SECURITY IN COMPUTING AND IT COSC2536/2537

ASSIGNMENT #2

Total Marks: 35 (Contributes 35% of the total Grade)

Submission Deadline: Week 10, Friday the 28th September 2018 11:59pm

Special Instructions:

- **There are 02 (two) parts of the assignment: PART-1 and PART-2.**
- **Questions from Q1 to Q4 belong to PART-1, and Q5 to Q6 belong to PART-2.**
- **You are required to submit PART-1 as a PDF or Word Document format by uploading in the CANVAS on or before due date.**
- **You are required to show all of the steps and intermediate results for each of the questions of PART-1.**
- **Please DO NOT provide ONLY codes as answers for PART-1.**
- **The PART-2 should be demonstrated during WEEK-10 to 12 tutorials.**

PART-1 (Submit in the CANVAS)

Q1. (Privacy-Preserving Computation) [6 Marks]

Suppose there are **11 voters** to vote for **YES** or **NO** to give their opinions.

YES NO

Design a secure voting prototype as shown in **Figure-Q1** using **Paillier cryptosystem** where the votes must be **encrypted** from **Voting Booth** before sending them to the **Voting Server**.

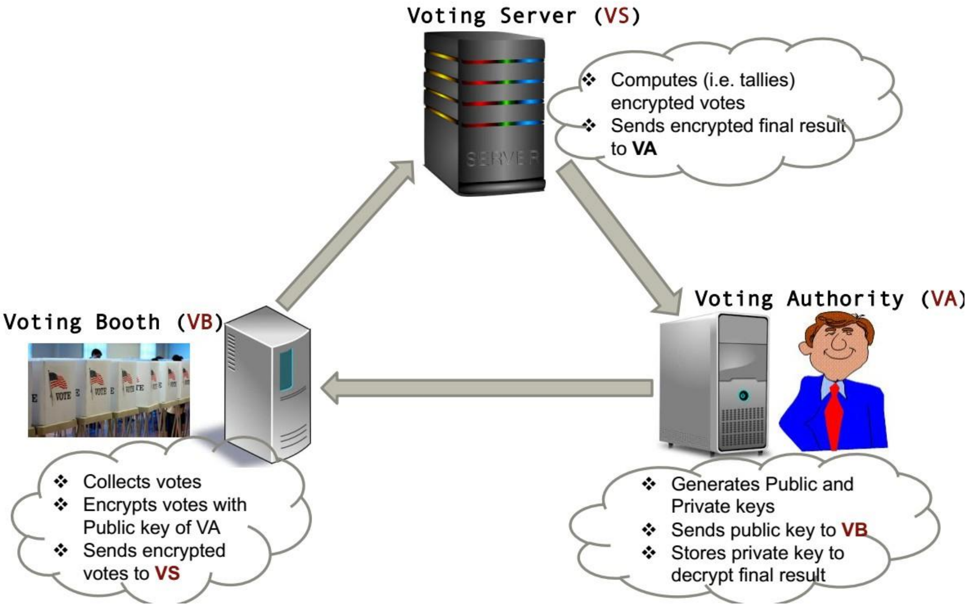

Voting Server (VS)
❖ Computes (i.e. tallies) encrypted votes
❖ Sends encrypted final result to VA
Voting Booth (VB)
Voting Authority (VA)
❖ Collects votes
❖ Encrypts votes with Public key of VA
❖ Sends encrypted votes to VS
❖ Generates Public and Private keys
❖ Sends public key to VB
❖ Stores private key to decrypt final result

Figure-Q1: Secure voting system

Assume, **seven voters** will vote for **YES** and **four voters** will vote for **NO**. The **Voting Authority** should find **seven YESs** and **four NOs** after counting the votes. The **Voting Authority** chooses ***p=59***, ***q=97*** and select ***g=5724***. The private numbers chosen by **11 voters** and their votes are as follows:

Voter No.	Voter's Private Number, *r*	Vote	Voting message, *m*
1	40	YES	**00010000 = 16**
2	41	YES	**00010000 = 16**
3	42	YES	**00010000 = 16**
4	43	YES	**00010000 = 16**
5	44	YES	**00010000 = 16**
6	45	YES	**00010000 = 16**
7	46	YES	**00010000 = 16**
8	47	NO	**00000001 = 1**
9	48	NO	**00000001 = 1**
10	49	NO	**00000001 = 1**
11	50	NO	**00000001 = 1**

Show the **encryption**, **homomorphic computations and decryption** processes. [***Hints***: Refer to the lecture-5 Secure e-voting. You need to represent the total number of votes by **8-bit string**. The first **4 (four)** bits should represent the votes for **YES** and the rests for **NO.** When adding a vote for **YES**, the system adds **00010000**, which is **16** in integer. Similarly, the system adds **00000001** when voting for **NO**, which is **1** in the integer form.]

