December 2019

METHOD OF PROTECTING MACHINE CERTIFICATES ISSUED TO LINUX CLIENTS OBTAINED BY USING SCEP PROTOCOL BY ENCRYPTING WITH A TPM DEVICE

HP INC

Method of protecting machine certificates issued to Linux clients obtained by using SCEP protocol by encrypting with a TPM device

The implemented solution leverages TPM as a security facility and streamlines the process from SCEP enrollment and renewal to get certificate and private key till the private key being used by a WPA Supplicant to get authenticated and authorized to access a secure network via 802.1x protocol. During the whole process, the administrator, who manages certificates and configures network settings, just needs to configure SCEP Client and 802.1x network as normal, except two extra steps to set TPM passwords and enable TPM. Besides configuring all settings in local GUI, there is a set of command line tool. The actual administrative efforts can be further reduced by executing command lines remotely in a mass deployment scenario. The administrator can run command line remotely via a secure channel to get everything setup.

To implement the solution as streamlined and user friendly as possible, there are three components modified in the operating system, the Certificate Management tool in the operating system, the Certificate and Private Key handling of SCEP client and the Key Reading and Usage part of a WPA Supplicant.

1. The Certificate Management tool is modified to be TPM compatible. A TPM section is added in the Certificate Management tool to allow administrator to set TPM passwords, keep TPM passwords securely, enable/disable TPM and display the status of TPM device. Besides that, TPM sealed private key can be detected and shown correctly as a TPM sealed private key.

2. The SCEP client is usually used to fetch CA and enroll/renew a certificate from a SCEP server. First, it generates a pair of certificate request and private key, and send the certificate request to the SCEP server. Then the SCEP server signs the certificate request and send back the signed certificate. The modified SCEP client ensures the private key is sealed after a successful enrollment, and it never appears in clear text again in the operating system.

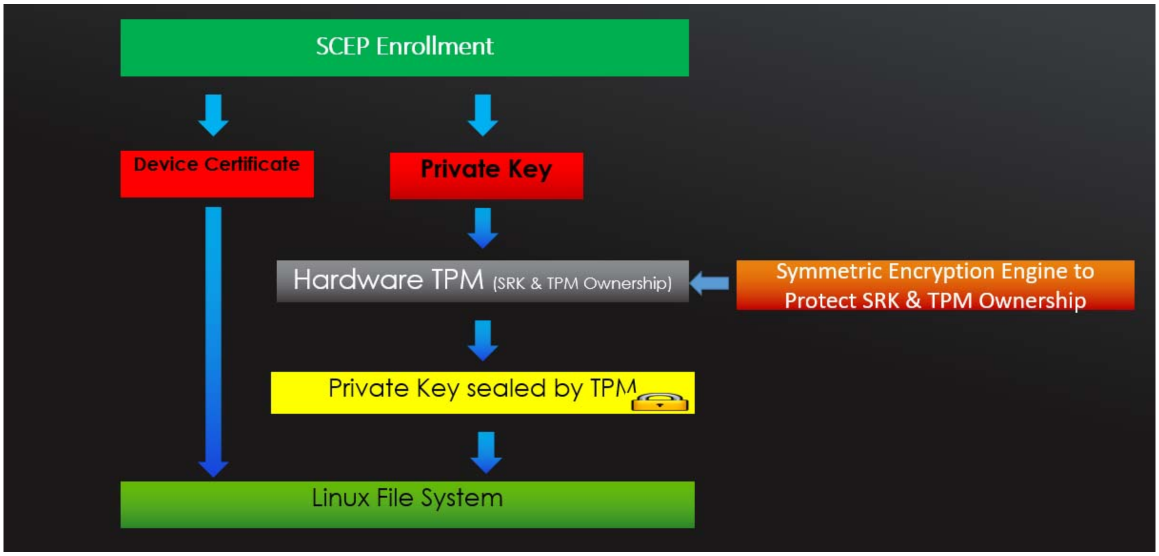
3. 802.1x-TTLS is the chosen protocol for 802.1x authentication as 802,1x TTLS uses private key to do 802.1x authentication and we can TPM seal the private key to make 802.1x authentication more secure. As the nature of TPM, even the TPM sealed private key is stolen are copied to another computer, the sealed private key is useless on another computer. The modified WPA Supplicant knows how to unseal a private key without showing the private key unencrypted anywhere in the Operating System.

