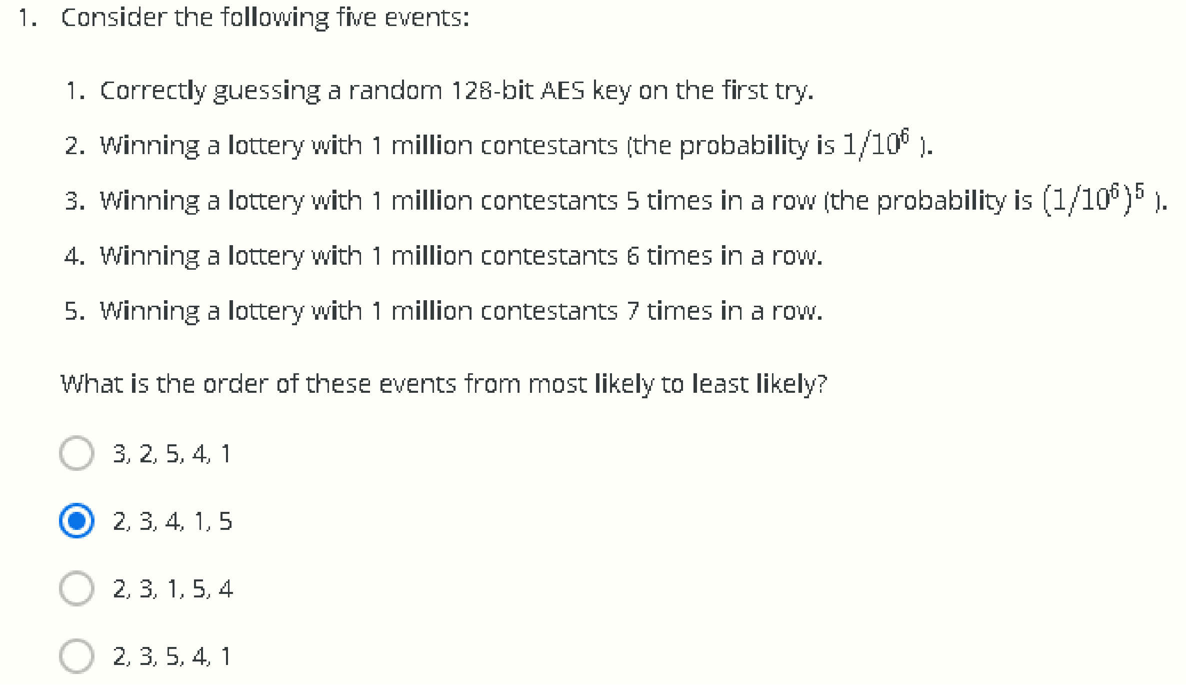
W2 Problem Set && Programming Assignment

Q1



1. Consider the following five events:

    1. Correctly guessing a random 128-bit AES key on the first try.
    2. Winning a lottery with 1 million contestants (the probability is $1/10^6$).
    3. Winning a lottery with 1 million contestants 5 times in a row (the probability is $(1/10^6)^5$).
    4. Winning a lottery with 1 million contestants 6 times in a row.
    5. Winning a lottery with 1 million contestants 7 times in a row.

    What is the order of these events from most likely to least likely?