Q2. (**Signatures**) **[2+2+2+1 = 7 Marks]**

Q2.1

Suppose Bob (the sender) wants to send a message ***m***=**123456** to Alice (the receiver). However, before sending the message he would like sign the message. When Alice receives the signed message, she would like to verify that the message is indeed from Bob. To facilitate signing and verification Bob generates public and private keys using **RSA encryption algorithm** and sends the public key to Alice. Bob uses parameter ***p* = 10193** and ***q* = 8287**, and chooses a suitable public key parameter ***e=5903*.** How would Bob sign message ***m*=123456**? How would Alice verify the signed message from Bob?

[***Hints:*** Refer to the lecture-6 and tutorial-6. You do not need to generate hash of the message ***m***.]

Q2.2

Suppose Bob (the sender) wants to send a message ***m***=**5432** to Alice (the receiver). However, before sending the message he would like sign the message. When Alice receives the signed message, she would like to verify that the message is indeed from Bob. To facilitate signing and verification Bob generates public and private keys using **ElGamal encryption algorithm** and sends the public key to Alice. Bob chooses ***p***=**9721***,* ***g***=**1909***,* ***x***=**47.** How would Bob sign message ***m*=5432**? How would Alice verify the signed message from Bob?

[***Hints:*** Refer to the lecture-6 and tutorial-6. You do not need to generate hash of the message ***m***.]

Q2.3

Suppose Bob (the sender) wants to send a large text message ***M*** to Alice (the receiver). You should download the text message file "**Message.txt**" from the CANVAS. The text message ***M*** is as follows:

Was every secret code used during the war cracked? The answer to that final question is a stunning surprise: the skilled code breakers of the time weren't able to crack every coded message sent during World War II. In fact, until recently, some messages sent by German agents were still coded, the world and the Allied Forces unsure of what the contents said.

Before sending the message, Bob generates a **hash *h(M)*** of the text message ***M*** using MD5 hash algorithm, and converts ***h(M)*** into integer message ***m***. Then, he signs the ***m*** and sends it to Alice. When Alice receives the signed message, she would like to verify that the message is indeed from Bob. To facilitate signing and verification Bob generates public and private keys using **RSA encryption algorithm** and sends the public key to Alice. Bob uses the following parameters:

p = 307699126915021078949717556805305347641

q = 286189067004968539490940912607240844261

Bob chooses a suitable public key parameter ***e=47.*** How would Bob sign message ***M***? How would Alice verify the signed message from Bob?

[***Hints:*** Refer to the "*Running Example of RSA Signature for Text Message"* of lecture-6. The document can be found here: https://rmit.instructure.com/courses/46189/files/3608593/download?wrap=1

Use the following links:

For generating MD5 hash: http://www.miraclesalad.com/webtools/md5.php

For converting hexadecimal to decimal:
https://www.rapidtables.com/convert/number/hex-to-decimal.html]

Q2.4

Suppose that Alice signs the message ***M*** = "I love you" and then encrypts it with Bob's public key before sending it to Bob. Bob can decrypt this to obtain the signed message and then encrypt the signed message with, say, Charlie's public key and forward the resulting ciphertext to Charlie. When Charlie receives the message he decrypts it and is amazed to see such a message which appears to come from Alice. What really went wrong here? Could Alice prevent this "attack" by using symmetric key cryptography? [***Hints:*** Refer to the lecture-9 and tutorial-9]

Q3 (BlockChain Technology) (Answer any 1 of the followings) [14 Marks]

(a) Write a report on how the blockchain technology can impact **IoT distributed systems, sensors and data**. Please consider the followings in your report sequentially:

i. Explain a motivating scenario of an IoT distributed system where the blockchain can be applied.

ii. Explain your understanding with necessary diagrams on how the specified IoT distributed system can be designed using blockchain.

iii. Among the popular consensus mechanisms, which one can be applied in your specified blockchain based IoT distributed systems and why? Justify your answer.

iv. Explain how the **integrity** and **traceability** of IoT data are obtained using blockchain in your specified scenario.

v. What are the advantages and disadvantages of using blockchain technology in your specified IoT distributed systems?