When any other application in the OS needs to utilize PKI based authentication, it is possible to modify the application the same way we modified WPA Supplicant. After the application is modified, it works more securely with a TPM sealed private key.

Symmetric Engine

Symmetric Engine **(Diagram 1.0)** uses AES-256-GCM as the encryption method, which is considered a strong cipher nowadays. It runs as an background encryption service and only a short list of permitted applications are allowed to decrypt data via the encryption engine. In our particular use case, TPM’s SRK password is encrypted by Symmetric Encryption Engine to add an extra layer of security. wpa supplicant is modified to be a Symmetric Engine qualified application that can decrypt TPM’s SRK password. So every time the private key is required for doing 802.1x authentication, the modified wpa supplicant decrypts the encrypted SRK password, and then unseals the TPM protected private key with the SRK password, and in the end completes 802.1x authentication with the private key. Both private key and the SRK password are normally saved in the Operating system in an encrypted/sealed form; when they are needed for 802.1x authentication, they are decrypted/unsealed by the modified version of wpa supplicant internally, and both are not exposed in a decrypted/unsealed form in the Operating System during the whole process.

Diagram 1.0


SCEP Enrollment
Device Certificate
Private Key
Hardware TPM (SRK & TPM Ownership)
Symmetric Encryption Engine to Protect SRK & TPM Ownership
Private Key sealed by TPM
Linux File System

When any other application in the OS needs to utilize PKI based authentication, it is possible to modify the application the same way we modified WPA Supplicant. After the application is modified, it works more securely with a TPM sealed private key.

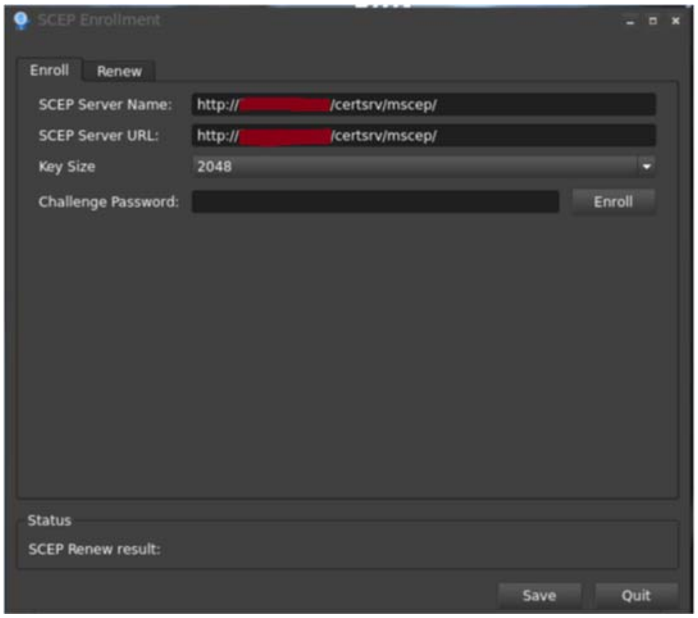
How SCEP leverages TPM in this implementation

After a successful SCEP enrollment, a pair of certificate and private key is generated. Both the certificate and private key are stored in clear text in the file system. Having a private key in clear text could be a concern in many aspects. When TPM is involved in this scenario, the private key is now sealed by the TPM. The TPM seal is a TPM feature used to protect sensitive data. Only the person who knows the TPM's SRK password can use it on the computer where the private key was originally sealed. Therefore, the private key is protected from being used without authorization or being copied and used on other computers. When the sealed private key is copied to another machine, it becomes useless. In the SCEP/TPM implementation, the private key is sealed after a successful SCEP enrollment, and afterward is never present on the filesystem in an unsealed state. Only the modified SCEP client and the modified WPA Supplicant on ThinPro can read the sealed private key to perform a SCEP renewal or 802.1x network authentication.

