NATIONAL UNIVERSITY OF SINGAPORE

CS2107 – Introduction to Information Security

(AY2021/22 Semester 1)

Mid-Term Quiz

Date: 1 Oct 2021 Time: 10:15 - 11:30AM

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

NAME :

INSTRUCTIONS TO CANDIDATES

1. This question paper consists of **NINETEEN (19)** questions in **THREE (3)** parts; and comprises **EIGHT (8)** printed pages, including this page.
2. Fill in your Student Number and Name 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. In Microsoft's STRIDE security threat model, two types of threat are *tampering with data* and *spoofing of user identity*. To deal with the latter, which cryptographic technique below can be used?

a) Block cipher
b) Stream cipher
c) Public-key encryption
d) Hash
e) MAC

A2. Alice and Bob use a common secret key to ensure the confidentiality of their communication. To increase the security of the cryptographic scheme employed, they use *two* different keys at the same time. They don't realize, however, that an adversary with sufficiently good computing power can brute force the scheme (a fact that is widely known in the security community). Which scheme below is the most likely one used by Alice and Bob?

a) One-Time Pad
b) DES
c) AES
d) HMAC
e) RSA

A3. You are tasked with deploying a multi-modal biometrics-based system to regulate an access to a very restricted *top-secret* document room. Which policy should you enforce on the document-room door's access?

a) Its False Match Rate (FMR) must be very high
b) Its successful false matches over all attempted false matches can be medium
c) Its (normalized) threshold value must be set to a value greater than 0.7
d) Its (normalized) threshold value must be set to a value lower than 0.5
e) Its Failure-to-Capture rate (FTC) should be high

A4. The following measures can prevent a Padding Oracle attack by an attacker as discussed in the class, *except*:

a) Avoid using the CBC mode-of-operation in encrypting a plaintext
b) Additionally employ MAC to protect the integrity of the ciphertext (including IV)
c) Additionally employ digital signature to protect the authenticity of the ciphertext (including IV)
d) Additionally encrypt each IV used so that the attacker cannot send its modified values
e) Do not entertain multiple decryption queries/requests on the same ciphertext from the same sender

A5. Suppose a hash function produces a 100-bit digest. Mallory wants to repeatedly generate message digests so that, with a probability of more than 0.5, she has two messages with the same digest. What is the *minimum* number of messages among the options below should Mallory hash in order to achieve her objective?

a) 2^{40}
b) 2^{50}
c) 2^{51}
d) 2^{60}
e) 2^{100}

Part B (10 marks): Security Terminology

Instructions:

The next ten questions (B1 to B10) give security-related descriptions, which are taken from various articles/writings on the Internet. 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
Self-signed certificate
Certification Authority
Certification path
Hash
MAC
Authenticated encryption
Nonce
Mode-of-operation

Cryptography Notions:
Symmetric Key Cryptography
Public Key Cryptography
RSA scheme
Public Key Infrastructure
Kerckhoffs' principle
One way

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

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

B1. A/an [block cipher] operates on a fixed-sized block of input, and can provide both high diffusion and high confusion properties.

B2. Petya is a piece of malware that targets Microsoft Windows-based systems. It infects the master boot record to execute a payload that encrypts a hard drive's file system table, and prevents Windows from booting. It subsequently demands that the user make a

payment in Bitcoin in order to regain access to the system.
Petya performs a/an [Denial of Service attack] on victim user's data.

B3. A different [Initial Value (IV)] must be selected every time a plaintext needs to be encrypted, and it will be sent to the recipient in clear as part of the generated ciphertext.

B4. In [Public Key Cryptography \| RSA scheme], two different keys are issued to and are employed by an entity in the communication system, namely public key and private key.

B5. The trust anchor for the digital certificate is the root certificate authority (CA). Since no other entity issues and signs the certificate of a root CA, a root CA typically uses a/an [self-signed certificate].

B6. A/an [Certification Authority] issues certificates for HTTPS servers, traditionally by using a "domain validation" technique in authenticating the certificate requester.

B7. When transfer of information from a machine inside an organization to an external host on the Internet is not allowed by the applicable security policy, a/an [covert channel] is sometimes employed by an attacker.

B8. One type of [side-channel attack] is timing attack, which exploits externally observable information extracted or inferred from the implementation of cryptographic schemes or protocols.

B9. A/an [botnet] is a number of Internet-connected attacker-controlled devices, which can be used to perform Distributed Denial-of-Service (DDoS) attacks.

