

Review – Project: TPM

October 10, 2017

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

The description is a bit terse, would probably suffice for anyone already comfortable with virtual machines, Linux and the TPM. **Points: 3/5**

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 believe 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 01 00 03
00 01 00 00 00 0C 00 00 08 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.*

Nothing about the SRK public key in the report. **Points: 0/2**

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

Q1: Difference between identity and signature keys?

A: One of the differences between the two is that the identity key is used for attestation while the signature key is used to sign user data. ✓

Note: "Signing keys only for signing" (L4 – 59), Attestation identity key involved in remote attestation (L5 – 23).

Q2: Which keys can be used for file encryption?

A: Legacy keys and storage keys can be used for file encryption. ✓

Note: TPM key types (L4 – 59).

Q2: There is one type of key that exists, but is not recommended. Which key is that, and why does it exist?

A: Legacy keys. The legacy keys exist because of the fact that they are compatible with older system. ✓

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: 6/6**

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

Q: Are all combinations possible? if not, why?

A: The key named C could not be created. The reason for this is that a non-migratable key can not be a child of a migratable key. ✓

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 correct, does not include the C - key.

Points: 2/2

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, it is not possible for a migratable key to be the parent of non-migratable keys. When a key is migrated, all of its children are migrated as well. Thus, all children of a migratable key have to be migratable ✓

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: `TPM_CreateMigrationBlob` is the command that implements the first step in the process of moving a key to a new parent or platform. ✗

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_AuthorizeMigrationKey` is the second command to be executed, after `TPM_CreateMigrationBlob`.

Q3: Give a short description of the command `TPM_ConvertMigrationBlob`

A: This command takes migration blob and creates a normal wrapped blob. The migrate blob must be loaded into the TPM using the normal `TPM_loadKey` function. ✗

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: `TPM_loadKey` ✗

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: The TSS handles the transfer of the migration blob. ✓

Note: 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). A more motivated answer wouldn't hurt. **Points: 8/10**

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

Commands are cut off, seems to be ok but can't be verified. **Points: 1/2**

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

Q2: When do you use a key of type `TPM_KEY_USAGE = TPM_Migrate`?

A: When using migration authority. **X**

Note: While the answer is 'correct' it is a bit short.

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 option of the `migrate` command used for?

A: The rewrap option is use to directly move a key to a new parent. **X**

Note: Missing explanation on what enables the key to be 'directly moved' to another parent. The 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: 2/4**

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: The `TPM_Extend` command can be used to extend the SHA-1 digest to a PCR by adding a new measurement to a PCR. **✓**

Note: Correct (TPM-part3 p.160).

Q2: Describe one TPM command that can be used to read a PCR value.

A: The `TPM_PCRread` command can be used to read a PCR value. **✓**

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 figure shows that the command has been run, but since it is run on `text.txt` and there is no indication of then file size of the file it's hard to figure out if the run illustrated the functionality or not. **Points: 1/2**

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

Q1: Why is `TSS_Bind` a TSS command, and not a TPM command?

A: The encryption is done outside of the TPM and is therefor not a TPM command. **✓**

Note: Correct (L5 - p.11).

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

A: Data binding uses a symmetric key for encryption and data sealing uses asymmetric encryption. **X**

Note: Could not verify that this is true, and since sealing is binding but where

the data is locked to a given PCR state, it should be the same type of encryption?

Q3: Can a key used for data sealing be migrated to another TPM?

A: No, because sealing data to one TPM platform makes it illegal to migrate a key to another TPM. X

Note: Deducting one point because the answer makes it sound like no migration is possible after sealing any data on a TPM. The key used to seal the data can not be migrated (L5 – p.2). **Points: 3/6**

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: When encrypting, only the public key is needed, while decryption uses the private key, which is not accessible outside of the TPM. ✓

Note: This makes sense, since this is how asymmetric cryptography works.

Q2: Migrate the binding key to TPM2 and see if the file can be decrypted there.

A: To be able to decrypt the binding key is needed. Since the binding key was migrated to TPM2, the file was able to be decrypted there too. ✓

Note: Should be possible.

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: It was not possible to do a sealing with a legacy key, binding key or signing-key. The reason for this is that sealing only can be done with storage keys. ✓

Note:

If the keyUsage field of the key indicated by the keyHandle dose 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: The storage key is not migratable and could therefor not be migrated to TPM2. Since TPM2 did not have access to the storage key, unsealing was not achieved. ✓

Note:

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 it is possible because only the public key is used. ✓

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: The command that is used to decrypt files is TPM_UnBind. 

Note: Same decryption command as earlier.

Q3: Can the decryption based authentication be done by using data sealing instead of binding?

A: No it can not because the TPMs have different PCR values. 

Note: Sealing can work to identify a TPM. If a user encrypts a piece of data with the public part of the key, only the TPM that has the correct private part and is in the correct measured configuration state can decrypt it and return the data. Which in turn proves that it's the same TPM, with the same configuration.

Points: 4/6

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 `bindefile` and `unbindefile` 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.

Only a small snippet provided, which should work. Should instruct where rest of the code is.

Points: 2/4

Total: 52/75