NATIONAL UNIVERSITY OF SINGAPORE

# CS2107 – Introduction to Information Security

(AY2018/9 Semester 1)

## Mid-Term Quiz

Date: 1 Oct 2018 Time: 2:15 - 3:30PM

---

**STUDENT NUMBER :** | A | | | | | | | | |

**NAME :**

**TUTORIAL GROUP :** **DAY:** **TIME:**

(Write your Name and Student Number legibly with **a pen** to prevent accidental erasure.)

## INSTRUCTIONS TO CANDIDATES

1. This question paper consists of **NINETEEN (19)** questions in **THREE (3)** parts; and comprises **NINE (9)** printed pages, including this page.
2. Fill in your Student Number, Name, and Tutorial Group information above with a pen.
3. This mid-term quiz has **30 marks**, and is worth **15%** of your final mark.
4. Answer **ALL** questions.
5. You may use pen or pencil to write your answers, but please erase cleanly, and write legibly. Marks may be deducted for illegible handwriting.
6. Write your answers on this **question paper**.
7. This is an **OPEN BOOK** assessment.
8. You are allowed to use **NUS APPROVED CALCULATORS**. Yet, you should be able to work out the answers without using a calculator.

# Part A (5 marks): Multiple Choice Questions

***Instructions***: Choose the ***best answer***, and circle/cross the corresponding letter choice below. No mark is deducted for wrong answers.

**A1.** Alice needs to ensure confidentiality with a high diffusion level. Which cryptographic technique should she use?

**a) Block cipher**
b) Stream cipher
c) Hash
d) MAC
e) Digital signature

**A2**. Bob wants to ensure the integrity of his messages sent to Charlie in the presence of active attackers. A secure channel between Bob and Charlie is, however, *not* available. Yet, Bob and Charlie share a secret key, and want to use this key to achieve the security requirement. Which cryptographic technique should both use in this case?

a) Block cipher
b) Stream cipher
c) Hash
**d) MAC**
e) Digital signature

**A3**. Bob wants to protect the authenticity of his messages sent to Charlie. Charlie now also requires an assurance that Bob cannot deny his previously-sent messages. Both of them insist on solely using their shared secret key. Which cryptographic technique can be used?

a) Block cipher
b) Stream cipher
c) Hash
d) MAC
**e) None of the above**

**A4.** Which statement regarding classical cipher(s) below is *false*:

a) Substitution cipher is insecure under known-plaintext attack
b) Substitution cipher is insecure under ciphertext-only attack
c) Permutation cipher even with a large block size is still considered insecure
**d) Since one-time-pad cipher failed in the "Venona Story", it is thus considered a broken cipher and must not be used**
e) Modern ciphers, instead of classical ciphers, should be used in general practical use cases in today's computing and Internet age

**A5**. The criminal practice of using social engineering over the (*voice-based*) *telephone system* to gain access to private personal and financial information is specifically known as:

a) Phishing
**b) Vishing**
c) Smishing
d) Pharming
e) Scanning

## Part B (10 marks): Security Terminology

***Instructions***:

The next ten questions (B1 to B10) give security-related descriptions. Below is a list of security terms. Fill in the blanks in the next ten questions with the ***most appropriate*** terms from the list. Put only one choice per blank. You may ignore any grammatical rules on plural forms. Note that it is possible for some choices to appear more than once in your answers in this part.

**Cryptography Objects:**
Block cipher
Stream cipher
Initial Value (IV)
Pseudo random sequence
One-time-pad
Symmetric key
Public key
Private key
Signature
Certificate
Certification Authority
Self-signed certificate
Hash
MAC
Authenticated encryption
Nonce
Mode-of-operation

**Cryptography Notions:**
Symmetric Key Cryptography
Public Key Cryptography
Public Key Infrastructure
Kerckhoffs's principle

**Attacks:**
Denial of Service
Man-in-the-middle
Chosen-plaintext
Known-plaintext
Frequency analysis
Brute-force
Side-channel
Phishing
Skimming
Birthday
Typo squatting

**Miscellaneous:**
2FA
Covert channel
Bring-your-own-device
Botnet
Worm

**B1**. A Certificate Revocation List (CRL) must be signed by the [ Certification Authority ] that previously issued the revoked certificates.

