Review – Project: TPM

October 10, 2017

Grading criterion; assignment 1 (5p): *Report describes the necessary steps to repeat the group's experimental setup.*

The reproducibility is high, most steps are explained in detail. **Points: 5/5**

Grading criterion; generating EK (2p): *Report should contain a dump EK and description how it was obtained.*

Contains both. However, there is a byte sequence in the beginning of the dump that does not change between different instances of TPMs. This leads one to belive that this sequence is not part of the EK, but some sort of preamble or overhead. The byte sequence follows here:

```
00 C4 00 00 01 3A 00 00 00 00 00 00 00 01 00 03
00 01 00 00 00 0C 00 00 08 00 00 00 00 02 00 00
00 00 00 00 01 00
```

It would have been more correct if the actual key part would have been marked in the dump. **Points: 1/2**

Grading criterion; dump SRK public key (2p): *Report should contain a dump SRK pubkey and description how it was obtained.*

Contains both. **Points: 2/2**

Grading criterion; Key hierarchy questions (6p): *Each correct answer to the questions equals 2 points.*

Q1: Difference between identity and signature keys?
A: An identity key must meet minimum security requirements and needs the SRK as its parent while the signature key does not need any specific security and can be created further down the key hierarchy. ✓
Note: Correct. `http://opensecuritytraining.info/IntroToTrustedComputing_files/Day1-7-tpm-keys.pdf` , slide 8.

Q2: Which keys can be used for file encryption?
A: All keys can theoretically be used for file encryption but, since we do not want to compromise our key hierarchy and provide adequate security, the best key to use is a storage key whose main purpose is to encrypt the file contents. **X**
Note: It is correct that all of the key types could theoretically in isolation be used for encryption, but in a TPM system that is not allowed; the documentation on the TPM places restrictions on using certain key types for certain operations, as well as almost full restriction on EK. TPM key types (L4 – 59), and:

> "The EK is only available for two operations: establishing the TPM Owner and establishing Attestation Identity Key (AIK) values and credentials. There is a prohibition on the use of the EK for any other operation." – (TPM Part1, page 58)

Correct that a storage key is best, but motivation is not complete or correct.

Q3: There is one type of key that exists, but is not recommended. Which key is that, and why does it exist?
A: Legacy key, which can be used both for signing and encryption although not recommended. They have lower security in order to provide backwards compatibility with older standards. ✓
Note: Combination of ”There was a TPM version 1.0 but we can today forget about that version” (L4 – 53) and ”Legacy: [...] (compatible with TPM v1)” (L4 – 59). **Points: 4/6**

Grading criterion; Key hierarchy – Possible combinations (2p):

Q: Are all combinations possible? if not, why?
A: Note that the creation of key C failed since all children of a migratable key also needs to be migratable. ✓
Note: Creating non-migratable keys with migratable parent as base should return `TPM_INVALID_KEYUSAGE` (TPM-part3 – p.74). **Points: 2/2**

Grading criterion; Key hierarchy – Correct drawing (2p): *Drawing / representation of correct hierarchy.*

Representation is almost correct, but includes the C - key. **Points: 1/2**

Grading criterion; Key migration - questions (10p): *Each correct answer gives 2 points.*

Q1: Is it possible for a migratable key to be the parent of a non-migratable key?
A: No, since when a migratable parent key is exported, all its children needs to be able to tag along. Having the child non-migratable would then cause a contradiction. ✓
Note: Creating non-migratable keys with migratable parent as base should return `TPM_INVALID_KEYUSAGE` (TPM-part3 – p.74).

Q2: Which command is the first to be executed when performing a key migration?
A: On the TPM you are about to import into [..] you should create a new migrateable storage key that can then be used with TPM1 to encrypt keys you want to export with the command `TPM_CreateMigrationBlob`. X
Note: Not correct according to description (TPM-part3 – p.85):

> The TPM Owner does the selection and authorization of migration public keys at any time prior to the execution of `TPM_CreateMigrationBlob` by performing the `TPM_AuthorizeMigrationKey` command.

According to this `TPM_CreateMigrationBlob` is the second command to be executed, after `TPM_AuthorizeMigrationKey`. This is also supported by `https:/¬/shazkhan.files.wordpress.com/2010/10/http__www-trust-rub-de_media¬_ei_lehrmaterialien_trusted-computing_keyreplication_.pdf`, page 4.