    ○ 3, 2, 5, 4, 1

    ● 2, 3, 4, 1, 5

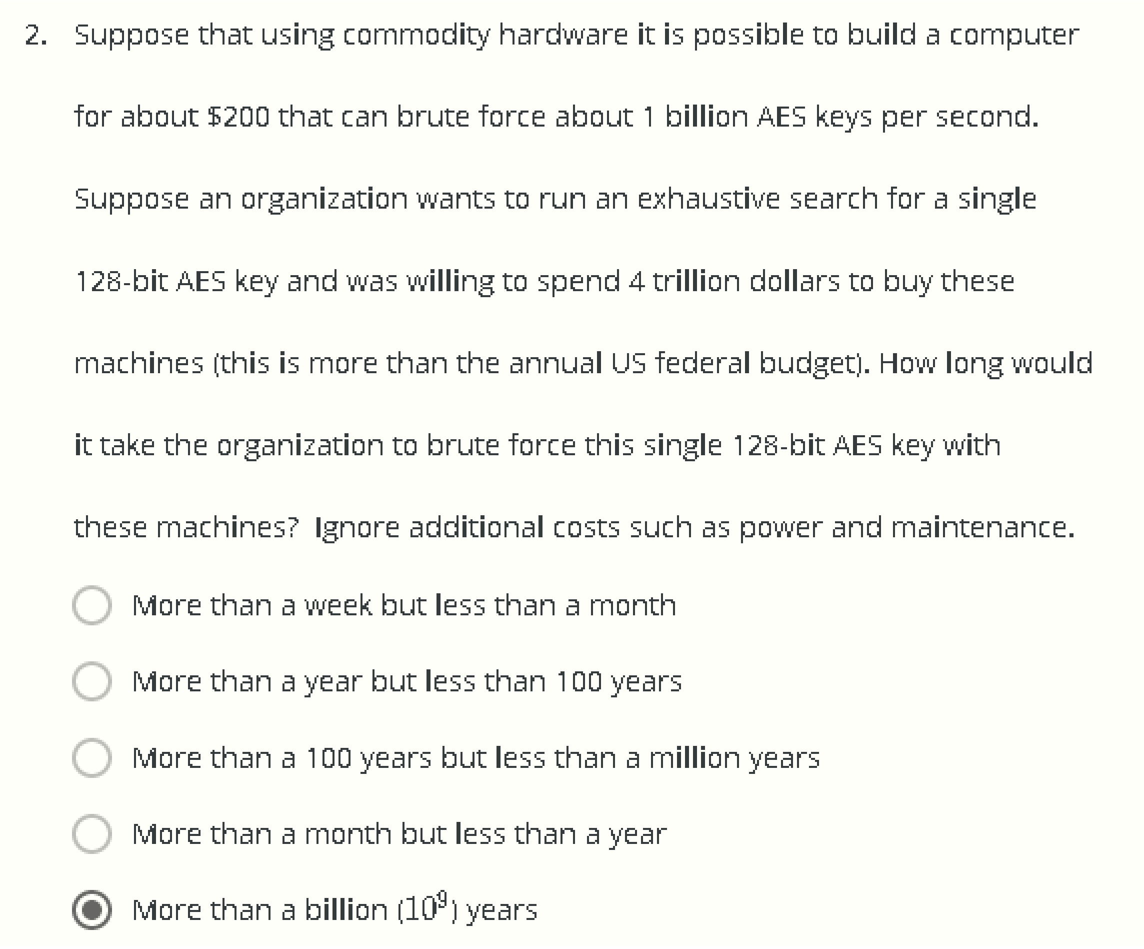
    ○ 2, 3, 1, 5, 4

    ○ 2, 3, 5, 4, 1

问：上述五个事件中，按发生的概率从大到小排序，正确的顺序为？

分析：事件1的概率为$1/2^{128}$，事件4的概率为$1/10^{36}\approx 1/2^{119.5}$，事件5的概率为$1/10^{42}\approx 1/2^{139}$，因此5的概率要比1小更多

Q2



2. Suppose that using commodity hardware it is possible to build a computer for about $200 that can brute force about 1 billion AES keys per second. Suppose an organization wants to run an exhaustive search for a single 128-bit AES key and was willing to spend 4 trillion dollars to buy these machines (this is more than the annual US federal budget). How long would it take the organization to brute force this single 128-bit AES key with these machines? Ignore additional costs such as power and maintenance.

