Embedded Security

Assignment 5
Solutions

Aftab Alam (2567712)
Atul Anand Jha (2572782)
Priyasha Chatterjee (2572739)

1) Tea Pee Ehm?

(a) **Done**

(b) Write a program that measures the operating system.

Find the program with name **hashextend5.c** that measures the operating system.

(c) Visit emsec.cispa.saarland/media/ and create an account there (you need the operating system integrity hash for it). Remember your credentials, you need them later on.

Done

(d) implemented the client software

Find the program with name **`Q_1_D.py.`**

First **Q_1_D.py** check itself by passing and measure itself using the hashextend5.c program.

Then call the endpoint of the given url **emsec.cispa.saarland/media/download/** and pass all the parameters and yield on error because this program is not uploaded (Not Trusted).

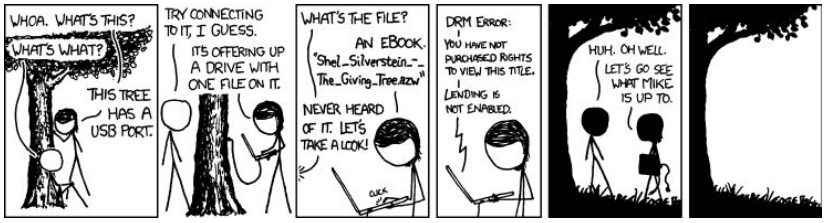
```
************************************************************
Error Your application is not a trusted application
************************************************************
```

After we upload the **`Q_1_D.py`** to the **emsec.cispa.saarland/media/download/**

Instruction to measure the **`Q_1_D.py itself just by running py program as`**

`Python Q_1_D.py`

We got success message and we print done when everything goes well and get the content from it and saved it as **content.png**. Secret is attached.



(e) The current setup has a flaw which invalidates the idea of using a TPM to establish a chain of trust. Briefly explain why and explain how to fix this problem!

In a Trusted environment, the validation of each layer is linked to the trust and validation on the previous layer in the process. Thus first the OS is measured, when the TPM knows the OS is trusted, it measures the application. In the given assignment, the validation of trust on the application and the OS are done independently. Thus there can be a change in the OS environment which will not affect the measurement of the application. Hence the secure content from the server can be downloaded as long as the integrity of the application is verified. This lack of dependency of the trust invalidates the chain of trust!

2) More TPM stuff

(a) Briefly explain the difference between sealing and encrypting data. Furthermore, give an example where sealing is wanted in contrast to encrypting.

In 'sealing' data, not only is the data encrypted, but the TPM state is also taken into consideration. An Enhanced Authorisation Policy (EAP) is used, which uses the PCRs in the TPM to protect the data. The private data that is to be securely stored, is bound to a particular TPM and a particular platform state, and can only be accessed by the specific TPM it was bound to, and when the PCR is in the same state as it was bound to.
In 'encrypting' data, the data is encrypted with a parent decryption key, and can be decrypted using a private key.

Sealing data could be wanted in contrast to encryption in order to protect a certain piece of digital data, or for Digital Rights Management purposes. For example, in protecting a music file. Sealing could be used to prevent an user from opening the file with an unauthorized player or computer. The music file would be securely encrypted using a key bound to the trusted platform module so that only the unmodified and untampered music player could be used to access it. This would prevent the user from making an unrestricted copy of the file, or unethically sharing the file.
If this file had been only encrypted, it would have been possible to access it knowing the decryption key, and it could potentially have been copied and/or distributed.

(b) Briefly explain why the restricted attribute is always set for storage keys.

The restricted attribute is set for storage keys to ensure that there are limitations on the data that can be signed or decrypted using these keys. This is done to eliminate the chances of the TPM being tricked into recognising external forged data as internal TPM data.

(c) Briefly explain why there is a certificate for the attestation key in the attestation concept.

The concept of attestation is used to provide user privacy when communicating with different sources. While the Ek could be used to secure communications, each Ek is unique to the TPM. This could potentially allow the platform's identity to be identified/recognised by the different sources being used for communication. So an Attestation Identity Key (AIK) is used as an alias for the Ek to provide user privacy when communicating with different sources. The idea of the AIK is to provide a unique identity that cannot be linked, for the TPM, for use with each different source.
The need for a certificate lies in the fact that the AIK verifies it is bound to a genuine TPM by this certificate containing the AIK public key which proves that the corresponding private key is indeed bound to a genuine TPM. This proof is guaranteed by a signature on the credential created by a trusted third party, known as a "Privacy CA".

(d) Show how a policy session can be established for the following scenarios. You should do that in the same fashion as done in the lecture.

a. Any app that is in possession of a specific password pwd and was built by developer D (with key pair sk/pk) should be given access to key K1 for the use-case of decryption.

Policy Password (pwd)
V11=H(pwd||TPM_CC_PolicyAuthValue)
Policy PCR (2,mA)
V12=H(pwd||TPM_CC_PolicyPCR||2,mA)
Policy Authorise (Sig Sk, (V12))
V13= H(0||TPM_CC_PolicyAuthorise||H(Pk))
Policy Command Code (CC_RSA_Decrypt)
V14=H(V13||TPM_CC_PolicyCommandCode||CC_RSA_Decrypt)

b. Any app running on a specific operating system (PCR1 = mOS) should be able to key K2 for decryption, as long as a user shows physical presence.

Policy PCR (2,mA)
V11=H(0||TPM_CC_PolicyPCR||2,mA)
Policy Authorise (Sig Sk2, (V11))
V12= H(0||TPM_CC_PolicyAuthorise||H(Pk2))
Policy Physical Presence (V12)
V13= H(V12||TPM_CC_PolicyPhysicalPresence)
Policy Command Code (CC_RSA_Decrypt)
V14=H(V13||TPM_CC_PolicyCommandCode||CC_RSA_Decrypt)

c. Any app that was built by a Developer D, or D' that was authorized by D, is allowed to use key K1 for signing.

Policy PCR (2,mA)
V11=H(0||TPM_CC_PolicyPCR||2,mA)
Policy Authorise (Sig Sk1, (V11))
V12= H(0||TPM_CC_PolicyAuthorise||H(Pk1))

Policy PCR (2,mA')
V11'=H(0||TPM_CC_PolicyPCR||2,mA')
Policy Authorise (Sig Sk1', (V11'))
V12'= H(0||TPM_CC_PolicyAuthorise||H(Pk1'))

Policy OR (<V12, V12'>)
V21=H(0||TPM_CC_PolicyOR||H<V12, V12'>)
Policy PCR (1,mOS)
V22=H(V21||TPM_CC_PolicyPCR||H(1,mOS))
Policy Command Code (CC_RSA_Decrypt)
V23=H(V22||TPM_CC_PolicyCommandCode||CC_RSA_Decrypt)