**B2**. A/an [ block cipher ] operates on a fixed-sized block of input, and can provide high diffusion and confusion properties.

**B3**. An attacker registered for the domain name “www.dbsbank.com", and then set up a maliciously-spoofed DBS bank website. The attacker was hoping that some Internet users would visit the website and mistakenly believe that they visit the website of DBS bank. This is an example of a/an [ typo squatting ] attack.

**B4**. Stream ciphers aim to simulate the [ One-time-pad ], which has a perfect secrecy property, since its ciphertext gives absolutely no additional information about the plaintext.

**B5**. One type of [ side-channel (attack) ] is timing attack, which measures how much time various computations (e.g. comparing an attacker's given password with the victim's unknown one) take to perform, without knowing the performed computations.

**B6**. MiFare Crypto 1 is a stream cipher used in London's Oyster card, Netherland's OV-Chipcard, and in numerous wireless access control and ticketing systems world-wide. Researchers were able to recover this algorithm by reverse engineering. The encryption uses a 48-bit key, which could be recovered in seconds on a PC given a known IV (from one single encryption). The card manufacturer failed to apply [ Kerckhoffs's principle ].

**B7**. A different [ Initial Value (IV) ] must be chosen for encrypting a plaintext, and will be sent in clear as part of the generated ciphertext.

**B8**. A/An [ authenticated encryption ] simultaneously provides confidentiality, integrity, and authenticity assurances on the data, by outputting both ciphertext and authentication tag during its encryption process.

**B9**. To encrypt a plaintext longer than its block size, a block cipher needs to employ a good [ mode-of-operation ] such Cipher Block Chaining (CBC), and not a weak one like Electronic Codebook (ECB).

**B10**. When a transfer of information objects between two separate processes is not supposed to be allowed by the applicable computer security policy, a/an [ covert channel ] is sometimes created by an attacker.

## Part C (15 marks): Structured Questions

***Instructions***: Write your answers in the spaces provided.

**C1**. *Usage of multiple cryptographic keys* (4 marks)

a) (2 marks) Bob knows that DES has a rather short key length of 56 bits. He, however, still wants to employ DES due to its widespread availability. Bob thinks that he has found a good way of addressing the limited key length of DES by randomly selecting *three* different keys $K_1$, $K_2$ and $K_3$. Bob then performs his DES encryption as follows:

$$C = E_{K_1 \oplus K_2 \oplus K_3}(P).$$

Decryption process is then performed using $K_1 \oplus K_2 \oplus K_3$ as its key. Bob argues that his method significantly increases the key space size. Is Bob's argument correct? Argue concisely by comparing the key space size of using one and three keys above.

Bob's argument is *incorrect*.

The new key $K_1 \oplus K_2 \oplus K_3$ has the *same length* as those of $K_1$ and $K_2$, namely 56 bits. This is since the XOR operation of three 56-bit strings is also a 56-bit string. Hence, the key space size of Bob's new method *remains* $2^{56}$.

b) (2 marks) Bob now uses only two secret keys $K_1$ and $K_2$. However, he modifies his encryption as follows:

$$C = E_{K_2}(E_{K_1}(P)).$$

Bob now believes that his double-encryption method indeed doubles the key space size to $2^{2 \cdot 56} = 2^{112}$, and brute-forcing correspondingly requires $2^{112}$ cryptographic operations. How can you tell Bob that, under the known-plaintext attack, there is a way to find his two keys by performing $2 \cdot 2^{56} = 2^{57}$ cryptographic operations only?

Given a pair of plaintext $P_1$ and its corresponding ciphertext $C_1$,
an attacker can perform the following steps:

1. Decrypt the ciphertext $C_1$ using all $2^{56}$ possible values of $K_2$,
   and store all the recovered plaintexts in list $L$;
2. Encrypt the plaintext $P_1$ using all $2^{56}$ possible values of $K_1$,
   and check if one outputted ciphertext $x$ is in the list $L$;
3. The value of $K_1$ that is used to generate $x$ in Step 2 is the key $K_1$ used by Bob;
4. The value of $K_2$ that is used to generate $x$ in Step 1 is the key $K_2$ used by Bob.