○ More than a week but less than a month

○ More than a year but less than 100 years

○ More than a 100 years but less than a million years

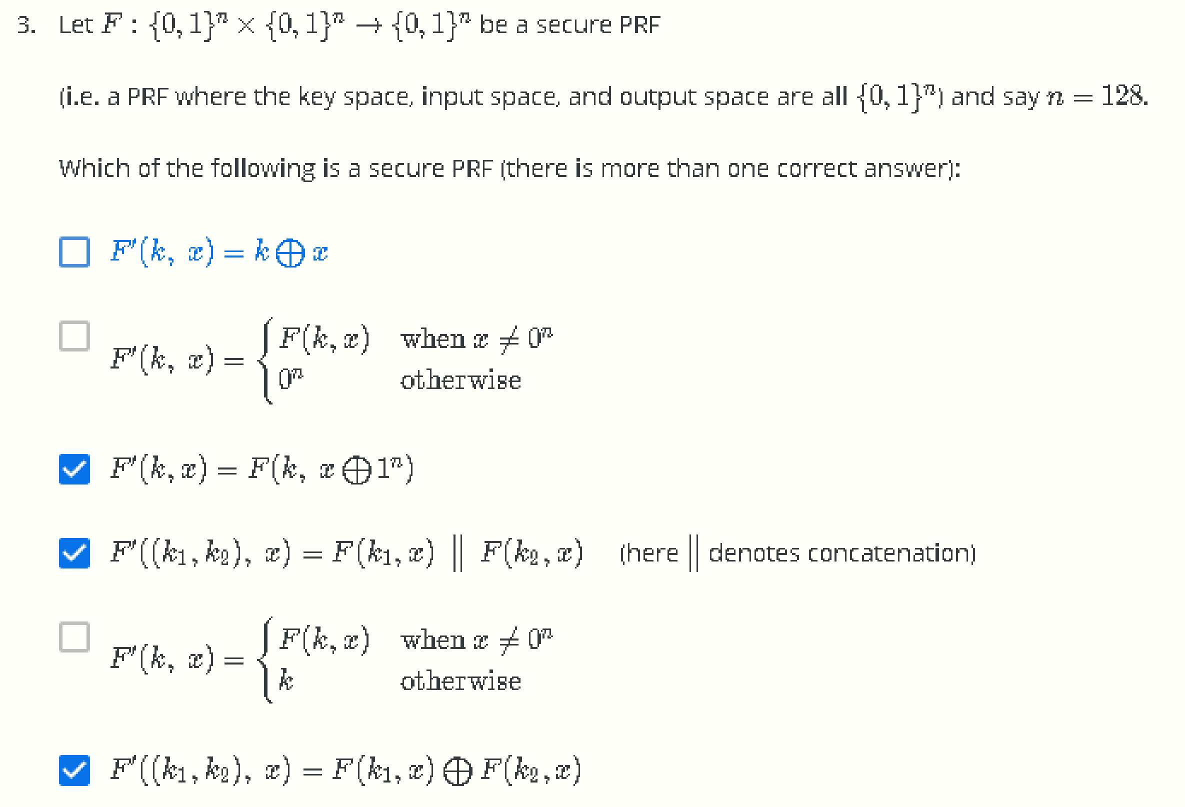
○ More than a month but less than a year

● More than a billion (10^9) years

问：若一种硬件可以做到每秒暴力搜索AES的密钥10亿次，且一个这种硬件需要200块钱，现有预算4万亿买这种硬件，不考虑除购买硬件以外的其他费用，破解一个128 bits的AES密钥需要多久？

分析：简单的计算题，根据题意，能买$2 * 10^{10}$个硬件，每秒计算速度总共为$10^9*(2 * 10^{10}) = 2 * 10^{19}$，需要共计$2^{128} / (2 * 10^{19}) = 1.7*10^{19}$秒，大约是5400亿年

Q3



3. Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF

(i.e. a PRF where the key space, input space, and output space are all $\{0,1\}^n$) and say $n = 128$.