[***Hints:*** Resources on existing blockchain based IoT distributed systems should be reviewed from online sources. Some of the sample projects can be found in the following link:

https://www.postscapes.com/blockchains-and-the-internet-of-things/

]

(b) Write a report on how the blockchain technology can revolutionize **real estate industry**. Please consider the followings in your report sequentially:

i. Explain a motivating scenario on a real estate application where the blockchain can be applied.

ii. Explain your understanding with necessary diagrams on how the specified real estate application can be designed using blockchain.

iii. Among the popular consensus mechanisms, which one can be applied in your specified blockchain based real estate application and why? Justify your answer.

iv. Explain how the **integrity** and **traceability** of data in real estate are obtained.

v. What are the advantages and disadvantages of using blockchain technology in your specified real estate application?

[***Hints:*** Resources on existing blockchain based real estate application should be reviewed from online sources. Some of the sample projects can be found in the following link:

https://ico.rentberry.com/

https://atlant.io/

https://www.beetoken.com/

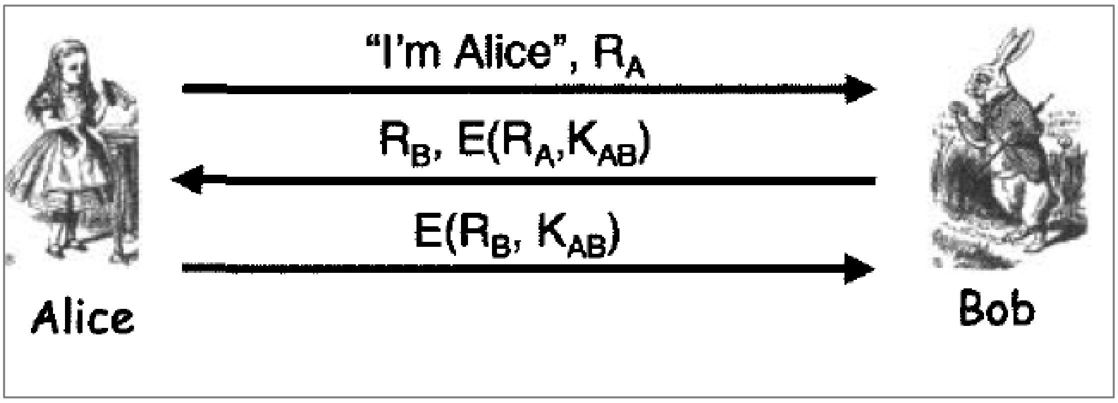
https://www.ubitquity.io/web/index.html

]

Q4 (Authentication Protocol) **[1.5+1.5 = 3 Marks]**

Q4.1

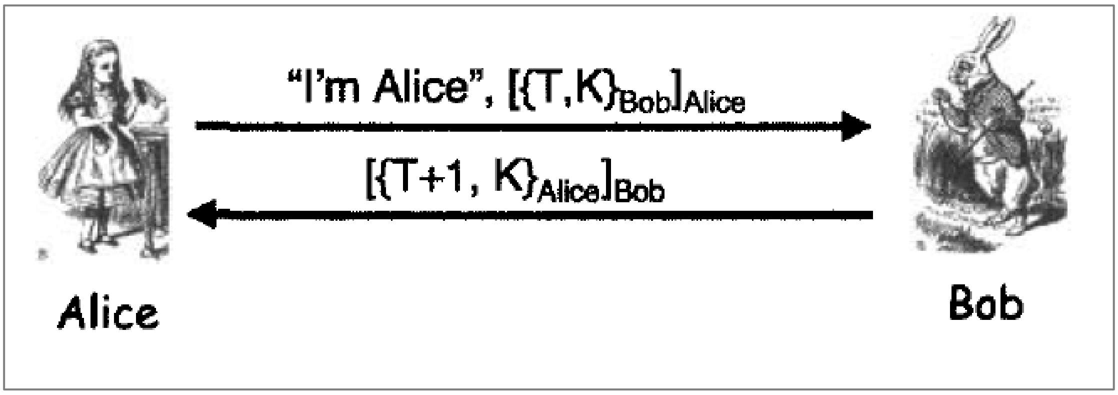
The following mutual authentication protocol is based on a shared symmetric key $\boldsymbol{K_{AB}}$. It can be shown that Trudy can attack the protocol to convince Bob that she is Alice, where, as usual, we assume that the cryptography is secure. Modify the protocol to prevent such an attack by Trudy. Justify your answer.



[*Hints*: The attack scenario was explained during the lecture. You need to show the modified protocol that prevents such attack with proper explanation.]

Q4.2

Consider the following mutual authentication and key establishment protocol, which employs a timestamp ***T*** and public key cryptography. It can be shown that Trudy can attack the protocol to discover the key ***K*** where, as usual, we assume that the cryptography is secure. Modify the protocol to prevent such an attack by Trudy. Justify your answer.


