# Assignment Two (COMP5355)

## Please submit the soft copy to the blackboard by 11:59 PM 2023-06-24

1. Bob wants to publish his public key based on RSA scheme. He selects two prime numbers p = 7 and q = 19, and let e = 23. Answer the following questions related to RSA. (10%)

(1) What will be uploaded by Bob to a public server so that others can use his public key to do encryption?

(2) What is the private key?

**Answer:**

**(1) A public key file including N and e will be uploaded to the public server.**

**N =p*q=7*19=133**

**φ(N) =(p-1)*(q-1)=108**

**e = 23**

**So Bob will upload the public key that is PU = [e,n] = [23, 133]**

**(2) The private key is another key parameter in the RSA encryption algorithm, which is used along with the public key to encrypt and decrypt data.**
**The private key in this case is a file that includes the modulus N and the private key exponent d.**

**N = 133**

**φ(N) =(p-1)*(q-1)=108**

**d = 47, because 47*23= 108*10+1**

**So the private key is PR = [d,n] = [47, 133]**

2. Alice and Bob use the Diffie-Hellman key exchange protocol to establish a common key. Let q=13 and a=9 be public. Alice selects XA=4 and Bob selects XB=5. Compute their common key. (15%).

**Answer:**

**Alice:**
**XA=4, compute public YA=a^(XA) mod q=9^4 mod 13=9**
**Bob:**
**XB=5, compute public YB=a^(XB) mod q=9^5 mod 13=3**
**Exchange YA and YB**

**Alice:**
**Calculate the secrete key K=(YB)^(XA) mod q=3^4 mod 13=3**
**Bob:**
**Calculate the secrete key K=(YA)^(XB) mod q=9^5 mod 13=3**
**So their common key K=3.**

3. Answer the following questions that are related to cryptography? (15%)

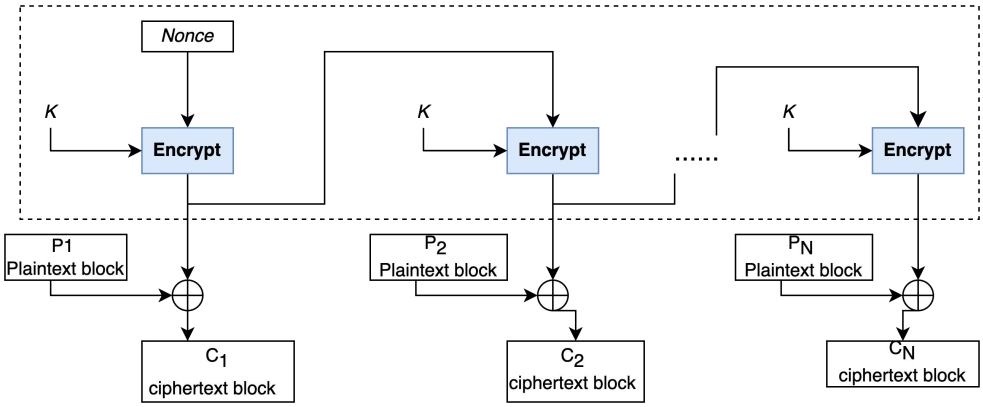
(1) How to achieve unconditionally secure? Is such approach practical? Why?

(2) Draw the encryption and the decryption structures of Output feedback (OFB).

