate signing chain
<code>-ekc <filename></code>	where the file contains the EK certificate
<code>-ekcxmlout <filename></code>	where filename is the name of the output EK Cert XML-formatted file
<code>-pcca <filename></code>	where the file contains the platform certificate CA certificate
<code>-pc <filename></code>	where the file contains the Platform certificate
<code>-crlurl <URL></code>	where URL is the URL to where the CRL can be downloaded
<code>-pcxmlout <filename></code>	where filename is the name of the output Platform Cert XML-formatted file
<code>-v</code>	verbose mode

Example:

```
./verifyEKandPlatCertsNoTPM.sh -ekcca ./sampleFiles/CAchainRSA.txt -ekc
./sampleFiles/sample_ekcert.pem -pcca platform_ca_cert_file -crlurl https://some\_url -pc
./sampleFiles/sample_plat_cert.cer -pcxmlout ./temp_plat_cert.xml
```

4. VERIFY EK CERTIFICATE MATCHES PLATFORM CERTIFICATE HOLDER – WITH TPM INTERACTION

This tool, intended to be used in the field after platform delivery to a customer, will verify that a Platform Certificate and an EK Certificate match. Specifically, the information in the EK matches the information in the Holder field of the Platform Certificate. The EK certificate will be fetched from the NV of the local TPM. Additionally, this tool will verify the signature on the Platform Certificate against the input CA certificate as well as ensuring the input CA certificate has not been revoked by checking against a CRL downloaded from the input CRL URL.

Note that this tool...

- ✚ Use files output by the tools run in other scenarios.
 - ✚ Must be able to fetch the EK certificate from the local TPM's NV.
 - ✚ Could find a matching platform certificate as explicit input or in a folder of platform certificates.
-
- Inputs:
 - EK Certificate CA Certificate
 - Platform Certificate Signing CA
 - One of...
 - (OPTIONAL) Platform Certificate or
 - (OPTIONAL) Directory of Platform Certificates
 - (OPTIONAL) EK Cert Index Indicator (1=RSA, 2=ECC, Default 1)
 - (OPTIONAL) TPM endorsement password (**Default is NULL**)
 - CRL URL
 - Outputs
 - Notification of verification success or failure
 - (OPTIONAL) Platform Certificate XML representation file
 - (OPTIONAL) EK Certificate PEM formatted file
 - (OPTIONAL) EK Certificate XML representation file
 - Actions
 - Fetch EK Cert from TPM NV – find the correct certificate based Template
 - Verify the EK Cert against the input EK CA Public Key Chain
 - Compare the EK in the Cert with the EK in the TPM
 - Verify the Platform Certificate Signing Certificate has not been revoked
 - If Platform Cert directory, repeat following steps until a match is found or checked all files in the directory. Otherwise, check against input Platform Certificate
 - Verify the Platform Cert against the input Platform Cert Signing Public Key Chain
 - Compare the EK Cert serial number and Platform Cert Holder serial number

```
verifyEKandPlatCertsWithTPM.sh -ekcca <filename> [-ekc <filename>] [-ekcout <filename>] [-ekcxmloit  
<filename>] -pcca <filename> [-pc <filename>] [-pcd <directory>] -crlurl <URL> [-endpw <password>] [-pcxmloit  
<filename>] [-v]
```

-ekcca <filename>	where the file contains a list of certificate files included in the EK certificate signing chain
-ekc <filename>	where the file contains the EK certificate
-ekcout <filename>	where filename is the name of the output EK Cert PEM file
-ekcxmlout <filename>	where filename is the name of the output EK Cert XML-formatted file
-pcca <filename>	where the file contains the signing public key chain of platform certificate
-pc <filename>	where the file contains the Platform certificate
-crlurl <URL>	where URL is the URL to where the CRL can be downloaded
-endpw <password>	TPM endorsement auth (password). This is not a required option. If not included on the command line, null will be used.
-pcxmlout <filename>	where filename is the name of the output Platform Cert XML-formatted file
-v	verbose mode

Example:

```
./verifyEKandPlatCertsWithTPM.sh -ekcca ./sampleFiles/CAchainRSA.txt -ekc
./sampleFiles/sample_ekcert.pem -pcca platform_ca_cert_file -crlurl https://some other url -pc
./sampleFiles/sample_plat_cert.cer -pcxmlout ./temp_plat_cert.xml
```