

ID:



Friday 22 February 2019 9:00 to 10:50 AM

ECS655U Security Engineering Midterm (20%) Duration: 110 minutes

**YOU ARE NOT PERMITTED TO READ THE CONTENTS OF THIS QUESTION PAPER UNTIL
INSTRUCTED TO DO SO BY AN INVIGILATOR.**

Instructions: Check whether the student ID on this page matches yours. This paper contains FOUR questions. **Answer all questions.** Cross out any answers that you do not wish to be marked.

Calculators are not permitted in this examination.

Possession of unauthorised material at any time when under examination conditions is an assessment offence and can lead to expulsion from QMUL. Check now to ensure you do not have any notes, mobile phones, smartwatches or unauthorised electronic devices on your person. If you do, raise your hand and give them to an invigilator immediately.

Exam papers must not be removed from the exam room.

Student ID Number:

Question	Points	Score
1	30	
2	30	
3	30	
4	30	
Total:	120	

Leave this table blank.

Examiner: Dr Arman Khouzani

© Queen Mary, University of London, 2019

Question 1

(a) This question is about basic security services and cryptographic primitives.

(i) Provide a short definition for each of the following security services:

[5 marks]

▷ Data Origin Authentication:

▷ Non Repudiation:

(ii) Does “Non-Repudiation” need “Data-Origin Authentication”? In other words, do we need to have data origin authentication if we want to have non-repudiation? Support your answer with a brief explanation.

[5 marks]

(iii) Determine which one of the above two security services are provided by “digital signatures”. Provide a brief explanation with your answer.

[5 marks]

Turn over

- (b) This question is regarding a generic “symmetric-key” cipher. Using the following notations, answer each question.

Notation	Description
m	a message (plaintext)
c	a ciphertext
k	a secret key
$E(m, k)$	Encryption algorithm applied to message m (the first argument) using secret key k (the second argument)
$D(c, k)$	Decryption algorithm applied to ciphertext c (the first argument) using secret key k (the second argument)
\forall	a logic symbol which means “given any”, or “for all”. For example: $\forall k, c$ means given any key k and c (i.e., for any given k and any given c)

- (i) Provide the “correctness” condition of this (symmetric-key) cipher. That is, what relation needs to hold for this cipher to be “correct”? Your answer can be very briefly stated in terms of the notations in the table. But if you cannot, express it in words using as much of the notations in the table as you can. **[5 marks]**

- (ii) Does the following statement hold for a correct symmetric cipher? That is, does a symmetric cipher have to satisfy the following property? Provide a brief reasoning. **[5 marks]**

$$\forall k, m_1, m_2 : m_1 \neq m_2 \Rightarrow E(m_1, k) \neq E(m_2, k)$$

- (iii) Answer the above question about the following statement. Again, is the following property necessary for a symmetric cipher? Support your answer with a brief reasoning. **[5 marks]**

$$\forall k_1, k_2, m_1, m_2 : m_1 \neq m_2, k_1 \neq k_2 \Rightarrow E(m_1, k_1) \neq E(m_2, k_2)$$

Question 2

- (a) Figure 1a shows an image in a format in which each pixel is represented in one byte. Figures 1b and 1c show the result of AES encryption of that image using ECB and CBC modes of operations respectively (using the same key). Elaborate why we can see a pattern of the original image in Fig. 1b but not in Fig. 1c?