The total no of *cryptographic* operations (i.e. encryptions and decryptions) in the steps above is: $2 \cdot 2^{56} = 2^{57}$. This attack technique is also known as the ***meet-in-the-middle*** attack, which can be mounted on ciphers that perform multiple encryption operations in sequence.

Note that, in Step 2, it is possible to have $i$ multiple entries $x_1, x_2, \ldots, x_i$, that exist in list $L$. Correspondingly, we do have $i$ possible pairs of $K_1$ and $K_2$.
In this case, the attacker just needs to test all these key-pair candidates using other available pairs of plaintext and ciphertext until there is only one unique applicable pair of $K_1$ and $K_2$.

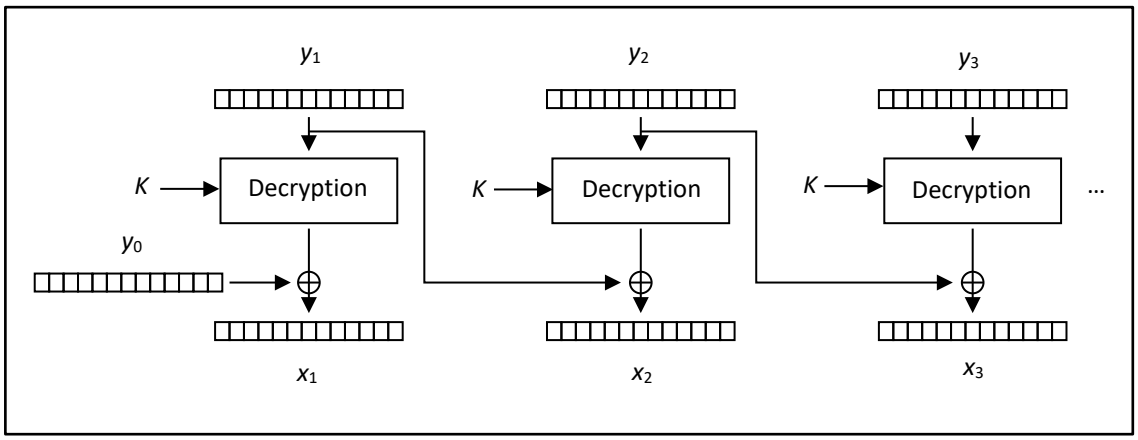
**C2**. *Mode-of-Operation* (4 marks)

Cipher Block Chaining (CBC) mode-of-operation is commonly used to encrypt a plaintext longer than a cipher's block. In CBC, each plaintext block is XOR-ed with the previous ciphertext block before being encrypted. An IV is used in encrypting the first plaintext block.

Mathematically, the encryption can thus be expressed as follows:
Given a $n$-block plaintext message $x_1, x_2, x_3, \ldots, x_n$, a secret key $K$, and an initial value $IV$, CBC outputs $(n+1)$-block ciphertext message $y_0, y_1, y_2, \ldots, y_n$, where:

- $y_0 = IV$;
- $y_k = \text{Enc}_K(x_k \oplus y_{k-1})$, for $k = 1, 2, 3, \ldots, n$.

a) (2 marks) Your lecture notes show a diagram depicting how a CBC-based encryption is done. Draw a diagram of the corresponding CBC-based *decryption*.




b) (1 mark) How is decryption affected if the first ciphertext block $y_0$ is removed from the ciphertext?

The plaintext block $x_1$ cannot be correctly recovered.
Other plaintext blocks $x_2, x_3, \ldots, x_n$, however, still can be recovered correctly.

c) (1 mark) Can the encryption processes of different blocks belonging to a plaintext run in parallel? How about the decryption of a ciphertext' s different blocks?

The encryption process *cannot* run in parallel. This is since the encryption at round $i$ to produce the ciphertext block $y_i$ does take as its input the ciphertext block $y_{i-1}$ that is generated only in the previous round $i$-1.

