# Security and Trusted Hardware Applications
# Week #10 Tutorial

Pedro Antunes - up201507254

May 30, 2022



Instructor: Bernardo Portela

# 1

In this scenario, we have an application with clients and aimed to securely archive informations on server-side. So, the goal of an attacker is to retrieve any client information. If we add trusted hardware technologies to this system, the goal of an attacker still the same. Therefore, we are dealing with more resourceful adversial power. This is because, if an attacker has trusted hardware control, he have impersonate power over the hardware and he may use the attack vectors in this example to defeat the trusted anchor-based system.

# 2

When we are evaluating the risk of an attack, we must care about physical attacks targeting the chosen trusted hardware. In a passive way, the attacker can read out data from BUS or individual with a auxiliary thin needle. In a active way, the attacker can do circuits modification like connect or disconnect security mechanism, or with a auxiliary laser cut or paste circuit tracks. He also can read clock and temperature values that with combination of side-channels attacks it can be very dangerous.

So we have to make sure that hardware is tamper-resistance by these physical attack vectors. Otherwise, an adversary that can gain physical access to secure hardware and manipulate it, can combine physical attacks with other types of attacks, and thus easily break the role of the trusted anchor hardware's.

# 3

Performing constant time execution guarantees that no matter what control flow did taken by the instance code, the program execution will take the same time on thoses. Even when we already had computed the algorithm response. Looking for complexity time, our constant time algorithm will always aimed to the worst-case, which can be expensive in some cases. So we need to evaluate whether constant time implementation it's worth it.

Implementing this, can prevent some physical time measurements for example. However, other time measurements can be done by other type of atacks. Speculative attacks will perform code execution without knowing some control flow variable. So, in normal situation of control flow, we have 2 possible paths. With speculative execution, the cpu will execute both.

Even discarding the work made by wrong path, this will have impact on other components like caches. The attacker can also try timing attacks knowing the values cached. AND HAVE NO REAL MITIGATIONS TO THIS ATTACK!!!

# 4

a) In this situation, the HSM guarantees the security of the application's client keys, which guarantee the privacy of the files stored on the server. If the attacker hacks the server, the stored files from the clients are only retrieved with the key that is inside the HSM. So even if the server is hacked, the attacker will not be able to recover the stored files.

   However in this situation, the attacker can destroy these files and if there is no backup, the files are lost, because HSMs don't have the computation power to doing backups.

b) If we consider an adversary that can instead gain control of the server and the HSM, he can easily retrieve the client information by ask for hsm to decrypt some file with client key (or server key if the key for all clients is the same).