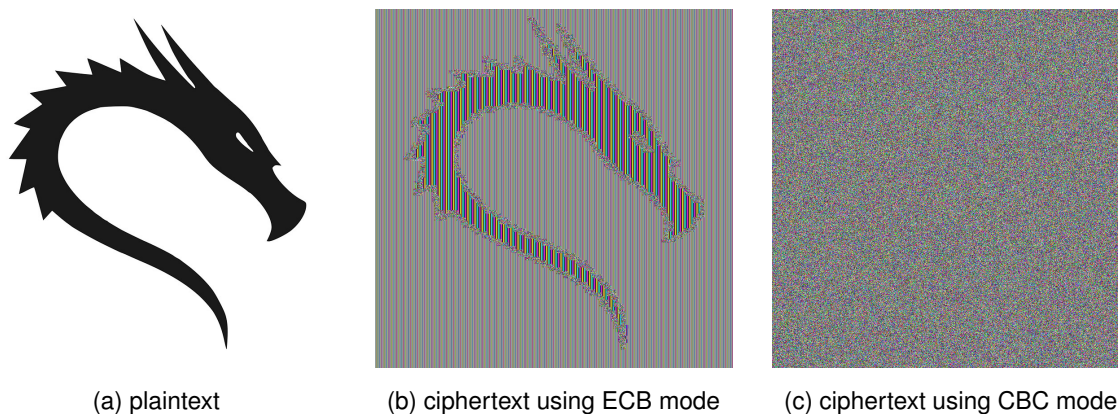
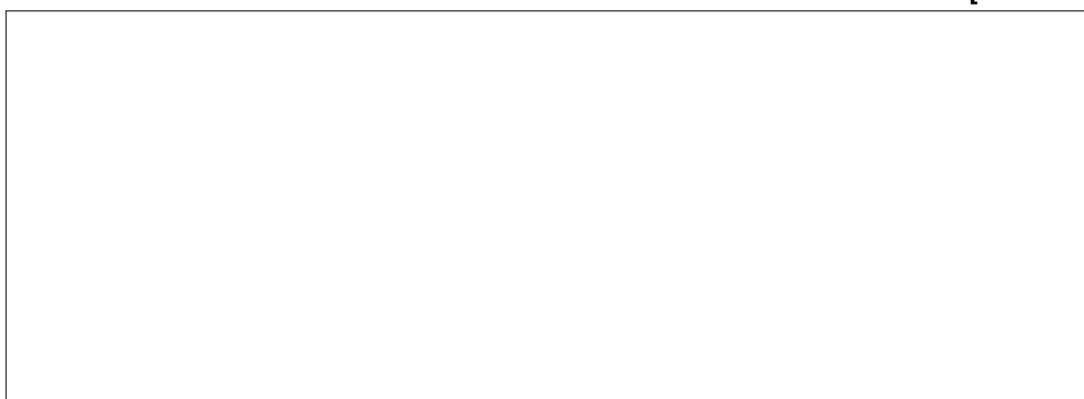


Figure 1: Encryption of pixel-encoded image (a) using the ECB mode of operation (b) and the CBC mode of operation of the AES block cipher.

[10 marks]



- (b) Consider the following diagram describing the encryption algorithm in a particular mode of operation of a block cipher like AES, and answer the subsequent questions.

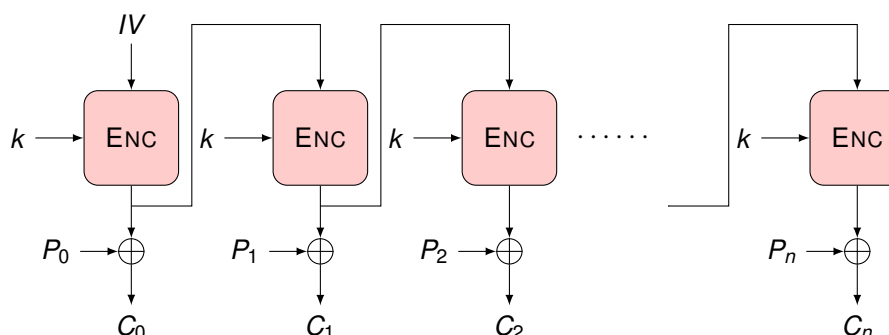


Figure 2: Q2(b) Notations: P_0, \dots, P_n : blocks of plaintext. C_0, \dots, C_n : blocks of ciphertext. k : cryptographic key. ENC: an encryption block. \oplus : XOR (eXclusive OR) operator. IV: initialisation vector.

Turn over

- (i) Draw the diagram of the corresponding decryption process (with clear labels). Hint: this mode of operation was not explicitly presented in class, but you should be able to answer this question from basic principles. Start from C_0 and try to recover P_0 from it, and then P_1 , P_2 , and from there a general P_i . **[10 marks]**

- (ii) Analyse the “error-propagation” of this mode of operation (in Figure 2). That is, if a single bit of the ciphertext is flipped (e.g. due to noise in the channel), what would be the effect on the recovered plaintext at the receiver. **[5 marks]**

- (iii) Provide one advantage and one disadvantage of this mode of operation compared to the CTR (i.e., Counter) mode. **[5 marks]**

Question 3

- (a) A cryptographic hash function on an input has created the following output in hex (base 16) format (40 hexadecimal digits):

644cf7d904b7f65b536627b54cb4f9ba0840fc28

- (i) On average, how many random inputs should you compute with this cryptographic hash function in order to find an input that gives you this specific output with at least 50% chance? **[5 marks]**

