Midterm Solutions

CS5285 2024

Question 1 (a)

Use the last 6-digits of your student number as a one-time pad to encrypt the plaintext message of the same length 0xA5285B. Provide answer as hexadecimal sequence. As an example:

0x indicates a hexadecimal number, for example 0x0 = 0 (decimal) = 0000 (binary) 0x1A2B would indicate the binary sequence 0001 1010 0011 1011

Use this table to map a hexadecimal value to decimal and binary. **Show all your calculations.** (6 marks)

0x0 = 0 = 0000
0x1 = 1 = 0001
0x2 = 2 = 0010
0x3 = 3 = 0011
0x4 = 4 = 0100
0x5 = 5 = 0101
0x6 = 6 = 0110
0x7 = 7 = 0111

0x8 = 8 = 1000
0x9 = 9 = 1001
0xA = 10 = 1010
0xB = 11 = 1011
0xC = 12 = 1100
0xD = 13 = 1101
0xE = 14 = 1110
0xF = 15 = 1111

Question 1 (a) Solution

Example computation is shown below but will be different for each student. If student number is 12345678, last digit = 345678, answer is 0x917e23

$$\begin{array}{r} 1010\ 0101\ 0010\ 1000\ 0101\ 1011 \\ \oplus\ \ 0011\ 0100\ 0101\ 0110\ 0111\ 1000 \\ \hline 1001\ 0001\ 0111\ 1110\ 0010\ 0011 \end{array}$$

Convert student number and message to binary, XOR to find ciphertext, convert back. (6).

Question 1 (b)

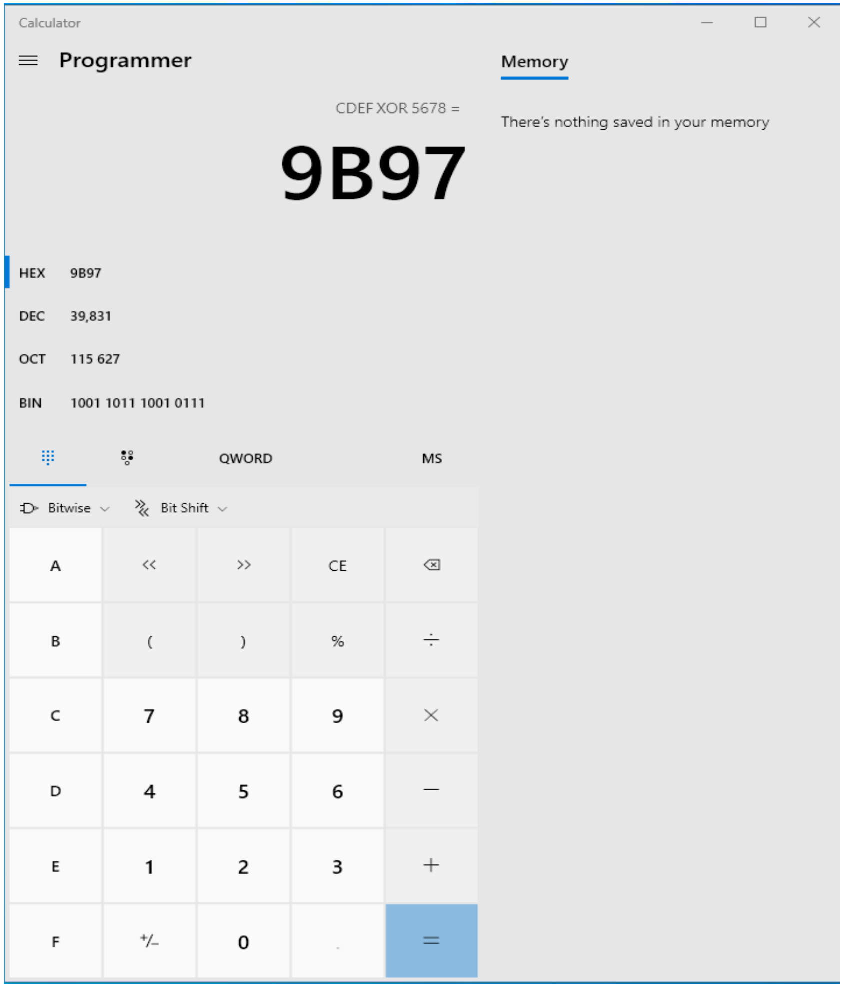
You intercept a ciphertext A5285B, and you know the middle part of the plaintext ??BB?? (? indicates the plaintext character position you do not know). Show how you would modify the ciphertext so that the known plaintext part will decrypt to 0x77 instead of 0xBB. Show all calculations (4 marks)

Solution

Convert 28 and BB to binary, XOR to find keystream 93. XOR 93 to 77 to nd E4 (3). New ciphertext is A5E45B (1).