Which of the following is a secure PRF (there is more than one correct answer):

□ $F'(k,\ x) = k \bigoplus x$

□ $F'(k,\ x) = \begin{cases} F(k,x) & \text{when } x \neq 0^n \\ 0^n & \text{otherwise} \end{cases}$

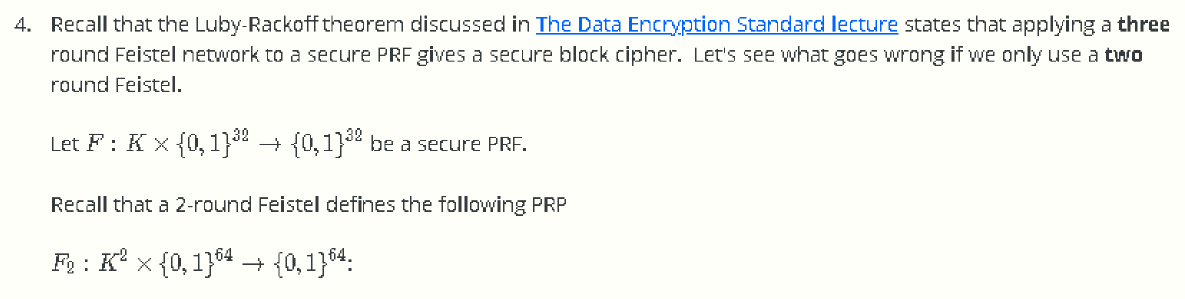
☑ $F'(k,x) = F(k,\ x \bigoplus 1^n)$

☑ $F'((k_1,k_2),\ x) = F(k_1,x)\ \|\ F(k_2,x)$ (here $\|$ denotes concatenation)

□ $F'(k,\ x) = \begin{cases} F(k,x) & \text{when } x \neq 0^n \\ k & \text{otherwise} \end{cases}$

