

FIT3031
Information & Network Security
Assignment 1 – Semester-B 2018

Submission Guidelines

- **Deadline:** Assignment 1 due on Monday 15th January 2018, 4:00 PM
- **Submission format:** PDF only. You can use any freely available PDF converter to make PDF file from editable one.
- **Submission platform:**
 - Clayton - Softcopy submission on **Moodle**.
- PLEASE INCLUDE **YOUR NAME** AND **SUID** WITHIN THE MAIN PDF SUBMISSION
- Files to submit: *Assign-1_FirstName_LastName_SUID.pdf*
- **No late submissions ONLY** via [special consideration request](#)
- **Late submissions:** An assignment handed in late without prior permission will receive a late penalty of a 5% deduction per day (including Saturday and Sunday) or part thereof, after the due date and time.
- **Plagiarism:** *It is an academic requirement that your submitted work be original. Zero marks will be awarded for the whole submission if there is any evidence of copying, collaboration, pasting from websites, or copying from textbooks. –Plagiarism policy applies to all assessments*

Marks

- *This assignment is worth **20%** of the total unit marks.*
- *The assignment is marked out of **60** nominal marks.*
- *For example, if you obtain **30** marks for this assignment, it will contribute*
- ***(30/60)*20=10 marks** to you final unit grade.*

Assignment Questions

1. Question-1: [2+2+2+2+2] = 10 marks

- a. We consider the security services (A) confidentiality, (B) integrity, (C) authenticity, and (D) non-repudiation, for a variety of simple protocols. The input is always the plaintext x . y is the packet sent from Alice to Bob. Describe which security services are achieved by the following protocols.

1. $y = [h(x), x]$, where $h(x)$ denotes a (collision resistant etc.) hash function.

(2 marks)

If the packet is **not** exposed to any form of passive or active attack then this protocol will provide only integrity of the message; Provided the communication is not error prone! Under real network communications scenario this packet will **not provide any security services to the payload.** $y = [h(x), x]$,
No confidentiality as x is unencrypted, no MAC, so No Integrity, No Signature-So No Authentication and also no src non-repudiation

2. $y = [MAC(x), x]$, where $MAC(x)$ denotes a secure message authentication code such as HMAC.

(2 marks)

Under real network communications scenario this packet **will provide integrity security services to the payload;** since the HMAC is secure hashed message authenticated code using shared symmetric keys.

$y = [MAC(x), x]$, **No confidentiality as x is unencrypted, MAC, so Integrity=yes, No Signature-So No Authentication and also no src non-repudiation**

3. $y = [enc_S(h(x)), x]$, where $enc_S()$ denotes a secure stream cipher.

(2 marks)

Under real network communications scenario this packet will **provide integrity security services** to the payload; since the $enc_S(h(x))$, is secure stream cipher of the hash of the message using a pseudorandom key stream.

$y = [enc_S(h(x)), x]$, **No confidentiality as x is unencrypted, MAC, so Integrity=yes, No Signature-So No Authentication and also no src non-repudiation**

4. $y = \text{encB}(x, h(x))$, where $\text{encB}()$ denotes a secure block cipher.

(2 marks)

Under real network communications scenario this packet will provide confidentiality and integrity, security services to the payload; since the $\text{encB}(x, h(x))$, is secure block cipher of the hash of the message along with the message using one of the block cipher scheme. E.g. ECB etc...

$y = \text{encB}(x, h(x))$, Confidentiality as x is encrypted, along with hash so Integrity=yes, No Signature-So No Authentication and also no src non-repudiation

5. $y = \text{encS}(x, \text{sig}(x))$, where x is the last block of a long message x ; $\text{encS}()$ denotes a secure stream cipher and $\text{sig}(x)$ is the signature of message x .

(2 marks)

Under real network communications scenario this packet will provide confidentiality, integrity, authenticity, and source non-repudiation security services to the payload; since the $\text{encB}(x, h(x))$, is secure block cipher of the hash of the message along with the message using one of the block cipher scheme. E.g. ECB etc...