The decryption process *can* run in parallel. This is because the decryption at round $i$ to recover the plaintext block $x_i$ depends only on the ciphertext blocks $y_i$ and $y_{i-1}$, which are both readily available from the sent ciphertext.

**C3**. *Hash Generation and Time-Storage Requirements* (3 marks)

A black-hat hacker managed to obtain the password file of an authentication system. Like in the 2012 LinkedIn hack case, the authentication system fails to use a salt when hashing a password entry to be stored into the password file.

Suppose the hash function *h* employed by the system takes $2^{30}$ clock cycles to produce the 128-bit digest of an input. Now, the hacker wants to "crack" the passwords of all users in the authentication system by using a dictionary of 16M commonly-used passwords.

a) (2 marks) Using a 4GHz single-core processor, how long does it take to exhaustively compute the digests of *all* password entries in the dictionary?
***Note***: 1K = $2^{10}$, 1M = $2^{20}$, 1G = $2^{30}$, 1 year ≈ $2^{25}$ seconds.

The hash function *h* takes $2^{30}$ clock cycles to compute the digest of an input.
There are 16M = $2^4 \cdot 2^{20} = 2^{24}$ password entries in the dictionary.
Computing the digests of all password entries thus takes $2^{30} \cdot 2^{24} = 2^{54}$ cycles.

A 4GHz single-core processor has $2^2 \cdot 2^{30} = 2^{32}$ cycles per second.
To generate all the digests, the processor thus needs $2^{54} / 2^{32} = 2^{22}$ seconds.
Since 1 year ≈ $2^{25}$ seconds, the total time needed is therefore:
$2^{22} / 2^{25} \approx 2^{-3} \approx 1/8$ year ≈ 1.5 months.

b) (1 mark) The hacker knows that he needs to quickly access his target authentication system once its password file is obtained. For his future cracking of weak salt-less authentication systems, he wants to pre-generate the digests of *all* password entries in the dictionary. For this time-memory trade-off (TMTO) effort, how much extra storage will the hacker need to store all the computed digests in his full lookup table? Express your answer in MB (megabyte) or GB (gigabyte).
**Note**: Please clearly differentiate bits and bytes in your answer.

Storing a digest requires 128 bits = 16 = $2^4$ bytes.

Storing the digests of all password entries in the dictionary thus requires:
$2^4 \cdot 2^{20} \cdot 2^4 = 2^{28}$ bytes
$= 2^{28} / 2^{20} = 2^8 = 256$ MB
= ¼ GB.

**C4.** *Birthday Attacks (4 marks)*

a) (2 marks) A car park is having 150 parked vehicles. The license plate of each vehicle contains a 4-digit number. Assuming a uniform probability distribution of 4-digit vehicle numbers (i.e. from 0000 to 9999), is there a good probability that two of the license plates currently in the car park have the same 4 digits? Explain why.

We need to apply the *standard* birthday attack formula.

Let $M$ = 150 and $T$ = 10,000.

We can see that the two numbers satisfy the following condition
$M > 1.17 \cdot \sqrt{T}$ , since 150 > 1.17 . 100 = 117.

Hence, the probability that there exist two vehicles with the same license plate digits in the car park is *greater* than 0.5.

b) (2 mark) Suppose Bob managed to obtain $2^{20}$ different digests that were generated by a hash function employed by a target system. The hash function outputs 8-byte digest of a message. Bob now wants to find a message that hashes into 1 (one) of the obtained digests. How many different messages should Bob approximately hash until there is a good probability that a generated digest will match 1 of the obtained digests? Show your working clearly and succinctly.

Let's apply the birthday attack *variant*.

Let $n$ = 8 . 8 = 64, and $k = 2^{20}$.

To simplify our calculation, we can assume the probability of 0.63 that Bob will generate a digest matching 1 of his previously-obtained digests.

For this probability, we need $m$ that satisfies: $k \cdot m = 2^n$.
Hence, $m = 2^n / k = 2^{64} / 2^{20} = 2^{44} = 2^4 \cdot 2^{40}$ = 16 T.
That is, Bob must approximately hash 16 T of different input messages.

This page is intentionally left blank.
You can use the space below if you need more space for your answers.

**~~~ END OF PAPER ~~~**