Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

SEMESTER 1 SOLUTIONS 2022/2023

MODULE: CA4005 - Cryptography and Security Protocols

PROGRAMME(S):

CASE	B.Sc. in Computer Applications (Sft.Eng.)
ECSAO	Study Abroad (Engineering & Computing)
ECSA	Study Abroad (Engineering & Computing)

YEAR OF STUDY: 4,O,X

EXAMINER(S):

Geoffrey Hamilton	(Internal)	(Ext:5017)
Prof. Arend Rensink	(External)	External

TIME ALLOWED: 3 Hours

INSTRUCTIONS: Answer all questions. All questions carry equal marks.

PLEASE DO NOT TURN OVER THIS PAGE UNTIL INSTRUCTED TO DO SO.
The use of programmable or text storing calculators is expressly forbidden.

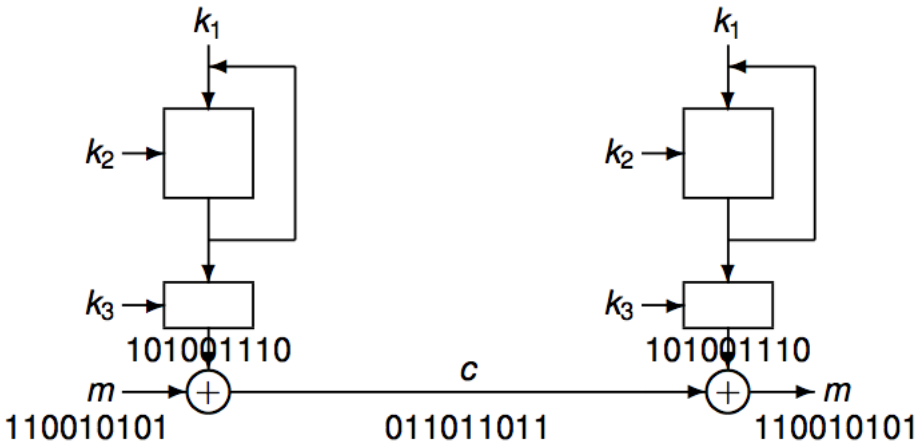
There are no additional requirements for this paper.

QUESTION 1 ***[Total marks: 20]***

1(a) [6 Marks]

Describe the general operation of *stream ciphers*, describing in particular how they perform encryption and decryption (using a diagram if necessary). What properties must the *keystream* have for the stream cipher to be considered secure?

Solution:
Stream ciphers encrypt a data stream one bit or byte at a time. The sender generates a pseudo-random keystream from a secret key which is XOR-ed with the data stream to produce the corresponding cipher stream. The receiver generates the same keystream from the same secret key, and this is XOR-ed with the cipher stream to recover the original data:



For the stream cipher to be considered secure, the keystream must:

- Look random i.e. pass pseudo-random tests.
- Be unpredictable i.e. have a long period.
- Have large linear complexity.
- Have low correlation between key bits and keystream bits.

1(b) [7 Marks]

Describe the *Cipher Block Chaining* (CBC) mode of operation for block ciphers. What is the role of the *Initialisation Vector* (IV)? What are the dangers if an IV is:

- altered by an attacker
- known to an attacker
- reused with the same key

Solution:
In CBC mode, the plaintext m is divided into t blocks of n bits $m_1, m_2, \ldots, m_t$ (the last block is padded if necessary).

Encryption:

- $c_1 = e_k(m_1 \oplus IV)$.

- $c_i = e_k(m_i \oplus c_{i-1})$ for $i > 1$.

Decryption:

- $m_1 = d_k(c_1) \oplus IV$.
- $m_i = d_k(c_i) \oplus c_{i-1}$ for $i > 1$.

The IV is used to give an initial value for the first block; this should be different for different messages to hide patterns and repetitions. Someone tampering with the IV could tamper with the resulting plaintext on decryption. There is no problem if the IV is known to an attacker, but if an IV is reused with the same key, patterns and repetitions within the underlying messages can be revealed.

1(c) [7 Marks]

Compare and contrast the *Output Feed Back* (OFB) and *Cipher Feed Back* (CFB) modes of operation for block ciphers with respect to the following:

- Encryption
- Decryption
- Error propagation

Solution:
In OFB mode, the keystream is generated by first encrypting the IV, selecting some bits of this for the keystream, and then feeding back the output of the encryption for further processing in the same way. The ciphertext is obtained from the exclusive-or of the keystream with the plaintext, and the plaintext can be recovered from the ciphertext by exclusive-or with the keystream. In CFB mode, the keystream is generated by first encrypting the IV, selecting some bits of this for the keystream, obtaining the corresponding ciphertext from the exclusive-or of this keystream with the plaintext, and then feeding back this ciphertext for further processing in the same way. The plaintext is also recovered from the ciphertext by exclusive-or with the keystream. In OFB mode, errors are only copied and none are propagated. In CFB mode, errors are propagated over $\lceil n/j \rceil + 1$ blocks where n is the input block size and j is the output block size.