☑ $F'((k_1,k_2),\ x) = F(k_1,x) \bigoplus F(k_2,x)$

Q4

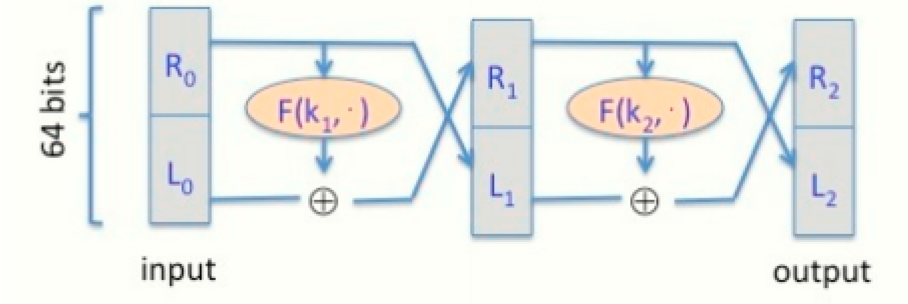


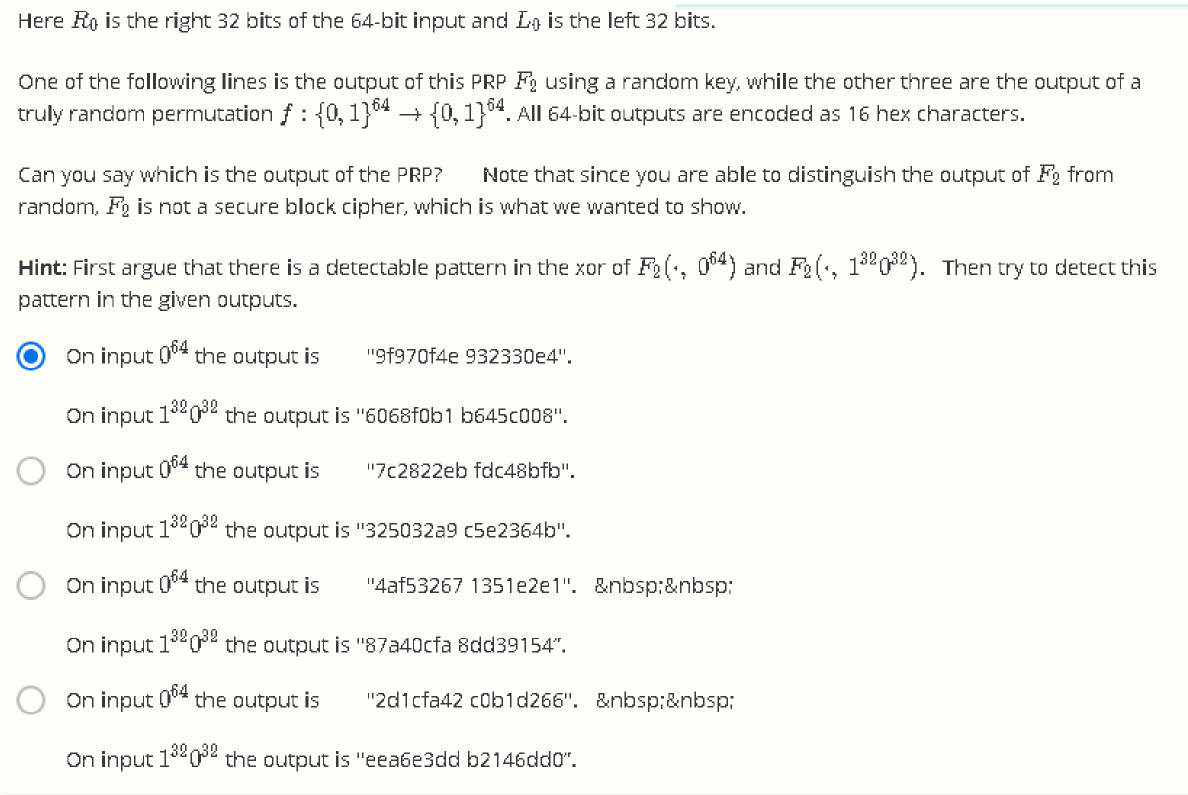
4. Recall that the Luby-Rackoff theorem discussed in The Data Encryption Standard lecture states that applying a **three** round Feistel network to a secure PRF gives a secure block cipher. Let's see what goes wrong if we only use a **two** round Feistel.

Let $F : K \times \{0,1\}^{32} \to \{0,1\}^{32}$ be a secure PRF.

Recall that a 2-round Feistel defines the following PRP

$F_2 : K^2 \times \{0,1\}^{64} \to \{0,1\}^{64}$:





Here R_0 is the right 32 bits of the 64-bit input and L_0 is the left 32 bits.

One of the following lines is the output of this PRP F_2 using a random key, while the other three are the output of a truly random permutation $f : \{0,1\}^{64} \to \{0,1\}^{64}$. All 64-bit outputs are encoded as 16 hex characters.

Can you say which is the output of the PRP? Note that since you are able to distinguish the output of F_2 from random, F_2 is not a secure block cipher, which is what we wanted to show.

Hint: First argue that there is a detectable pattern in the xor of $F_2(\cdot,\ 0^{64})$ and $F_2(\cdot,\ 1^{32}0^{32})$. Then try to detect this pattern in the given outputs.

● On input 0^{64} the output is "9f970f4e 932330e4".

On input $1^{32}0^{32}$ the output is "6068f0b1 b645c008".

○ On input 0^{64} the output is "7c2822eb fdc48bfb".

On input $1^{32}0^{32}$ the output is "325032a9 c5e2364b".

○ On input 0^{64} the output is "4af53267 1351e2e1".

On input $1^{32}0^{32}$ the output is "87a40cfa 8dd39154".

○ On input 0^{64} the output is "2d1cfa42 c0b1d266".

