Security and Trusted Hardware Applications
Week #6 Tutorial

Pedro Antunes - up201507254

May 30, 2022

Instructor: Bernardo Portela

1

One of the security features of a TPM is the secure generation, storage and management of cryptographic keys. These keys that are used to perform trusted cryptographic processing. A TPM uses its own internal firmware and logic circuits to process instructions. Therefore, it is not dependent on the operating system and is not exposed to vulnerabilities that may exist in the operating system or application software.

If a TPM fails to provide secure storage of the cryptographic keys, then an adversary residing on the system outside the TPM can retrieve the stored keys, by breaking the harware bus through processes or other components or even in physical access, and thus compromise the information that has been subject to cryptographic operations from the stored keys. As an example, without the hardware bus enforced by a TPM, an attacker can decrypt encrypted messages, or also, can decrypt keys that derive from the key stored in a TPM.

2

The PKCS#11 standard specifies the API Cryptoki(Cryptographic token interface) and it defines a platform-independent to cryptographic tokens, such as HSMs and smart cards. The API defines most commonly used cryptographic object types(RSA keys, X.509 certificates, AES keys) and all the functions needed to operate those objects, such as generate, modify and delete.

As written above, the benefits of taking this approach are the object-based approach, technology independence, resource sharing. These features of a device implementing PKCS#11 lead to the details of cryptographic operations being isolated from an application. It also adds portability, since if an application modifies one HSM for another, the application does not suffer because it interacts with this standard API.

3

a) The trusted hardware that is compatible with this system functionalities is a smart card.

b) The technologies we have studied wouldn't fit this problem because they don't provide portability by nature (they are more aimed at performing secure processes from a central entity in the system) and they have more computational power and functions, which makes such technologies more expensive, that weren't required on problem specification.

4

a) The HSM implementation provided in this exercise is not correct and is it not defined as a possible implementation intended for trusted hardware. This is because it would allow the keys generated in the context of HSM to be extracted and exposed outside the hardware.

   What this hardware is intended to do is precisely a secure and internal storage to store the keys. Or else, if this is not the case, extracted to the hardware's external system but always encrypted with a key that is stored inside the trusted hardware, so that we can only decrypt and retrieve the key inside the hardware.

b) To fix this problem we need to add a dictionary data structure to handler the generated keys and return the respective handler identification key to outside context. The fixed code is sent alongside with this report.