[End of Question 1]

QUESTION 2 *[Total marks: 20]*

2(a) [6 Marks]

A cryptographically secure hash function should be *pre-image resistant* and *collision-free*. Define these properties and why they are important for hash functions.

Solution:
A hash function is pre-image resistant if it is computationally infeasible to recover data from its digest. This is important because the original data should remain secure.

A hash function is weakly collision-free or second pre-image resistant if, given M, it is computationally infeasible to find a different M' such that $H(M) = H(M')$. It is strongly collision-free if it is computationally infeasible to find different messages M and M' such that $H(M) = H(M')$. This is important because being able to find collisions relatively easily allows an attacker to replace one message with another which they have found to have the same digest.

2(b) [7 Marks]

Describe how hash functions can be used to implement digital signatures. Explain why it is important in this context that a hash function has a digest size of at least 160 bits. Describe a simple fraud that could be perpetrated using digital signatures if the hash function digest size were only 64 bits.

Solution:
Rather than signing the original message, which may be computationally expensive, a hash function can be used to produce the message digest, which can then be signed. The receiver of the message with digest should apply the same hash function to the message to verify that the same digest is produced.

For a n-bit digest, on average $\sqrt{n}$ trials are required before a collision occurs, which we would like to avoid. For a 160-bit digest on average 2^{80} trials would therefore be required before finding a collision, which is computationally infeasible.

For a 64-bit digest, only 2^{32} trials would be required before finding a collision which is computationally feasible. A malicious participant A could therefore generate 2^{32} messages which are acceptable to another participant B and 2^{32} messages which are not acceptable, and find a collision between these two sets. If A sends the acceptable collision message to B, and B signs it, then A could later claim that B had actually signed the unacceptable collision message, since they will both have the same digest. Similarly, B could sign one message and later claim to have signed the other one.

2(c) [7 Marks]

Describe how hash functions can be used for message authentication. How do *Message Authentication Codes* (MACs) differ from *Manipulation Detection Codes* (MDCs)? Describe how a MAC can be constructed from a block cipher, and how a MAC can be constructed from a MDC.

Solution:
By sending the digest of a message along with the message itself, the integrity of the message can be checked by seeing whether applying the same hash function to the message gives the same digest.

A MDC is a hash function without a key, so the MDC of a message has to be sent over an authenticated channel to prevent tampering. A MAC is a hash function with a key, so the MAC of a message does not have to sent over an authenticated channel.

A MAC can be constructed from a block cipher by using CBC mode, with the output of the final block giving the MAC value. A MAC can be constructed from a MDC using a HMAC where $HMAC_k(m) = h(k||p_1||h(k||p_2||m))$, where k is the key, m is the message and p_1, p_2 are fixed strings used to pad k to a full block.

[End of Question 2]

QUESTION 3 ***[Total marks: 20]***

3(a) [7 Marks]

Calculate $6/119 \pmod{191}$.

Solution:
We need to use the extended Euclidean GCD algorithm to calculate this:

$$\begin{aligned} 191 &= 119 + 72 \\ 119 &= 72 + 47 \\ 72 &= 47 + 25 \\ 47 &= 25 + 22 \\ 25 &= 22 + 3 \\ 22 &= (7 \times 3) + 1 \end{aligned}$$

So:

$$\begin{aligned} 72 &= 191 - 119 \\ 47 &= 119 - 72 = 119 - (191 - 119) = (2 \times 119) - 191 \\ 25 &= 72 - 47 = 191 - 119 - (2 \times 119) + 191 = (2 \times 191) - (3 \times 119) \\ 22 &= 47 - 25 = (2 \times 119) - 191 - (2 \times 191) + (3 \times 119) = (5 \times 119) - (3 \times 191) \\ 3 &= 25 - 22 = (2 \times 191) - (3 \times 119) - (5 \times 119) + (3 \times 191) = (5 \times 191) - (8 \times 119) \\ 1 &= 22 - (7 \times 3) = (5 \times 119) - (3 \times 191) - (35 \times 191) + (56 \times 119) = (61 \times 119) - (38 \times 191) \end{aligned}$$

So $119^{-1} \pmod{191} = 61$

$6/119 \pmod{191} = 6 \times 61 \pmod{191} = 175$

3(b) [6 Marks]

Calculate $\phi(24)$, where ϕ is the Euler Totient function. Use this to calculate $27^{615} \pmod{24}$.

Solution:
$\phi(24) = 8$

$27^{615} \pmod{24} = 3^{615 \pmod{\phi(24)}} \pmod{24} = 3^{615 \pmod{8}} \pmod{24} = 3^7 \pmod{24} = 3$

3(c) [7 Marks]

Find a primitive root of $\mathbb{Z}_{17}^*$

Solution:
This has to be done by trial and error. The primitive roots of $\mathbb{Z}_{17}^*$ are 3, 5, 6, 7, 10, 11, 12, 14.

[End of Question 3]

QUESTION 4 ***[Total marks: 20]***

4(a) [6 Marks]

Compare and contrast the RSA cryptosystem with the Rabin cryptosystem.