Q3: Give a short description of the command `TPM_ConvertMigrationBlob`
A: This command takes a migration blob and decrypts it to a normal wrapped blob which is then possible load into the TPM using the `TPM_LoadKey` function. Note that the command migrates private keys only. The migration of the associated public keys is not specified by TPM because they are not security sensitive. X
Note: One point deduction due to `TPM_loadKey` being deprecated and `TPM_loadKey2` should be used instead (TPM-part3 – p.318).

Q4: Which TPM command load the migrated keys into the TPM?
A: After the migration blob has been converted, the command to load the wrapped key into the TPM is `TPM_LoadKey`. X
Note: Same as above, `TPM_loadKey` is deprecated, should use `TPM_loadKey2` instead.

Q5: Is it the TPM or the TSS that handles the transfer of the migration blob?
A: It is the TSS, since TPM:s have no ability to communicate directly with

another TPM. ✓

Note: Correct. Since the TPM only handles conversion of input and output data, the transfer of the resulting/necessary data is handled by the software stack (TSS). **Points: 7/10**

Grading criterion; Key migration – migration & documentation (2p): *Do the key migration specified in the project instructions and document it (Q1).*

Good. **Points: 2/2**

Grading criterion; Key migration – remaining questions (4p):

Q2: When do you use a key of type `TPM_KEY_USAGE = TPM_Migrate`?
A: `TPM_KEY_USAGE` can have the value `TPM_KEY_MIGRATE`, which is used when there is a need for a migration authority (we did not find any `TPM_migrate` command). ✗
Note: The `TPM_KEY_USAGE = TPM_Migrate` is used to restrict a specific key in such a way that it can only be used in the `TPM_MigrateKey` function. Since this function performs the function of a migration authority with limited knowledge about the key, the physical security of the executing system is assumed to be high (TPM-part3 p.93).

Q3: What is the rewrap option of the `migrate` command used for?
A: This is used when migration authority is needed. ✗
Note: Not correct. It allows a key to be directly moved to another parent, either in the same or another TPM. The re-wrap flag tells the TPM to re-wrap (decrypt→encrypt) the key with a new parent, which enables that parent to load the key as a normal encrypted element (TPM-part3 p.85). **Points: 1/4**

Grading criterion; Extending values to the PCRs – Questions (4p):

Q1: Describe one TPM command that can be used to extend the SHA-1 digest to a PCR.
A: `TPM_SHA1CompleteExtend` “This capability terminates a pending SHA-1 calculation and EXTENDS the result into a Platform Configuration Register using a SHA-1 hash process.” ✓
Note: Correct (TPM-part3 p.160).

Q2: Describe one TPM command that can be used to read a PCR value.
A: `TPM_PCRRead` “The `TPM_PCRRead` operation provides non-cryptographic reporting of the contents of a named PCR.” ✓
Note: Correct (TPM-part3 p.162) **Points: 4/4**

Grading criterion; Extending the PCRs – PCR ’overflow’ (2p): *Run* `sha -if <filename> -ix <PCR index>` *on a large file and show the result.*

The command was run on the `tpmbios` file and adequate prints have been included in the report. **Points: 2/2**

Grading criterion; File encryption – Questions (6p):

Q1: Why is `TSS_Bind` a TSS command, and not a TPM command?
A: The public part of the binding key pair is used for encrypting data. This should be possible to do from anywhere, so the need to load the private key is unnecessary. Once the data is bound, only the one with control over the private part should be able to decrypt it, which is why the private key needs to be loaded for decryption to be possible. This only requires the unbind operation to be a TPM command. ✓
Note: Correct (L5 - p.11).

Q2: Give some difference between data binding and data sealing.
A:

- "Binding": we can encrypt data on another computer, and decryption can only be done on the computer which has the TPM with the private key.
- "Sealing": binds data to a certain value of the PCR and a key that is not migratable. Then only this specific TPM can decrypt (unseal) the data, and even then only if the PCR value is the same as when encryption happened (sealing).

✓
Note: Correct.

Q3: Can a key used for data sealing be migrated to another TPM?
A: No since, the key used for the sealing is required to be non-migratable. ✓
Note: Correct. **Points: 6/6**

Grading criterion; TPM – Data binding (4p):

Q1: Why does the key have to be loaded inside the TPM when decrypting, but not when encrypting?
A: Since we not only need secure keys but also a secure platform in order to correctly decrypt the file, i.e. not possible without a authorized TPM. This is implemented in parts by requiring the private key to be descended from SRK which in turn is related to the unique secret of the TPM chip, the decryption needs parts of the secret only known to the chip manufacturer. [..] The key is also encrypted via the need for password previously set by the TPM. ✓
Note: Not a very clear description, but correct.