Confidentiality as x is encrypted, along with Sig so Integrity=yes, Signature-So it provides Authentication and also src non-repudiation

2. Question-2: [3+3+2 = 8 Marks]

Assuming you can do 2^{30} encryptions per second and **key size is 64 bits**:

- How long would a brute force attack take? (Both maximum and average values)
- Give a scenario where this would be practical and another where it wouldn't.
- What happens if you double the key size?

Show the working process of your work in few steps.

Question 2-Ans:

- I. Key Size = 64 bit \rightarrow key space = 2^{64}

Total time = 2^{64} (1 key scheduling + 1 encryption) \div (2^{30} encryptions per sec)

Since key scheduling time is not given, we assume the key space has been pre-generated.

Thus,

Maximum time (worst case):

2^{64} encryptions \div 2^{30} encryptions per sec

$= 2^{(64 - 30)} = 2^{34}$ seconds

≈ 544.8 years

Average time:

$= 2^{(64 - 30)} \div 2 = 2^{33}$ seconds

≈ 272.4 years

[3 marks]

- II. If the time taken to crack the key is shorter than the useful life of the information, or the benefit gained from cracking the key is greater than the effort and resource required to crack it (E.g. fast computer, small key space), brute-force attack is feasible. The students should

give one Scenario for such a brute force attack protection/no-protection
[1.5 marks]

Under two circumstances brute-force attack is infeasible:

1. The effort and resource required to crack the key is greater than the benefit gained from cracking the key

2. The time taken to crack the key is longer than the useful life of the information

[1.5 marks]

III. In terms of exhaustively searching the entire key space, if the key size is doubled:

64-bit $\times 2 = 128$ -bit key \rightarrow key space $= 2^{128}$

Total time $= 2^{128}$ encryptions $\div 2^{30}$ encryptions per sec

$= 2^{(128 - 30)} = 2^{98}$ seconds

$\approx 1.0049234210332869 \times 10^{22}$ years

Which is $2^{(98 - 34)} = 2^{64}$ times of the time required to exhaustively search a 64-bit key space

[2 marks]

3. Question-3: [3+3+3 = 9 Marks]

User **A** and **B** use Diffie-Hellman algorithm to exchange a shared key and generate public keys of their own. Consider a common prime number $q=71$ and a (a primitive root of q) = 7. Determine the followings:

- If user **A** has private key=5, what is **A's** public key?
- If user **B** has private key=12, what is **B's** public key?
- What is the shared key?

Show the working process of your work in at least three steps. Consult lecture notes and the text book.

Question 3-Ans

A. $X_A = 5$

$$Y_A = a^{X_A} \bmod q = 7^5 \bmod 71 = 51$$

[3 marks]

B. $X_B = 12$

$$Y_B = a^{X_B} \bmod q = 7^{12} \bmod 71 = 4$$

[3 marks]

C. $Y_B^{X_A} \bmod p = 4^5 \bmod 71 = 30$

$$Y_A^{X_B} \bmod p = 51^{12} \bmod 71 = 30$$

[3 marks]

Students should show their working towards the answer:

1 marks for correct formula,

1 mark for showing the calculations,

1 mark for correct result

4. Question-4: [2+3+3 = 8 Marks]

Alice and Bob agree to communicate privately via email using a scheme based on RC4, but they want to avoid using a new secret key for each transmission. Alice and Bob privately agree on a 128-bit key k . To encrypt a message m consisting of a string of bits, the following procedure is used:

- Choose a random 80-bit value v
- Generate the ciphertext $c = RC4(v \parallel k) \oplus m$
- Send the bit string $(v \parallel c)$

Answer the following:

- Suppose Alice uses this procedure to send a message to Bob. Describe how Bob can recover the message m from $(v \parallel c)$ using k .
- If an adversary observes several values of $(v_1 \parallel c_1), (v_2 \parallel c_2), \dots$. Transmitted between Alice and Bob, how can he/she determine when the same key stream has been used to encrypt two messages?
- If Alice and Bob agree to use 16-bit Cipher Feedback (CFB) mode instead of RC4, and a bit error occurs in the transmission of a ciphertext; how far does the error propagate?