"I'm Alice", $[\{T,K\}_{Bob}]_{Alice}$
$[\{T+1, K\}_{Alice}]_{Bob}$
Alice
Bob

[***Hints***: The attack scenario was explained during the lecture. You need to show the modified protocol that prevents such attack with proper explanation.**]**

[Please turn over for PART-2]

PART-2 (Demonstrate during Tutorial)

Q5 (Data Hiding) [2 Marks]

Assume that you have a *cover image file* “mona_lisa.jpg” (refer to the **Figure-Q5**). You should download the image file from the CANVAS. The image has ***2196 pixels*** horizontally and ***2860 pixels*** vertically. In other words, the image width is ***2196 pixels*** and height is ***2860 pixels***. You need to hide your **student number** in the image file using **LSB Image steganography** technique and produce the *stego image file* “stego_mona_lisa.jpg”. Use any programming language (ex: JAVA, Python, etc.) to perform this task.

Demonstrate your code during tutorials within **WEEK 10 to 12**.

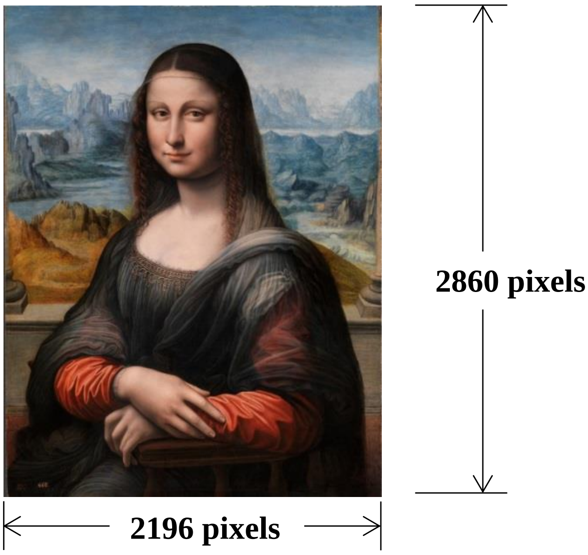


Figure-Q5: The image “mona_lisa.jpg”

Pixels to hide your student number are provided in **Table-Q5**. Each pixel represents **RGB** (**RED**, **GREEN** and **BLUE)** value. You are required to demonstrate the followings:

1. Convert your student number into binary.
2. Hide a bit of the binary string of your student number in the least significant bits (LSBs) of **RED** value in a pixel.
3. Generate new **RED** values for the specified pixels.
4. Produce the ***stego image file*** using new **RED** values.
5. Do you find any difference between the actual image and the new image after embedding secret message? Justify your answer.

[***Hints:*** Convert your student number into binary. For example, your student number is **3123456**. The equivalent binary string of the student number **3123456** is **1011111010100100000000**. There are 22 bits in the binary string. Find the binary string of each of the RED values. Next, hide 1-bit of the binary string of your student number in each RED value. From the left of the binary string of your student number, the first bit should be stored in the *least significant bit (LSB)* of RED value of *Pixel 1*, the second bit should be stored in the LSB of RED value of *Pixel 2*, and so on.]