B10. A/an [dictionary attack] is carried out on an authentication system by trying all candidate entries from a set of popular words and terms.

Part C (15 marks): Scenario-based Questions

Instructions: Write your answers in the spaces provided. For questions that require calculations, please **show your workings sufficiently**. Giving correct answers without any workings shown will give you ***partial marks only***!

C1. *Strength Analysis of a Cipher* (3 marks)

Bob wants to use the **permutation/transposition** cipher to encrypt his English plaintexts. Suppose Bob sets the length of each plaintext block to 15 characters, and this block-length value is made publicly known by Bob.

a) (1 mark) What is the **key space size** of the permutation cipher as used by Bob?

15!

b) (2 marks) Suppose now Bob aims to strengthen his permutation/transposition cipher. In encrypting a 15-character plaintext block, he first divides it into 3 sub-blocks of 5 characters, and then apply a (first) permutation on the 3 sub-blocks using the key p_1. Subsequently, within each of the 3 sub-blocks, he performs another (second) permutation using the key p_2. Again, Bob makes public of 15 (the block-length value), 3 (the number of sub-blocks) and 5 (the number of characters in each sub-block).
What is the **key space size** of Bob's new permutation cipher?
Is it *really stronger* than the original one used in Part (a)? Please compare the strength of both schemes.

3! * 5!

Comparing the two key space sizes, you can easily see that: 15! > 3! * 5!
Hence, Bob's new permutation cipher is *not* stronger than his original one.

C2. *Encryption Scheme's Requirements* (4 marks)

Bob attempts to construct a special substitution cipher, which also encrypts each letter in the plaintext into another letter. In Bob's scheme, the alphabet consists of only the 26 lowercase letters, i.e. {a, b, ..., z}. Each of these letters gets mapped into a number according to the following mapping scheme: a → 0, b → 1, ⋯ , z → 25.

For his cipher's encryption, Bob uses a *multiplication* operation in modular arithmetic. He defines his encryption of a number x by key k as follows: $E_k(x) = (x * k) \bmod 26$.

a) (2 marks) Bob wants to verify that his cipher meets the *correctness requirement* of a cipher. For that, he first considers if his encryption key k=4, and checks all possible encryption outputs.
Tell something about Bob's cipher under this selected particular key: is the cipher's correctness requirement met? If so, please argue succinctly for that. Otherwise, do point out the issue, for instance, by giving a counterexample.

> With the selected k=4, the cipher's correctness requirement is *not* met.
> This is since the defined encryption function is *not* bijective,
> i.e. it is *not* a one-to-one correspondence.
> For instance, $E_4(1) = E_4(14) = 4$.
>
> As such, the cipher's corresponding decryption, although it is not given/ shown in the problem description, *cannot* always give the original plaintext.
> In other words, the requirement of $D_k(E_k(x)) = x$ for all $x \in \{0, 1, 2, \ldots, 25\}$ does *not* hold.

b) (2 marks) Next, Bob investigates if his encryption key k=3, and again checks all possible encryption outputs.
Tell something about Bob's cipher when this key is selected: is the cipher's correctness requirement met? If so, please argue succinctly for that. Otherwise, do point out the issue, for instance, by giving a counterexample.

> With k=3, you can see that each number in the set {0, 1, 2, …, 25} gets encrypted uniquely into a number in the set.
> That is, when 3 is the "multiplication factor" used in the defined encryption, the encryption function becomes a bijection or one-to-one correspondence.
> (In fact, the corresponding decryption function can be defined by multiplying the ciphertext with the multiplicative inverse of 3.)
>
> Given the above, the cipher's correctness requirement is thus met,
> i.e. the requirement of $D_k(E_k(x)) = x$ for all $x \in \{0, 1, 2, \ldots, 25\}$ holds.

C3. *Hash Generation and Time-Memory Trade-Off* (4 marks)

A black-hat hacker managed to obtain the password file of an authentication system. The authentication system, however, *fails to incorporate a salt* when hashing a password entry to be stored into the extracted password file.

Suppose the hash function *h*() employed by the system takes 2^{40} clock cycles to produce the 256-bit digest of an input. Now, the hacker wants to offline-crack the passwords of all users in the extracted password file by using a dictionary of 1M (2^{20}) commonly used passwords.

a) (2 marks) Using his 4GHz *quad*-core processor, how long does it take for the hacker to exhaustively compute the digests of all password entries in the employed dictionary?