[2+3+3 = 8 Marks]

Question 5-Ans

- A. Since Bob already has the shared key k , after receiving Alice's message $(v \parallel c)$, Bob can extract v from $(v \parallel c)$ and compute:

$$\begin{aligned} & RC4(v \parallel k) \oplus c \\ &= RC4(v \parallel k) \oplus (RC4(v \parallel k) \oplus m) \\ &= RC4(v \parallel k) \oplus RC4(v \parallel k) \oplus m \\ &= 0 \oplus m \\ &= m \end{aligned}$$

[2 mark]

- B. Here we know k is constant, it's only the vector v that makes the key stream $RC4(v \parallel k)$ different.

Therefore, if we observe two message M_1 and M_2 such that

$$M_1 = (v_1 \parallel c_1) \text{ and } M_2 = (v_1 \parallel c_2),$$

since the same vector v_1 appears in both M_1 and M_2 , the key stream used to compute c_1 and c_2 must only be $RC4(v_1 \parallel k)$.

Therefore, we can deduce that the same key stream is used to encrypt two messages.

[3 marks]

- C. Block size b is 16-bit

AES block size is 128-bit, so shift register size s should be the same

For CFB, if one block of transmitted ciphertext is corrupted, $(1 + b/s)$ blocks of plaintext blocks will be corrupted, in which the first block is directly affected after the XOR operation, then the error propagates (b/s) blocks because the shift register contains the corrupted cipher text block for (b/s) shifts.

Therefore, the error propagates $(128 / 16) = 8$ blocks after the first corrupted block (Students may write 9 blocks in total as answer, which is also acceptable)

[3 marks]

5. Question-5: [2+2 = 4 Marks]

- a. Is it possible to perform encryption operations in parallel on multiple blocks of plaintext in CBC mode? **Justify your answer.**
- b. How about decryption? **Justify your answer.** **[2+2 = 4 Marks]**

Question 5-Ans

- In CBC encryption, the input block to each forward cipher operation (except the first) depends on the result of the previous forward cipher operation, so the forward cipher operations cannot be performed in parallel.
- In CBC decryption, however, the input blocks for the inverse cipher function (i.e., the ciphertext blocks) are immediately available, so that multiple inverse cipher operations can be performed in parallel.
- The students have to give some reason and not a simple yes/no answer to get full marks, otherwise deduct 0.5 mark for each part.

6. Question-6: [2+2+2 = 6 Marks]

Consider $p=17$ and $q=31$ in the RSA encryption/decryption algorithm to be used to encrypt a message $M=15$. Using the algorithm, determine the followings:

- a. Generate a public-private key pair.
- b. Using the generated public key, encrypt the message to get the cipher text C .
- c. Apply the private key on C to decrypt the original message M .

Show the working process of your work in at least three steps in both encryption and decryption. **[2+2+2 = 6 Marks]**

Question 6-Ans

- Select two prime numbers, in this case $p=17$, $q=31$
 Calculate $n = pq = 17 \times 31 = 527$
 Calculate Euler Totient:
 $\phi(n) = (p-1)(q-1)$
 $= (17-1)(31-1) = 480$
 Select e such that e is relatively prime to $\phi(n) = 480$,

$$\text{GCD}(\phi(n), e) = 1 \quad (1)$$

$$1 < e < \text{Euler_tolerant}(n) \quad (2)$$

The $e = 7$ is selected because $\text{GCD}(480, 7) = 1$

Determine d such that $de \bmod 480 = 1$ and $d < 480$

$$(d \times 7) \bmod 480 = 1$$

attempting with different values of we get

$$343 \times 7 \bmod 480 = 2401 \bmod 480 = 1$$

therefore $d = 343$

So, public key $KU = \{e, n\} = \{7, 527\}$

and private key $KR = \{d, n\} = \{343, 527\}$

The students have to show at least few steps of calculations to get full marks, otherwise deduct 1 mark.