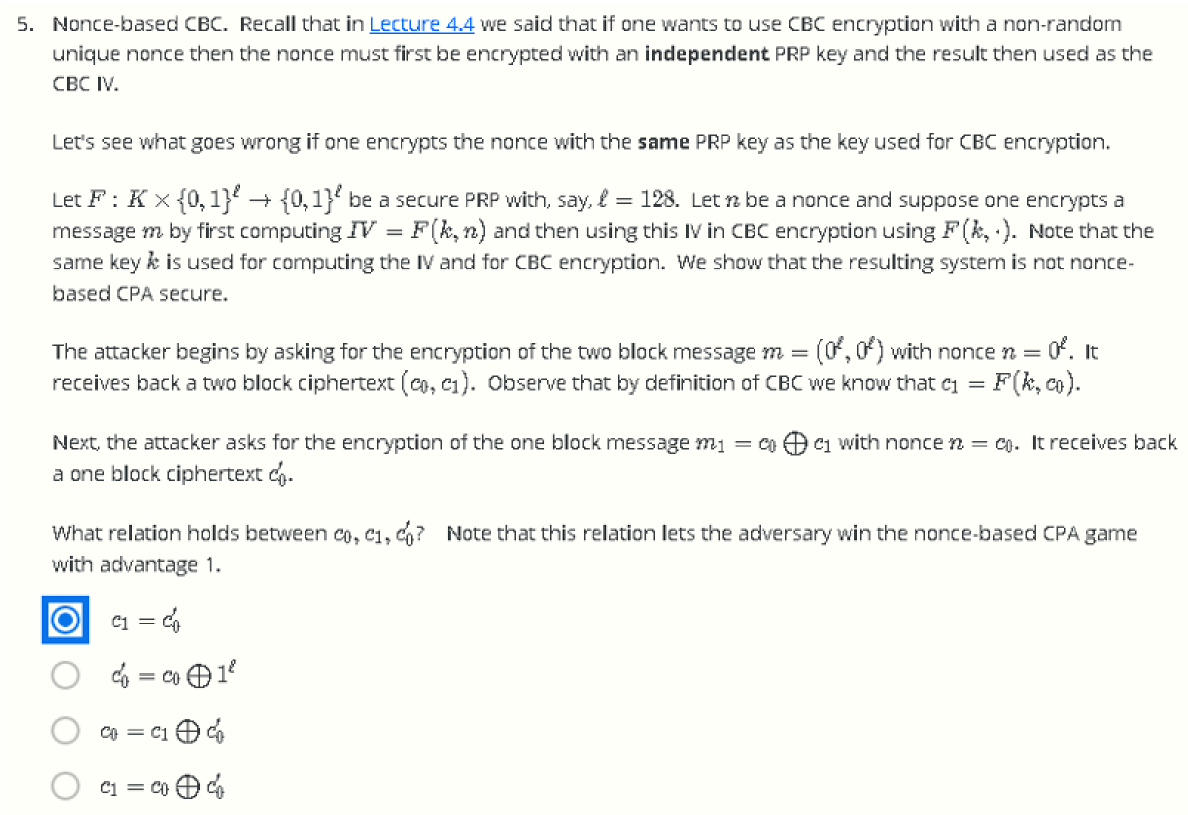
On input $1^{32}0^{32}$ the output is "eea6e3dd b2146dd0".

问：在两轮Feistel网络中会出现安全问题，下列四个输出中有一个为使用随机密钥的PRP F$_2$的输出，其余三个为真随机替换函数f的输出，问哪个是F$_2$的输出

提示：将两个值异或

第一个选项中9f970f4e932330e4与6068f0b1b645c008异或，结果的前32 bits为0xFFFF FFFF

Q5



5. Nonce-based CBC. Recall that in Lecture 4.4 we said that if one wants to use CBC encryption with a non-random unique nonce then the nonce must first be encrypted with an **independent** PRP key and the result then used as the CBC IV.

Let's see what goes wrong if one encrypts the nonce with the **same** PRP key as the key used for CBC encryption.

Let $F : K \times \{0,1\}^\ell \to \{0,1\}^\ell$ be a secure PRP with, say, $\ell = 128$. Let n be a nonce and suppose one encrypts a message m by first computing $IV = F(k, n)$ and then using this IV in CBC encryption using $F(k, \cdot)$. Note that the same key k is used for computing the IV and for CBC encryption. We show that the resulting system is not nonce-based CPA secure.