“Programmer” Calculator

If calculating binary/hex for any purpose makes it easier



```
CDEF XOR 5678 =
9B97

HEX   9B97
DEC   39,831
OCT   115 627
BIN   1001 1011 1001 0111
```

There's nothing saved in your memory

Question 2(a)

The encryption equations for two block cipher modes are provided:

Mode of Operation 1:

$C_0 = ENC_K(IV) \oplus P_0$

$C_1 = ENC_K(C_0) \oplus P_1$

$C_2 = ENC_K(C_1) \oplus P_2$

Mode of Operation 2:

$C_0 = ENC_K(IV) \oplus P_0$

$C_1 = ENC_K(IV+1) \oplus P_1$

$C_2 = ENC_K(IV+2) \oplus P_2$

Give the decryption equations of both modes for P_0, P_1, P_2. Clearly show if you use block cipher as Encrypt (ENC) or Decrypt (DEC).(4 marks)

Solution: The decryption is as follows:

Mode of Operation 1:

$P_0 = ENC_K(IV) \oplus C_0$

$P_1 = ENC_K(C_0) \oplus C_1$

$P_2 = ENC_K(C_1) \oplus C_2$

Mode of Operation 2:

$P_0 = ENC_K(IV) \oplus C_0$

$P_1 = ENC_K(IV+1) \oplus C_1$

$P_2 = ENC_K(IV+2) \oplus C_2$

Question 2 (b)

During transmission one bit of C_0 is changed. In which mode is the error propagation less? Briefly explain the error propagation in each mode.(3 marks)

Solution

Mode 2 (1), Mode 1 has two block error (1), Mode 2 has one block error (1).

$P_1 = \mathrm{ENC}_K(IV\ ) \oplus C_0$

$P_2 = \mathrm{ENC}_K(C_0\ ) \oplus C_1$

$P_3 = \mathrm{ENC}_K(C_1\ ) \oplus C_2$

$P_0 = \mathrm{ENC}_K(IV) \oplus C_0$

$P_1 = \mathrm{ENC}_K(IV+1) \oplus C_1$

$P_2 = \mathrm{ENC}_K(IV+2) \oplus C_2$

Question 2 (c)

During transmission the entire block C_0 is lost (the receiver does not know it has been lost and thinks C_1 is C_0). All subsequent ciphertext blocks are received. In which mode is the error propagation less? Briefly explain the error propagation in each mode.(3 marks)

Solution

Mode 1 (1), Mode 1 will have missing block and 1 block error (1), Mode 2 will have missing block and then everything error (since desynchronised with sender). (1)

Apart from the one missing block, the rest of the message

$P_1 = \mathrm{ENC}_K(IV) \oplus C_1$

$P_2 = \mathrm{ENC}_K(C_1) \oplus C_2$

$P_3 = \mathrm{ENC}_K(C_2) \oplus C_3$

$P_0 = \mathrm{ENC}_K(IV) \oplus C_1$

$P_1 = \mathrm{ENC}_K(IV+1) \oplus C_2$

$P_2 = \mathrm{ENC}_K(IV+2) \oplus C_3$

Question 2 (d)

You need to encrypt customer records that are then stored on a server. You regularly need to access and update these records. Which of these modes would be more efficient to use? Briefly explain why. (3 marks)

Solution

Mode 2 (1), Mode 1 would need all subsequent entries to be re-encrypted if data is updated (1), Mode 2 just need to modify the ciphertext blocks where the plaintext is changed.