SCEP Components

The SCEP client used on ThinPro is an open source SCEP client called *sscep*. sscep supports the following operations: obtaining a root CA, enrolling a client certificate, and renewing a client certificate. *sscep* in this SCEP implementation is a modified version of the open source sscep. The modifications allow us to support HTTPS URL for the SCEP request and read a TPM-sealed private key. These two modifications do not exist in sscep outside of implementation.

In addition to the SCEP Client, there is a GUI tool called *hptc-scep-mgr* that allows an administrator to configure *sscep* and perform the SCEP Enrollment and SCEP Renewal. Below is the screenshot of the *hptc-scep-mgr* GUI. The *hptc-scep-mgr* application is able to accept either an HTTP or HTTPS URL.



SCEP Server Name: http://[redacted]/certsrv/mscep/

SCEP Server URL: http://[redacted]/certsrv/mscep/

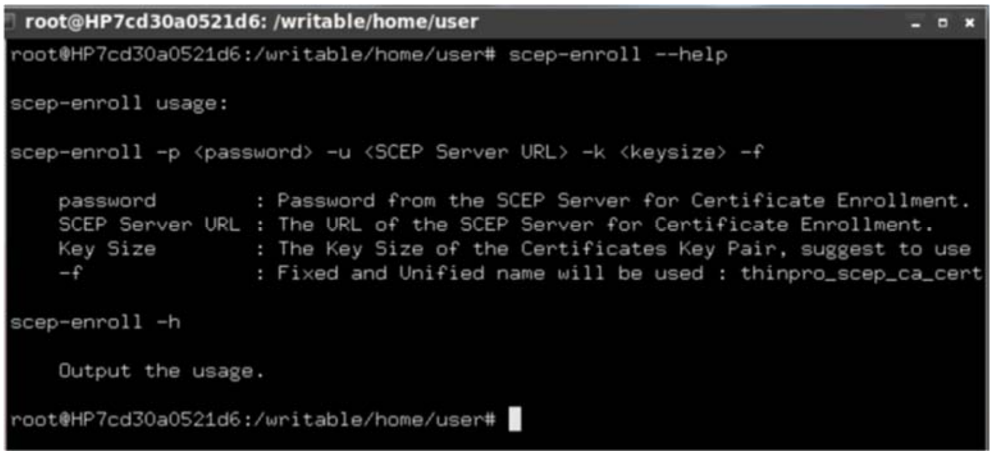
Key Size 2048

Challenge Password:

Status

SCEP Renew result:

There are three Linux shell scripts *scep-enroll*, *scep-renew* and *scep-mgr-service* that run behind the scene. The *scep-enroll* and *scep-renew* scripts call the *sscep* client to perform SCEP enrollment and SCEP renewal respectively. When SCEP auto-renewal is enabled, *scep-mgr-service* is a service running in the background to monitor the expiration date of enrolled certificates every half-day. SCEP auto-renewal is triggered when the expiration date falls into the time-frame defined in the registry. Below screenshots show the usage of *sscep-enroll*.