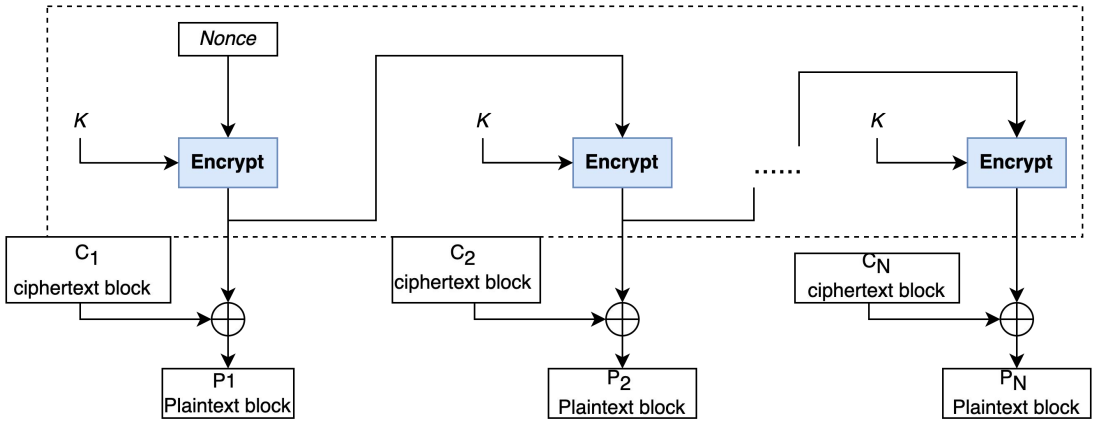
**Answer:**

**(1)**
**1. Unconditionally secure:**
**The ciphertext does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available.No matter how much time an opponent has, he/she cannot decrypt the ciphertext.**
**Only one-time pad is unconditionally secure.This method of encryption uses a random number generator to generate a random key of the same length as the message, which is then heterogeneous with the plaintext to generate the ciphertext. As each key bit can only be used once, there is not enough information available to break the ciphertext.**
**2. Is such approach practical:**
**There are significant practical limitations to this method.**
**3. Why:**
**There are two main reasons for this:**
**It is difficult to produce large numbers of random keys.**
**Difficult to distribute and protect keys.**

**(2)**
**Encryption structure:**




**Decryption structure:**




4. Answer the following questions related to IPSec. (15%)

(1) Describe how IPSec defends against the replay attacks.

(2) What are the differences between the Authentication Header (AH) and the Encapsulating Security Payload (ESP)?

**Answer:**

**(1)**
**There are two main ways to do this:**
**1. Using Sequence Number.**
**If the anti-replay service is enabled, the sequence number of the packet is checked to make sure that it is new and falls within the anti-replay window.**
**2. Anti-Replay Window.**
**Used to determine whether an inbound AH or ESP packet is a replay.**

**(2)**
**For AH:**
**1. AH provides datagram-origin authentication, data integrity, and protection from replay attacks. The main focus is on the integrity check of the data and source address authentication.**
**2. Compute a cryptographic MAC (integrity check value, ICV) over parts of the IP header and the entire payload data.**
**3. AH cannot authenticate the entire IP headers.**
**For ESP:**
**1. It provides confidentiality, authentication, integrity, anti-replay services and (limited) traffic confidentiality.The main function is to encrypt and authenticate data.**
**2. The entire IP datagram is encrypted by the ESP packet.**

5. Answer the following questions that are related to TCP security. (15%)
(1) What are the differences between in-path TCP attacker and the off-path TCP attacker?
(2) Describe two types of attacks that can be launched by a malicious TCP receiver.
(3) Describe the process of ACK storm attack and its effect on the network.

**Answer:**

**(1)The difference between an in-path TCP attacker and an off-path TCP attacker is where they are located. An in-path TCP attacker is an attacker who has access to the path of a TCP connection, for example between two connected parties, whereas an off-path TCP attacker is an attacker who does not have direct access to the path of a TCP connection, for example by attacking one of the connected parties through network topology or IP address forgery.**

**(2)There are two types of attack methods that a malicious TCP receiver may employ:**
**1. The first method is to force the sender to send more data to the network by establishing multiple connections or bypassing congestion control mechanisms. An attacker could implement various techniques such as manipulating the timing of ACK packet delivery or misleading the TCP sender to quickly increase its congestion window.**
**2. The second method of attack is to use fingerprinting techniques to infer the operating system used by the remote host and launch an attack against it, including an ACK storm attack. This type of attack is an in-path attack where the attacker will tap into the network and will pretend to be the IP address of each end, sending false ACK packets to each end of that TCP connection, thus causing packets from the real end to be discarded. Eventually, both ends will continue to send ACK packets to the other end, but all of these ACK packets will contain incorrect acknowledgement values, causing all ACK packets to be dropped.**

**(3)**
**The process of ACK Storm Attack:**
**1. In-path attacker first eavesdrops the traffic and sends two packets to either end of the connection.**
**2. For each end, the packets from real end will be dropped, because the real ACK packets acknowledge some data that has not been sent yet.**
**3. The attacker continues to send more requests, and both ends continue to send ACK packets to the other end, resulting in a large number of ACK packets being generated at both ends. But all ACK packets are discarded.**
**Effect:**
**It potentially cause network congestion and degradation of quality of service. Because the victim host is in a constant state of consuming network bandwidth and processing power. This may cause network congestion and delays, disrupting normal network traffic transmission and making it impossible for users to access the victim's server or website.**

# Part 2. Questions from Research Papers

1. Read the paper [1] and the reference therein, and then answer the following questions. (15%)
(1) Enumerate the common cryptography misuse problems?
(2) How does the usability of cryptography APIs affect the software security?
(3) How should cryptography APIs be designed to help developers avoid insecure code?