The attacker begins by asking for the encryption of the two block message $m = (0^\ell, 0^\ell)$ with nonce $n = 0^\ell$. It receives back a two block ciphertext (c_0, c_1). Observe that by definition of CBC we know that $c_1 = F(k, c_0)$.

Next, the attacker asks for the encryption of the one block message $m_1 = c_0 \bigoplus c_1$ with nonce $n = c_0$. It receives back a one block ciphertext c_0'.

What relation holds between c_0, c_1, c_0'? Note that this relation lets the adversary win the nonce-based CPA game with advantage 1.

● $c_1 = c_0'$

○ $c_0' = c_0 \bigoplus 1^\ell$

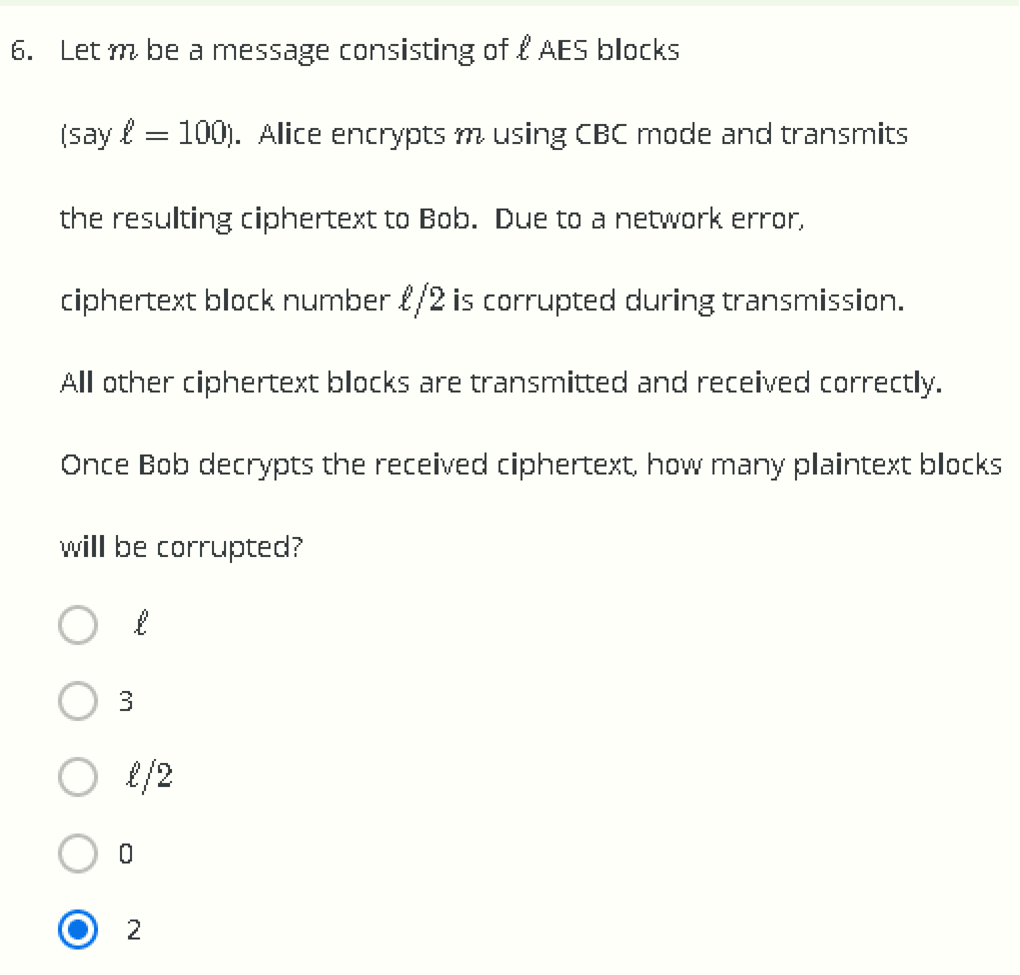
○ $c_0 = c_1 \bigoplus c_0'$

○ $c_1 = c_0 \bigoplus c_0'$

问：基于nonce的CBC模式需要使用独立的密钥先将nonce加密，之后将加密结果作为CBC模式的IV，若在CBC模式中使用与加密nonce相同的密钥则会发生什么？假设……懒得翻译了