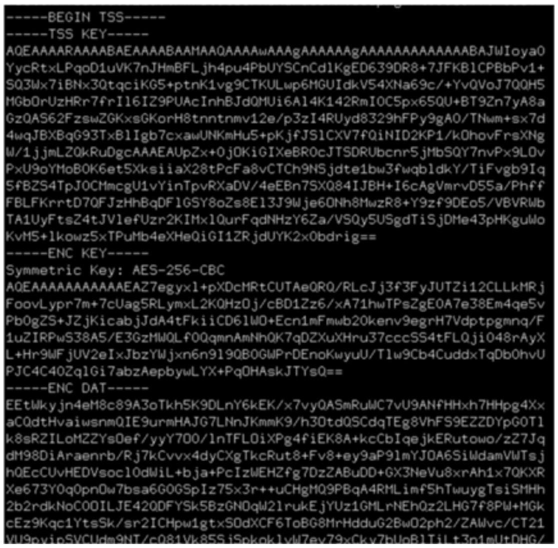
```
root@HP7cd30a0521d6: /writable/home/user
root@HP7cd30a0521d6:/writable/home/user# scep-enroll --help

scep-enroll usage:

scep-enroll -p <password> -u <SCEP Server URL> -k <keysize> -f

    password        : Password from the SCEP Server for Certificate Enrollment.
    SCEP Server URL : The URL of the SCEP Server for Certificate Enrollment.
    Key Size        : The Key Size of the Certificates Key Pair, suggest to use
    -f              : Fixed and Unified name will be used : thinpro_scep_ca_cert

scep-enroll -h

    Output the usage.

root@HP7cd30a0521d6:/writable/home/user#
```

SCEP enrollment with TPM

1. Configure SCEP Enrollment in the hptc-scep-mgr GUI, including the URL of the SCEP server, Key Size, Challenge password, and Certificate Attributes
2. Generate a private key and certificate request
3. Execute the scep-enroll script
    a. Run sscep to obtain the CA certificate. A challenge password is required to authenticate against the SCEP server.
    b. Run sscep again to enroll a certificate. A certificate request is sent out and a signed certificate is retrieved
4. The private key and signed certificate are imported into the ThinPro system

5. When the enrollment is successful, the private key will be sealed and the original (clear text) form of private key will be deleted. The sealed private key and certificate are imported into ThinPro system for further use.