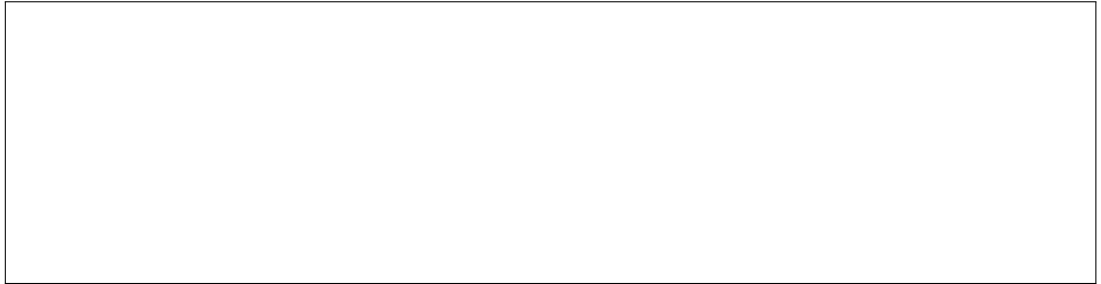
- (ii) If a single bit of the input changes, how many of the output hex characters do you expect to change on average? (Recall that each hex character/digit represents 4 bits.) **[5 marks]**

- (b) A digital signature has a signing and a verification algorithm. Focusing on the verification algorithm, identify its input and output arguments (what input arguments does it take and what output(s) does it produce?). **[5 marks]**

(c) This part is regarding Public key certificates.

- (i) What are the essential components of a public key certificate (without which, the certificate cannot serve its basic functionality)?

[5 marks]



- (ii) Briefly describe the procedure of verifying a certificate (What do we need to do to verify a certificate?)

[5 marks]



- (iii) Suppose a certificate verifies. What conclusion can we make?

[5 marks]



Question 4

(a) Consider the following cryptographic protocol:

- Alice (the sender) and Bob (receiver) have established two symmetric keys k_1 and k_2 . That is, k_1 and k_2 are only known by (both) Alice and Bob.
- Both parties have also already agreed on the choice of a strong symmetric-key encryption algorithm E , and a strong message-authentication-code algorithm MAC .
- Alice (the sender) performs the following on her message m , and sends the output y to Bob over a public channel:

$$y = E_{k_1}(m) \parallel MAC_{k_2}(E_{k_1}(m))$$

- (i) Describe the corresponding process in the receiver; i.e., what does Bob do to y upon receiving it?

[5 marks]

- (ii) For each of the following security services, determine whether our protocol provides it. Each of your answers should be supported by a brief but clear justification:

- Confidentiality
- Data Integrity

[5 marks]

- (b) Suppose Alice wants to access her photos on her account on `wonderland.wl`, which are located at the following url: `wonderland.wl/users/alice/photos`. As a means of authentication, she needs to create the following HTTP request:

```
GET /users/alice/photos HTTP/1.1
Host: wonderland.wl
Authentication: hmac username:[digest]
```

in which:

```
digest = base64encode(hmac("sha512", "key", "GET || /users/alice/photos"))
```

where “key” is a shared secret key between Alice and the server, and “sha512” is the hashing algorithm to be used in computing the HMAC. Note that `base64encode` is just a text-based encoding of binaries.

There is unfortunately no SSL/TLS and we have to send this request over TCP/IP directly. Can an attacker (Eve) that can sniff the traffic use this request to POST a new photo to Alice's account? Explain your answer.

[5 marks]

Hint: Eve needs to be able to construct the following message (a http post request with the correct header):

```
POST /users/alice/photos HTTP/1.1
Host: wonderland.wl
Authentication: hmac username:[digest]
```

where

```
digest = base64encode(hmac("sha512", "key", "POST || /users/alice/photos"))
```

- (c) Now suppose that the system is upgraded and in the new system, Alice needs to create the following digest instead:

```
digest = base64encode(hmac("sha512", "key",
    "GET || /users/alice/photos || <timestamp> || <nonce>"))
```

where `<nonce>` is a pseudorandom number generated by Alice, and the timestamp is a global time-stamp in the internet, e.g. 28 FEB 2019 10:00:00.

Alice then sends the following GET request to view her photos:

Turn over

```
GET /users/alice/photos HTTP/1.1
Host: wonderland.wl
Authentication: hmac username:<nonce>:[digest]
Timestamp: 28 FEB 2019 10:00:00
```

- (i) What problem is the new upgrade trying to resolve? Explain with a specific example scenario.

[5 marks]

- (ii) Why does Alice need to include the nonce “in the clear” as well as using it inside the digest? (pay attention to the line “Authentication: hmac username:<nonce>:[digest]”).

[5 marks]

- (iii) What happens if we had only used the <nonce> and not timestamp in this new mechanism? What about if we had only used the time-stamp?

[5 marks]

End of questions