Table-Q5: List of Selected Pixels

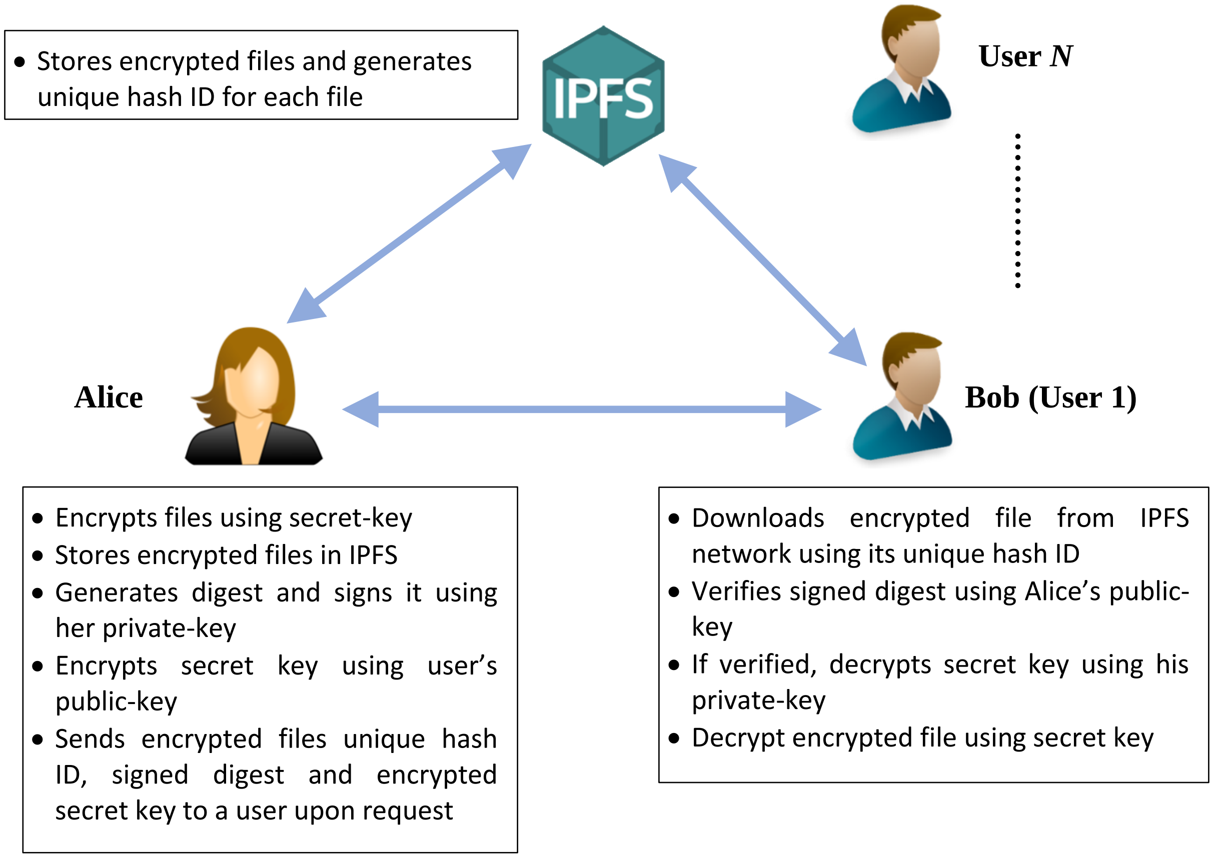
Pixel No.	Pixels
1	[10][50]
2	[20][100]
3	[30][150]
4	[40][200]
5	[50][250]
6	[60][300]
7	[70][350]
8	[80][400]
9	[90][450]
10	[100][500]
11	[110][550]
12	[120][600]
13	[130][650]
14	[140][700]
15	[150][750]
16	[160][800]
17	[170][850]
18	[180][900]
19	[190][950]
20	[200][1000]
21	[210][1050]
22	[220][1100]

Q6 (OpenSSL and IPFS) [3 Marks]

Assume that Alice has an IPFS-based repository of encrypted files. Alice encrypts each file with a secret key using OpenSSL and stores it to IPFS network. Alice uses AES algorithm as symmetric key encryption. Each encrypted file is given a unique hash ID by the IPFS network. Alice allows access to those files only to the authorized users.

Assume that Bob is an authorized user in the IPFS network. Whenever Bob wants to access a file, Alice generates the message digest of that encrypted file using SHA-256 hash algorithm implementation of OpenSSL. Next, she signs the message digest with her RSA private-key using OpenSSL. Further, Alice encrypts the AES secret-key (that is used during file encryption) with the Bob's RSA public-key using OpenSS. Finally, she sends the unique hash ID of the encrypted file, signed message digest and encrypted secret key to Bob through email.

Upon receiving them, Bob downloads the encrypted file from IPFS network using the unique hash ID. Bob generates the message digest using his RSA public-key and SHA-256 algorithm to verify the integrity of the encrypted file. If the verification is successful, then Bob decrypts the shared secret-key using his RSA private-key. Finally, Bob decrypts the encrypted file using the secret key. For the verification of the encrypted file and decryption, Bob uses OpenSSL. The scenario is illustrated in the **Figure-Q6** below.
Demonstrate each step stated above during tutorials within **week-10 to 12.**



- Stores encrypted files and generates unique hash ID for each file

- Encrypts files using secret-key
- Stores encrypted files in IPFS
- Generates digest and signs it using her private-key
- Encrypts secret key using user's public-key
- Sends encrypted files unique hash ID, signed digest and encrypted secret key to a user upon request

- Downloads encrypted file from IPFS network using its unique hash ID
- Verifies signed digest using Alice's public-key
- If verified, decrypts secret key using his private-key
- Decrypt encrypted file using secret key

Figure-Q6: IPFS based encrypted file storage