Note: 1K = 2^{10}, 1M = 2^{20}, 1G = 2^{30}, 1T = 2^{40}, 1 year ≈ 2^{25} seconds.

The hash function *h*() takes 2^{40} clock cycles to compute the digest of an input.
There are 1M = 2^{20} password entries in the dictionary.
Computing the digests of all password entries thus takes $2^{40} . 2^{20} = 2^{60}$ cycles.

A 4GHz quad-core processor has $2^2 . 2^2 . 2^{30} = 2^{34}$ cycles per second.
To generate all the digests, the processor thus needs $2^{60} / 2^{34} = 2^{26}$ seconds.
Since 1 year ≈ 2^{25} seconds, the total time needed is therefore: $2^{26} / 2^{25} \approx 2$ years.

(***Note on answer marking***: If you happen to interpret "1M (2^{20})" as 1M × (2^{20}) instead of the intended 1M (= 2^{20}), I will evaluate your *calculation logic* instead of checking the total time needed as produced by your calculation.)

b) (2 marks) The hacker realizes that he needs to quickly access his target authentication system once its password file can be pwned (obtained). For his future cracking of salt-less authentication systems, he wants to pre-generate the digests of all password entries in the dictionary. How much storage will the hacker need in order to construct his full lookup table, where each entry in the table consists of both a clear password and its hashed version?

You can just state the *storage-size increase* relative to the original dictionary file due to additionally storing all the digests (hashed passwords). Express the size increase in MB (megabyte), GB (gigabyte), or TB (terabyte).
Note: Please clearly differentiate bits and bytes in your answer.

The full lookup table stores all the password entries in the dictionary together with the corresponding hashed passwords.

Storing a digest requires 256 bits = 32 = 2^5 bytes.
Hence, storing the digests of all password entries requires:
$2^{20} . 2^5 = 2^{25}$ bytes = $2^{25} / 2^{20} = 2^5$ = 32MB.

The size of the full lookup table is therefore the size of the dictionary + 32MB.
(***Note on the answer***: the size of the dictionary should not be more than 32MB since a password is typically shorter than 32 characters/bytes.)

(***Note on answer marking***: The size of the dictionary is not given. It's fine if you make your own reasonable assumption about the size, since I will mainly evaluate the size increase due to additionally storing all the generated digests.)

C4. *RSA: RSA Numbers, and SQ*[3] (4 marks)

a) (2 marks) To test your understanding of how (the classroom) RSA works, do answer the two given questions (i-ii) below.
Alice uses RSA encryption for her secure communication internally in her company. Suppose Mallory knows that Alice's public key uses the fixed exponentiation parameter e=3. However, the RSA modulus n is unknown to Mallory.
One day, due to Alice's carelessness, Mallory finds out that Alice's $\varphi(n)$ is 352=16*22.
(i) Which among the following possible values should be d (and explain briefly why): 67, 137, 167, or 235?
(ii) What is the modulus n used by Alice?

(i) d is 235, since 3 * 235 = 1 (mod 352).

(ii) $\varphi(n)$ = 16 * 22 = (p-1) * (q-1).
Hence, p = 17 and q = 23.
The modulus n is thus: p * q = 17 * 23 = 391.

b) (2 marks) Bob wants to use the classroom RSA scheme to **message Alice** without any PKCS paddings or authenticity-related measures incorporated. Bob encrypts a number m that represents the amount of money he has transferred to Alice. Bob's generated **ciphertext** is: $c = m^e \pmod n$, with (n, e) as Alice's RSA public key.

Mallory gets jealous by Bob's action. Mallory happens to be a **woman-in-the-middle**, who can intercept c from Bob, modify it into c', and then send c' to Alice. Now, Mallory wants to turn m into m^3 in c', but without knowing m (as she can only see c). What should Mallory set c' to? Explain briefly why your c' should work.
Note: The number m is an integer, and for simplicity in this case is $1 < m < 10{,}000$.

Mallory can set c' for Alice to: $(m^e)^3 \pmod n =$
$(m^{e3}) \pmod n = (m^{3e}) \pmod n = (m^3)^e \pmod n$.

(***Note***: You can refer to the following links for "exponentiation rules": https://mathinsight.org/exponentiation_basic_rules, https://en.wikipedia.org/wiki/Exponentiation#Identities_and_properties.)

When Alice recovers the plaintext, it will be:
$(m^3)^{e^d} = (m^3)^{ed} = m^3 \pmod n$.

~~~ END OF PAPER ~~~