- $C = M^7 \bmod 527$

$$C = 15^7 \bmod 527$$

$$C = 178$$

Exploiting the properties of modular arithmetic we get

$$15^7 \bmod 527 = [(15^1 \bmod 527) \times (15^2 \bmod 527) \times (15^4 \bmod 527)]$$

$$(15^1 \bmod 527) = 15$$

$$(15^2 \bmod 527) = 225$$

$$(15^4 \bmod 527) = 33$$

So $15^7 \bmod 527 = (15 \times 225 \times 33) \bmod 527 = 380070 \bmod 527 = 178$ therefore encrypted message is 178

The students have to show at least few steps of calculations to get full marks, otherwise deduct 0.5 mark.

- $C = 178$

$$M = C^d \bmod n$$

$$M = 178^{343} \bmod 527$$

Exploiting the properties of modular arithmetic we get

$$178^{343} \bmod 527 = [(178^1 \bmod 527) * (178^2 \bmod 527) * (178^4 \bmod 527) * (178^{16} \bmod 527) * (178^{64} \bmod 527) * (178^{256} \bmod 527)] \bmod 527$$

$$\begin{aligned} \text{As, } (178^1 \bmod 527) &= 178 \\ (178^2 \bmod 527) &= 64 \\ (178^4 \bmod 527) &= 407 \\ (178^{16} \bmod 527) &= (407 * 407 * 407 * 407) \bmod 527 = 256 \\ (178^{64} \bmod 527) &= (256 * 256 * 256 * 256) \bmod 527 = 35 \\ (178^{256} \bmod 527) &= (35 * 35 * 35 * 35) \bmod 527 = 256 \end{aligned}$$

$$\begin{aligned} \text{So } M &= (178 * 64 * 407 * 256 * 35 * 256) \bmod 527 \\ &= [(178 * 64 * 407) \bmod 527 * (256 * 35 * 256) \bmod 527] \bmod 527 \\ &= [525 * 256] \bmod 527 \\ &= 15 \end{aligned}$$

The students have to show at least few steps of calculations to get full marks, otherwise deduct 0.5 mark.

7. Question-7: [2+2+2 = 6 Marks]

In this problem we shall compare the security services that are provided by digital signatures (DS) and message authentication codes (MAC). We assume that Oscar is able to observe all messages sent from Alice to Bob and vice versa. Oscar has no knowledge of any keys but the public one in case of DS. State whether and how DS and MAC protect against each attack. The value *auth* (*x*) is computed with a DS or a MAC algorithm, respectively.

- (Message Integrity) Alice sends a message *x* = “*Transfer \$1000 to Mark*” in the clear and also sends *auth* (*x*) to Bob. Oscar intercepts the message and replaces “*Mark*” with “*Oscar*”. Will Bob detect this?
- (Replay) Alice sends a message *x* = “*Transfer \$1000 to Oscar*” in the clear and also sends *auth* (*x*) to Bob. Oscar observes the message and signature and sends them 100 times to Bob. Will Bob detect this?
- (Authentication with Bob cheating) Bob claims that he received a message *x* with a valid signature *auth* (*x*) from Alice (e.g., “*Transfer \$1000 from Alice to Bob*”) but Alice claims she has never sent it. Can Alice clear this question **in either case**?