Q2: Migrate the binding key to TPM2 and see if the file can be decrypted there.
A: ✓
Note: Should be possible, and it worked. **Points: 4/4**

Grading criterion; TPM – Data sealing (5p):

Q1: Test if you can do a sealing with a legacy key, a binding key or a signing key. If not, why?
A: As can be seen, it is only the non-migratable storage key that can be used for sealing. [..] The key needs to be non-migratable in order to be used by the

sealfile command. It was however possible to use migratable keys when using bindfile command, since with bind we only encrypt with the key itself. Contrast this with sealing were we also bind the encryption with the PCR values of the TPM. Therefore it is impossible to seal a file with migratable keys as they would be useless for decryption since some PCR values can't be extracted out of the TPM. ✓

Note:

> If the keyUsage field of the key indicated by the keyHandle does not have the value `TPM_KEY_STORAGE` the TPM must return the error code `TPM_INVALID_KEYUSAGE`.
>
> – Documentation of `TPM_Seal` (TPM-part3 p.63)

Q2: Now migrate the storage key to TPM2 and see if you can unseal the file there too. Explain what you observe.
A:

✓

Note: An attempt was made, but did not work, it is described why ("As can be seen, it is only the non-migratable storage key that can be used for sealing.")

> If the keyHandle points to a migratable key then the TPM MUST return the error code `TPM_INVALID_KEYUSAGE`.
>
> – Documentation of `TPM_Seal` (TPM-part3 p.63)

Points: 5/5

Grading criterion; TPM – Authentication (6p):

Q1: In the above, could the `verifyfile` command been done by another TPM?
A: Yes, since the signing is done by the private key, the public part can be used for verifying the private signing. And since that part is public, any TPM can use it. ✓
Note: The usage string for verifyfile agrees with only using public part: `verifyfile [-ss info|der] -is <sig file> -if <data file> -ik <pubkey file (.pem)>`.

Q2: Which TPM command is used to decrypt the file?
A: “`TPM_UnBind` takes the data blob that is the result of a `Tspi_Data_Bind` command and decrypts it for export to the User." [..] The binding operation is undone by unbind. This requires the private part of the keypair and is done inside the TPM. ✓
Note: Same decryption command as earlier.

Q3: Can the decryption based authentication be done by using data sealing instead of binding?
A: Not really. Sealing limits the encryption and decryption to only one specific TPM. No other TPM should be able to encrypt or decrypt it which means the only external party that the TPM can authenticate towards is itself, which is kind of moot. ✓
Note:

From `http://opensecuritytraining.info/IntroToTrustedComputing_files¬/Day2-1-auth-and-att.pdf` one can see that "If you want.. `Decryption-based Authentication`, Use key type.. `Binding` with.. `Bind`" (paraphrase of table on Summary page). **Points: 6/6**

Grading criterion; Signing (2p): *Use* `signfile` *and* `verifyfile` *to sign and verify a text file.*

Presented commands are correct. **Points: 2/2**

Grading criterion; Encryption (2p): *Use* `bindfile` *and* `unbindfile` *to encrypt and decrypt a text file.*

Presented commands are correct. **Points: 2/2**

Grading criterion; Attestation – signature (2p): *Use the* `identity` *and* `quote` *commands to create an AIK and use that to quote a PCR value.*

Provided commands are correct. **Points: 2/2**

Grading criterion; Attestation – decryption (3p): *Use the* `createkey -ix`*,* `sealfile` *and* `unsealfile` *commands to bind a hash-digest for a file to a storage key. Unseal the original file, which should work. Modify the file and extend the PCR with the new file, the unsealing should fail.*

The correct commands are provided. **Points: 3/3**

Grading criterion; Creating a TPM program (4p): *Provide correctly working program with source code and documentation to repeat the work.*

Program is included, should work. **Points: 4/4**

Total: 65/75

One could see that several parts of the report were rewritten parts from the project description. On the other hand, this together with some augmenting explanations, clear commands and printouts made for a very readable report that left few questions as to reproducibility. Good job.