Solution:
Both cryptosystems use exponentiation to perform encryption. In the Rabin cryptosystem, the encryption exponent is 2 while in RSA the encryption exponent is given by the public key. In the Rabin cryptosystem, decryption is performed by calculating the modular square root of the ciphertext, while in RSA, decryption is performed by also performing exponentiation where the decryption exponent is given by the private key. Thus in the Rabin cryptosystem there are four possible decryptions, while in the RSA cryptosystem there is only one. Both cryptosystems make use of composite numbers which are the product of two large primes and are based on the assumption that it is infeasible to factor such composites. However, breaking the Rabin cryptosystem has been proven to be equivalent to the integer factorisation problem, while breaking the RSA cryptosystem has not.

4(b) [5 Marks]

Consider a toy Rabin cryptosystem in which the public key $N = 77$. Describe how encryption is done in the Rabin cryptosystem and use this to encrypt the message 29.

Solution:
Encryption is performed by squaring the plaintext.

For this example, we calculate $29^2 \pmod{77} = 71$

4(c) [9 Marks]

Describe how decryption is done in the Rabin cryptosystem using the prime factors of the modulus, and how this can be made faster by choosing these prime factors to satisfy specific properties. Use this decryption technique to find all possible decryptions for the ciphertext value 37 when the public key $N = 77$ as above. In practice, how would we determine which of the possible decryptions of a ciphertext is the correct one?

Solution:
Decryption is performed by calculating the modular square root of the ciphertext. Since the person decrypting the ciphertext will have knowledge of the prime factors of the modulus, the square root can be calculating modulo each of these factors, and then combined using the Chinese Remainder Theorem. The prime factors are normally chosen so that they are congruent to 3 modulo 4, thus making the calculation of the square roots modulo these factors easy and fast.

For this example, we calculate:

$\sqrt{37} \pmod{7} = \pm 37^2 \pmod{7} = \pm 2^2 \pmod{7} = \pm 4 \pmod{7}$

$\sqrt{37} \pmod{11} = \pm 37^3 \pmod{11} = \pm 4^3 \pmod{11} = \pm 9 \pmod{11}$

Using the Chinese Remainder Theorem, we obtain the four possible decryptions: 53, 24, 46 and 31.

In practice, we would add redundancy to the plaintext, and check for this same redundancy in the decryption to determine which decryption is the correct one.

[End of Question 4]

QUESTION 5 ***[Total marks: 20]***

Consider the following protocol that allows entities A and B to mutually authenticate each other using a shared secret key $K_{\langle A,B\rangle}$.

1. $A \rightarrow B : A, N_A$
2. $B \rightarrow A : B, N_B, \{N_A\}_{K_{\langle A,B\rangle}}$
3. $A \rightarrow B : \{N_B\}_{K_{\langle A,B\rangle}}$

5(a) [4 Marks]

Explain the use of nonces in this protocol.

Solution:
The nonces are used as challenges to the other participant in the protocol. If the other participant responds to this by encrypting the nonce with the shared secret key, then they provide proof of their knowledge of the shared secret key and thus authenticate themselves.

5(b) [8 Marks]

Describe a *reflection* attack on this protocol.

Solution:
A reflection attack on this protocol in which a dishonest entity E wishes to impersonate A to B is as follows:

- First E starts one instance of the protocol with B.
  1. $E(A) \rightarrow B : A, N_A$
  2. $B \rightarrow E(A) : B, N_B, \{N_A\}_{K_{\langle A,B\rangle}}$
- At this stage, E is stuck as she cannot encrypt N_B.
- However, E can start a second instance of the protocol and get B to encrypt N_B.
  3. $E(A) \rightarrow B : A, N_B$
  4. $B \rightarrow E(A) : B, N_{B'}, \{N_B\}_{K_{\langle A,B\rangle}}$
- E abandons the second instance and resumes the first instance.
  5. $E(A) \rightarrow B : \{N_B\}_{K_{\langle A,B\rangle}}$

5(c) [8 Marks]

Describe two techniques, that do not require the use of additional keys, for preventing the reflection attack described in your previous answer.

Solution:
Two techniques for preventing this attack that do not require the use of additional keys are as follows:

1. The initiator could prove its identity first:
   1. $A \rightarrow B : A$
   2. $B \rightarrow A : B, N_B$
   3. $A \rightarrow B : N_A, \{N_B\}_{K_{\langle A,B\rangle}}$

4. $B \rightarrow A : \{N_A\}_{K_{\langle A,B \rangle}}$

In this case B does not encrypt anything until it has authenticated A.

2. The nature of the challenges produced by A and B could be different, i.e. a challenge encrypted by B could not be confused as a challenge to be encrypted by A.

   For example, this can be achieved by including different information in the ciphertext sent between the parties:

   1. $A \rightarrow B : A, N_A$
   2. $B \rightarrow A : B, N_B, \{N_A, A\}_{K_{\langle A,B \rangle}}$
   3. $A \rightarrow B : \{N_B\}_{K_{\langle A,B \rangle}}$

[End of Question 5]

[END OF EXAM]