Below is the screenshot of a sealed private key.



```
-----BEGIN TSS-----
-----TSS KEY-----
AQEAAAARAAAABAEAAAABAAMAAQAAAAwAAAgAAAAAAgAAAAAAAAAAAAAABAJWIoya0
YycRtxLPqoD1uVK7nJHmBFLjh4pu4PbUYSCnCdlKgED639DR8+7JFKBlCPBbPv1+
SQ3Wx7iBNx3QtqciKG5+ptnK1vg9CTKULwp6MGUIdkV54XNa69c/+YvQVoJ7QQH5
MGbOrUzHRr7frIl6IZ9PUAcInhBJdQMUi6Al4K142RmIOC5px65QU+BT9Zn7yA8a
GzQAS62FzswZGKxsGKorH8tnntnmv12e/p3zI4RUyd8329hFPy9gA0/TNwm+sx7d
4wqJBXBqG93TxBlIgb7cxawUNKmHu5+pKjfJSlCXV7fQiNID2KP1/kOhovFrsXNg
W/1jjmLZQkRuDgcAAAEAUpZx+0jOKiGIXeBROcJTSDRUbcnr5jMbSQY7nvPx9LOv
PxU9oYMoBOK6et5XksiiaX28tPcFa8vCTCh9NSjdte1bw3fwqbldkY/TiFvgb9Iq
5fBZS4TpJOCMmcgU1vYinTpvRXaDV/4eEBn7SXQ84IJBH+I6cAgVmrvD55a/Phff
FBLFKrrtD7QFJzHhBqDFlGSY8oZs8El3J9Wje6ONh8MwzR8+Y9zf9DEo5/VBVRWb
TA1UyFtsZ4tJVlefUzr2KIMxlQurFqdNHzY6Za/VSQy5USgdTiSjDMe43pHKguWo
KvM5+lkowz5xTPuMb4eXHeQiGI1ZRjdUYK2x0bdrig==
-----ENC KEY-----
Symmetric Key: AES-256-CBC
AQEAAAAAAAAAAEAZ7egyxl+pXDcMRtCUTAeQRQ/RLcJj3f3FyJUTZi12CLLkMRj
FoovLypr7m+7cUag5RLymxL2KQHzOj/cBD1Zz6/xA71hwTPsZgE0A7e38Em4qe5v
Pb0gZS+JZjKicabjJdA4tFkiiCD6lWO+Ecn1mFmwb20kenv9egrH7Vdptpgmnq/F
1uZIRPwS38A5/E3GzMWQLfOQqmnAmNhQK7qDZXuXHru37cccSS4tFLQji048rAyX
L+Hr9WFjUV2eIxJbzYWjxn6n9l9QBOGWPrDEnoKwyuU/Tlw9Cb4CuddxTqDbOhvU
PJC4C40ZqlGi7abzAepbywLYX+PqOHAskJTYsQ==
-----ENC DAT-----
EEtWkyjn4eM8c89A3oTkh5K9DLnY6kEK/x7vyQASmRuWC7vU9ANfHHxh7HHpg4Xx
aCQdtHvaiwsnmQIE9urmHAJG7LNnJKmmK9/h3OtdQSCdqTEg8VhFS9EZZDYpGOTl
k8sRZILoMZZYsOef/yyY7OO/lnTFLOiXPg4fiEK8A+kcCbIqejkERutowo/zZ7Jq
dM98DiAraenrb/Rj7kCvvx4dyCXgTkcRut8+Fv8+ey9aP9lmYJOA6SiWdamVWTsj
hQEcCUvHEDVsoclOdWiL+bja+PcIzWEHZfg7DzZABuDD+GX3NeVu8xrAh1x7QKXR
Xe673YOqOpnOw7bsa6GOGSpIz75x3r++uCHgMQ9PBqA4RMLimf5hTwuygTsiSMHh
2b2rdkNoCOOILJE42QDFYSk5BzGNDqW2lrukEjYUz1GMLrNEhQz2LHG7f8PW+MGk
cEz9Kqc1YtsSk/sr2ICHpw1gtxSOdXCF6ToBG8MrHdduG2BwO2ph2/ZAWvc/CT21
VU9pvipSVCUdm9NT/c081Vk8SSjSpkoklvW7ev79xCkv7bUoBlIiLt3p1mUtDHG/
```

This sealed private key begins with a line like “-----BEGIN TSS-----" (Trusted Software Stack), which means it is sealed by TPM and only can be unsealed by the TPM that seals it.

SCEP renewal with TPM

During the renewal process, sscep reads the old sealed private key, the old certificate, the new certificate request and the new private key. The old sealed private key is unsealed by the sscep and used in the renewal process. When the renewal is successful, the new private key is sealed by TPM. Then, the new sealed private key and the new certificate will be imported into ThinPro system for further use.

Auto-renewal and how it is triggered

When auto-renewal is enabled on ThinPro, scep-mgr-service runs in the background to check the validity of enrolled certificates every half-day. When a certificate is about to expire as per the auto--renewal time frame settings, SCEP auto renewal will be triggered.

Certificate Management after SCEP operations

After a successful SCEP enrollment – the CA certificate(s), private key, and signed client certificate will all be imported into ThinPro system. The CA certificate(s) will be copied to /usr/local/share/ca-certificates, where all other CA certificates are stored on ThinPro. The private key will be sealed and stored under /etc/tpm/certs and /etc/ssl/private, the latter being where all other private keys are stored on ThinPro. The signed certificate will be copied to /etc/ThinProCertificates and /etc/ssl/certs, the latter being where all other certificates are stored on ThinPro.

The same process occurs after a successful SCEP renewal or a successful SCEP auto-renewal.

Method of protecting machine certificates issued to Linux clients obtained by using SCEP protocol by encrypting with a TPM device

Disclosed by Zhiwei Yu, Cody Gerhardt & Michael Frick, HP Inc.