National Taiwan Normal University
CSIE Information Security

Instructor: Po-Wen Chi
Due Date: May 10, 2021, PM 11:59

Assignment 4

Policies:

- Zero tolerance for late submission.
- Please pack all your submissions in one zip file. **RAR is not allowed!!**
- I only accept **PDF**. MS Word is not allowed.
- Hand-writing is not allowed.
- Please use **Chinese.**

4.1 Problem Equivalent (20 pts)

Consider a specific cyclic group $\mathbb{G}$ of prime order q generated by $g \in \mathbb{G}$. Show that the following problems are deterministic poly-time equivalent:

- Given g^α, g^β, compute $g^{\alpha\beta}$.
- Given g^α, compute g^{α^2}.
- Given g^α and $\alpha \neq 0$, compute $g^{\frac{1}{\alpha}}$.
- Given g^α, g^β with $\beta \neq 0$, compute $g^{\frac{\alpha}{\beta}}$.

Note that all problem instances are defined with respect to the same group $\mathbb{G}$ and generator $g \in \mathbb{G}$.

4.2 ElGamal Threshold Decryption (20 pts)

Ideally, during decryption, the secret key sk is never reconstituted in a single location. This ensures that there is no single point of failure that an adversary can attack to steal the key. In such a system, there are s key servers, and an additional entity called a combiner that orchestrates the decryption process. The combiner takes as input a ciphertext c to decrypt, and forwards c to all the key servers. Every online server applies its key share to c, and sends back a *partial decryption*. Once t responses are

received from the key servers, the combiner can construct the complete decryption of c. Overall, the system should decrypt c without reconstituting the key sk in a single location. Such a system is said to support **threshold decryption**.

The formal definition is as follows:

Definition 4.1 (Public-key Threshold Decryption)**.** A public-key threshold decryption scheme $\mathcal{E} = (G, E, D, C)$ is a tuple of four efficient algorithms:

- $G(s, t) \rightarrow (pk, sk_1, sk_2, \ldots, sk_s)$: This is a probabilistic algorithm that outputs a public key pk and s shares of the private keys.
- $E(pk, m) \rightarrow c$: This is an encryption algorithm as in a public key encryption scheme.
- $D(c, sk_i) \rightarrow c'_i$: This is as deterministic algorithm that outputs a partial decryption of c using sk_i.
- $C(c, c'_1, \ldots, c'_t) \rightarrow m$: Given t partial decryptions of c, this algorithm outputs the original message m.

Now please design two threshold decryption schemes based on the ElGamal encryption scheme:

1. 2-out-of-2 (5 pts).
2. 2-out-of-t (15 pts).
    - Hint: you can google **Shamir secret sharing**.

Note that you need to show that each key server learns nothing with its partial private key.

4.3 Ettercap: MITM (15 pts)

Ettercap is a comprehensive suite for man in the middle attacks. Please use this tool to launch a man in the middle attack. You need to capture your neighbor's computer's login to NTNU CSIE webmail and get his/her password. You need to write a report about how you use this tool. Undoubtedly, in Chinese.

For convenience, you can use Kali Linux since Kali linux has this tool suite.

4.4 Reading Assignment: Dependency Confusion (15 pts)

Please read the following article.

https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610

Its title sounds amazing, right? Please write down how it works in **Chinese** and propose a defense solution.

4.5 Webgoat (15 pts)

In this class, I have shown you what **webgoat is**. However, we do not have enough time to play all exercises of webgoat. So I give you this homework to force you to play this game

Please solve the problems listed in **Access Control Flaws** → **Insecure Direct Object References**. Write down how you solve this problems in detail, including your steps and screenshots, and in **Chinese**. Note that I know there are lots of write-ups, but I need you to practice them yourself. I will ask the TA to check if you are a copycat.

4.6 Lab: RSA (15 pts)

In this class, I have told you that you should not use MD5 anymore. Let's see how to make a MD5 collision pair for two executable binaries.

- Lab: https://seedsecuritylabs.org/Labs_20.04/Files/Crypto_RSA/Crypto_RSA.pdf

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising. Please also list the important code snippets followed by explanation. Simply attaching code without any explanation will not receive credits.