HW3

Q1. What is the most simplest things/strategies you can do or apply in the encryption algorithms to prevent Ciphertext only, Known-Plaintext, and Chosen-plaintext attacks?

In result, the goal of these attacks is to get key for decryption. So the simplest way to prevent that attacks is changing key periodically. Then, even though attacker gets key, after changing key, the old key becomes useless.

Q2. How does Telegram provide the end-to-end encryption?

There are two major encryption methods, symmetric key and asymmetric key method. Each method has pros and cons. Symmetric key method have the advantage of being very fast in encryption/decryption but there is a problem that key has to be exchanged. If sender and receiver can't physically exchange keys directly, during transmission, attacker can steal the keys. For solving this problem, asymmetric key method uses two distinct keys for encryption/dycrption. but this way is very slow. In telegram, they use both ways to provide the end-to-end encryption. First, sender and receiver create their own public key and private key, and share the public keys each other. Second, sender encrypts symmetric key to receiver's public key and sends the symmetric key that is encrypted with message. Finally, receiver decrypts the symmetric key and also decrypts message with that key. This is how end-to-end encryption works.

Q3. Find how KakaoChat provide chat message protection. Do they provide end-to-end encryption? What encryption/integrity mechanisms do KakaoChat

use?

Kakao chat provides end-to-end encryption, but it is not default setting. End-to-end encryption is supported only when using secret chat. Kakao chat also uses various security policies as well as encryption.(storing messages just 2~3 days in database)

Q4. Find how WeChat provide chat message protection. Do they provide end-to-end encryption? What encryption/integrity mechanisms do WeChat use?

WeChat don't provide end-to-end encryption, instead they use client-to-server and server-to-client encryption. Especially, the algorithm of encryption used in WeChat is AES-256. It is one of the most popular symmetric key encryption methods.

Q5. Use the following to do the Colab and share your Colab notebook link. You can use the crypto library that python provides.

A. Calculate MD5 hash of an input: "sowon"

   0fb9e29adad33b3d975b8130c75f2d29

B. Calculate MD5 hash of an input: "so won"

   aa49d035e67f3b2e95327e0bee8e3fd1

C. Calculate MD5 hash of an input: your own name

   7a2c0a3b494d5b44bc48c270c7ef87c6

D. Calculate SHA256 hash of an input: "sowon"

301d9673853b23540b5919e1322a38d909608d125a2ee94646351f14144bbcaa

E. Calculate the SHA256 hash of the following files:

https://the.earth.li/~sgtatham/putty/latest/putty-0.76.tar.gz

547cd97a8daa87ef71037fab0773bceb54a8abccb2f825a49ef8eba5e045713f

Colab URL:

https://colab.research.google.com/drive/1j61O8plvr7R69mAuxr-WN3eswp_Hn-bl#scrollTo=bEIvBrrh7cpl

Q6. What's the advantage of having the fixed size of hash output?

For encryption and decryption we have to store the hash values in our data structure, so if the size of hash value is not fixed(variable), it can lead to unnecessary waste of storage space. Also, it has advantage in terms of speed. if the keys' size are fixed, it is really convenient while comparing hash keys.

Q7. Explain why collision-free is impossible to achieve in hash?

The output of hash algorithm has limited size, but the number of input values is infinite. In other words, there are more keys than indexes in hash table. It can be explained by pigeonhole principle. The contents of the pigeonhole principle are as follows. if more than n items(like n+1) are put into n containers, then at least one container must contain more than one item. So all hash algorithms can't avoid from hash-collision.