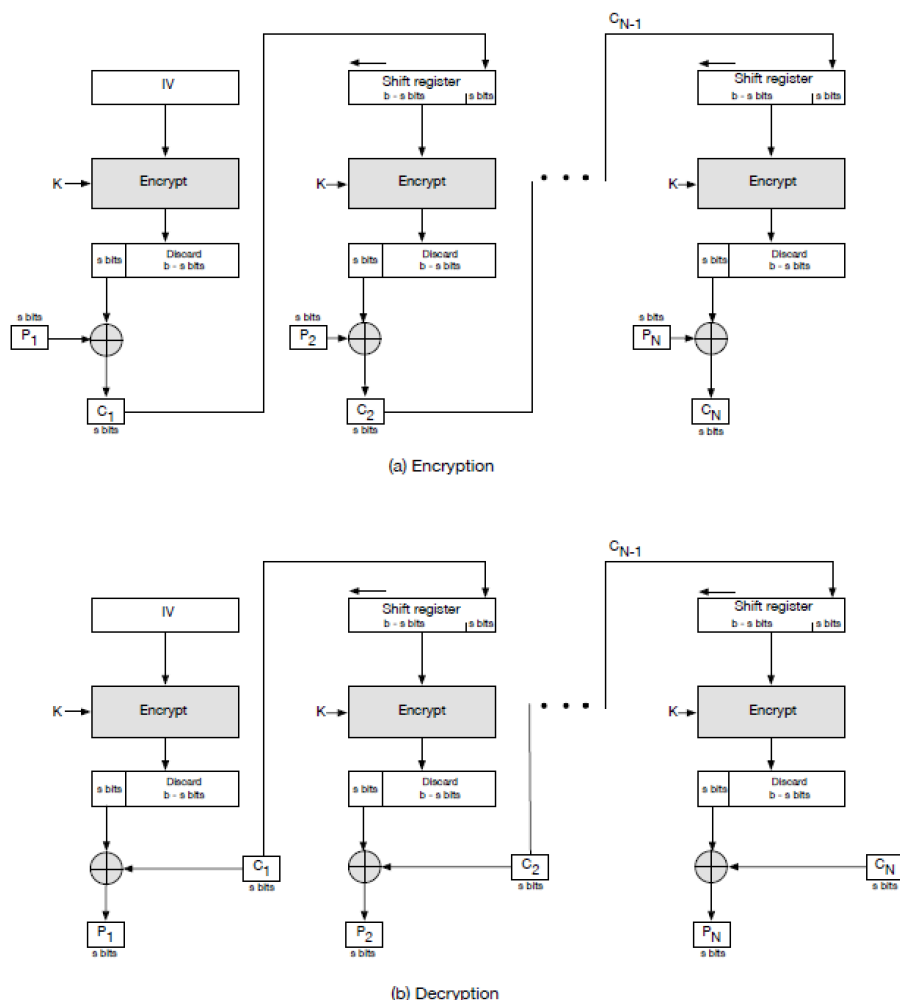
[2+2+2 = 6 Marks]

Question 7-Ans

- Will be detected with both (i) DS and (ii) MAC
- Won't be detected by either DS or MAC. Unless the message includes timestamps. (Hint is use Timestamps if you want to detect it!)
- (i) DS: Alice has to force Bob to prove his claim by sending her a copy of the message in question with the signature that was sent earlier. Then Alice can show that message and signature can be verified with Bob's public key. Bob must have generated the message.

8. Question-8:[3+2 + 2+2] =9 marks

Alice is using CFB mode of operation to encrypt a 16KB file to send it to Bob (1KB=1024 bytes).



Referring to the above figure of CFB, answer the following questions:-

- How many ciphertext blocks will be produced if Alice uses 8-bit CFB-DES (DES or Data Encryption Standard algorithm as the block cipher in Figure above)?
- How many ciphertext blocks will be produced if Alice uses 16-bit CFB-AES (AES or Advanced Encryption Standard algorithm as the block cipher in Figure above)?
- If there is an error in transmitted ciphertext block C_1 , how many plaintext blocks will be corrupted when Bob decrypts the file for 8-bit CFB-DES?
- If there is an error in transmitted ciphertext block C_1 , how many corrupted plaintext blocks when 16-bit CFB-AES is used?

b : the block size of the block cipher algorithm
 s : number of bits used in CFB mode of operation

- for 8-bit CFB the number of bits per blocks is 8 hence total number of blocks = $\frac{\text{file size in bits}}{\text{8bits per block}} = \frac{16 \times 1024 \times 8}{8} = 16384$, the block size of the block cipher has no effect on the number of ciphertext blocks.
- for 16-bit CFB as the bit size is doubled the block numbers will be halved so: 8192
- using the formula: 1 (by XOR) + $\frac{b}{s}$ (by shift register) = $1 + \frac{64}{8} = 9$, so P_1 to P_9 , since C_2, C_3, \dots, C_9 are all received correctly when C_1 is shifted out the plaintexts will be recovered correctly onward.
- using the formula: 1 (by XOR) + $\frac{b}{s}$ (by shift register) = $1 + \frac{128}{16} = 9$ so P_1 to P_9
 (Note: even though the number of blocks are the same the extent of error propagation is twice in terms of affected bits compared to 8-bit CFB-DES as the affected plaintext blocks are 16-bit each)