**Answer:**
**(1)The cases illustrated in this thesis focus on weak ciphers and insufficient randomness, two common cryptographic misuse problems.**
**In fact common cryptographic misuse problems include: weak encryption algorithms, incorrect encryption key lengths, incorrect encryption patterns, static IVs, lack of key derivation functions, custom key derivation functions, lack of KDF salts, lack of KDF algorithms, lack of KDF iterations, certificate verification errors, host name verification errors, date verification errors and signature checking errors.**
**(2)In this thesis the researchers conducted experiments with 256 Python developers by having them use five different APIs for typical encryption tasks. This experiment allowed for the elaboration of the relationship between the usability of the cryptography api and software security.**
**While APIs designed for simplicity can yield security benefits, poor documentation, lack of code examples and lack of secure key storage caused participants to experience problems with basic functional correctness and security. Comprehensive**

**documentation and the availability of easy-to-use code examples seem to compensate for more complex APIs in terms of functional correctness and participant satisfaction, but this does not apply to security outcomes. Libraries that want to promote effective security should provide a simple and convenient interface, but should also ensure that a wide range of common tasks are supported and that secure and easy-to-use documentation and code examples are available.**

**(3)**

**1. According to the research in the First Conference on Finding Encryption APIs that Work, new encryption libraries should provide a simple, convenient interface, ensure support for common tasks and provide easy-to-use code examples to ensure correct functionality and participant satisfaction.**

**2. At the same time, the API should support a range of use cases and should not force developers to learn and use new APIs that are intensively relevant.**

**3. In addition to easy-to-use design, useful documentation needs to be designed in a way that draws developers' attention to the need to provide safe, easy-to-use code examples to help them learn to use it quickly, thereby avoiding the introduction of potential security risks.**

**4. It is worth noting that for approximately 20% of tasks that function correctly, participants believe that their code is safe, but in reality it is not. Therefore, to prevent the introduction of security risks, communication between developers and security experts also needs to be enhanced.**

2. Read the article [2] and the reference therein, and then answer the following questions. (15%)

(1) What are the common security problems in password storage?

(2) Why do these security problems occur frequently?

(3) How to avoid such security problems?

**Answer:**

**(1)**

**1. There are often security vulnerabilities in password storage, with developers often storing passwords in plain text - such as in plain text as mentioned in the papers [20, 23, 38, 78].**

**2. used unsafe hashing functions.**

**3. static salts.**

**(2)**

**1. According to the introduction section of the first part of the thesis, we know for a fact that security problems often occur when developers store passwords because the complexity of the task causes developers to fail to implement security measures, even though they have understood the security practices.**

**2. According to the section on disscussion in part 6, it is known that developers often ignore security and some developers do not feel responsible and do not focus on implementing security when it is not explicitly mentioned.**

**(3)From the first and sixth part of the paper and the seventh part we can understand that some options make some extensions according to the references:**
**1. firstly, easier-to-use APIs are not sufficient to guarantee security, which needs to be explicitly required;**
**2. secondly, between functionality and security, functionality should be prioritised;**
**3. thirdly, continuous learning is essential to avoid security problems; and finally, qualitative research is needed to gain insight into developers' misconceptions and lack of knowledge;**
**4. researchers are encouraged to generalise their findings to tutorials, standardisation bodies and frameworks, which will help add security to practical development**
**5. default security settings should be provided as part of the solution to prevent developers from making mistakes when storing passwords;**

# References:

[1] Comparing the Usability of Cryptographic APIs, IEEE S&P, 2017. https://www.cl.cam.ac.uk/~rja14/shb17/fahl.pdf
[2] Why Do Developers Get Password Storage Wrong?: A Qualitative Usability Study, CCS 2017. https://arxiv.org/pdf/1708.08759