Question 2 (e)

You build a system where a hardware device is used to collect sensor data in a factory. To control the factory operation, when the sensor data is taken the device has to encrypt and upload it within 0.1 second. Sensor data is taken every 1 second, and it is three plaintext blocks long. Your device can do an XOR in 0.01 seconds, and has a block cipher that takes 0.065 seconds to encrypt one block. Can you can use one of the modes of operation in 2a? Explain your answer.(3 marks)

Solution

Mode 2 (1), you can pre-compute ENCK(IV), ENCK(IV +1) and ENCK(IV +2) and then just XOR to the three plaintext blocks once sensor reader is taken. This takes only 0.03 seconds. (2)

Question 3 (a)

Given p=13, q=23 and e=19. Write down the equation for signing message m=1234 using RSA**. Show all your calculations** (10 marks)

Solution

$n = p.q$ = 13.23= 299 , $phi(n) = (p\text{-}1)(q\text{-}1)$ = 264 (2)

$d = e^{-1}$ *mod phi(n)*, so do Extended Euclidean algorithm

264=13.19+17

19=1.17+2

17=8.2+1

1=17-8.2=17-8(19-1.17)=9.17-8.19

9.17-8.19 = 9.(264-13.19)-8.19 =9.264-125.17

Apply *mod* 296 both sides 1 *mod* 296 = −125.17 *mod* 264 so d = 139 (5)

s = M^{d} mod n so s = 1234^{139} mod 299 (3)

Only write equation, not required to solve

Question 3 (b)i

You have an ElGamal encryption system with public key y = 7, p = 11 and g = 2? i) Calculate the private key x. (3 marks)

Solution

y = g^x mod p, need to try values between 1 and 10 for x until answer is 7 (x=7) (3).

Question 3 (b)ii+iii

ii) Briefly explain the difficult problem you need to solve to do this. [2]

iii) Why are you able to solve this problem described in 3.b.ii. [1]

Solution

ii) This is the discrete logarithm problem, for $a = b^c \bmod d$, a is easy to calculate with b,c and n. c is difficult to calculate with a, b and n (2).(2).

Iii) The numbers are small, if the numbers were larger it would take longer. (1)

Question 4 (a)

What mechanism can be used for non-repudiation? Explain why it provides non-repudiation.(2 mark)

- Solution

Digital signature, only one person with private key can sign (2)

Question 4 (b)

Would you be able to use one of the block cipher modes in Q2a to construct a MAC in a way similar to CBC-MAC? Briefly explain your answer.(3 marks)

Solution

No, a MAC is sent together with the plaintext. Both these modes use XOR so the final block (the MAC value) C_x would be keystream XOR to P_x. It would therefore be easy to change the last block to be correct for any modification made to the last plaintext block by recovering the keystream and XORing the new value to make a new MAC value (3).

Question 4 (c)

For the crypto system in Q3a, what is the largest message we could securely sign? In practice, how would we ensure that we could sign a message of any length?(2 marks)

Solution

We cannot sign message larger than n (can say 299) (1). We should hash the message and then sign the hash result (1).

Question 4 (d)

Alice wants to send the data D to Bob and also provide data origin authentication of D so that it cannot be modied. Alice sends Bob only the message MAC_K_{AB} (M). Does this message achieve what Alice wanted to do? (2 marks)

- Solution

No, since *M* is never sent and Bob does not have *M* (so cannot verify or get M from MAC_K_{AB} (M). (2).

Question 5

a) AES has a 192-bit block size.

Solution: False. 128-bit block size.

b) Perfect secrecy means that encryption method is secure because attackers cannot practically find the plaintext with the computing resources they have available today.

Solution: False. It is secure even if they have infinite resources (this describes computationally secure).

c) Asymmetric cryptography can be used to provide confidentiality, data origin authentication and non-repudiation.

Solution: True. Encrypt, signature (non-repudation and integrity).

Question 5

d) In Bitcoin, miners are essentially hashing a random number *r* until *h(r)* is equal to a set answer provided by the system. This is most closely related to breaking collision resistance of the hash function.

Solution: False, it is more comparable to wither one-wayness/second pre-image resistance (either OK), given a result h(x), find another value hash to same value.

e) DES key brute force search requires on average 2^{63} guesses.

Solution: False. DES has 56-bit key, effort estimate is $2^{N-1} = 2^{55}$.