分析：一顿计算可知选第一个

Q6



6. Let m be a message consisting of ℓ AES blocks (say $\ell = 100$). Alice encrypts m using CBC mode and transmits the resulting ciphertext to Bob. Due to a network error, ciphertext block number $\ell/2$ is corrupted during transmission. All other ciphertext blocks are transmitted and received correctly. Once Bob decrypts the received ciphertext, how many plaintext blocks will be corrupted?

○ ℓ

○ 3

○ $\ell/2$

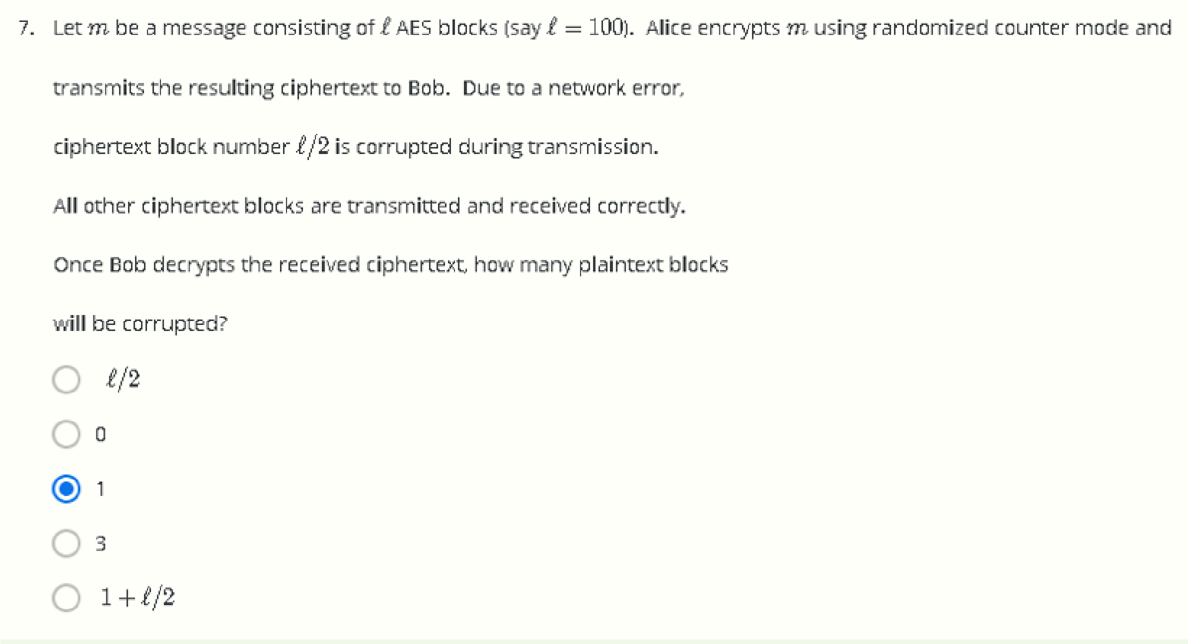
○ 0

● 2

问：Alice向Bob传输AES的加密块，使用CBC模式加密，若某一块在传输过程中出错，其他块均正确接收，Bob解密时有多少明文块会解密出错

分析：由CBC模式的特性知道，密文块出错时，解密只会影响本块和下一块的解密正确性，其余块不受影响，因此为2块

Q7



7. Let m be a message consisting of ℓ AES blocks (say $\ell = 100$). Alice encrypts m using randomized counter mode and

transmits the resulting ciphertext to Bob. Due to a network error,

ciphertext block number $\ell/2$ is corrupted during transmission.

All other ciphertext blocks are transmitted and received correctly.

Once Bob decrypts the received ciphertext, how many plaintext blocks

will be corrupted?

○ $\ell/2$

○ 0

● 1

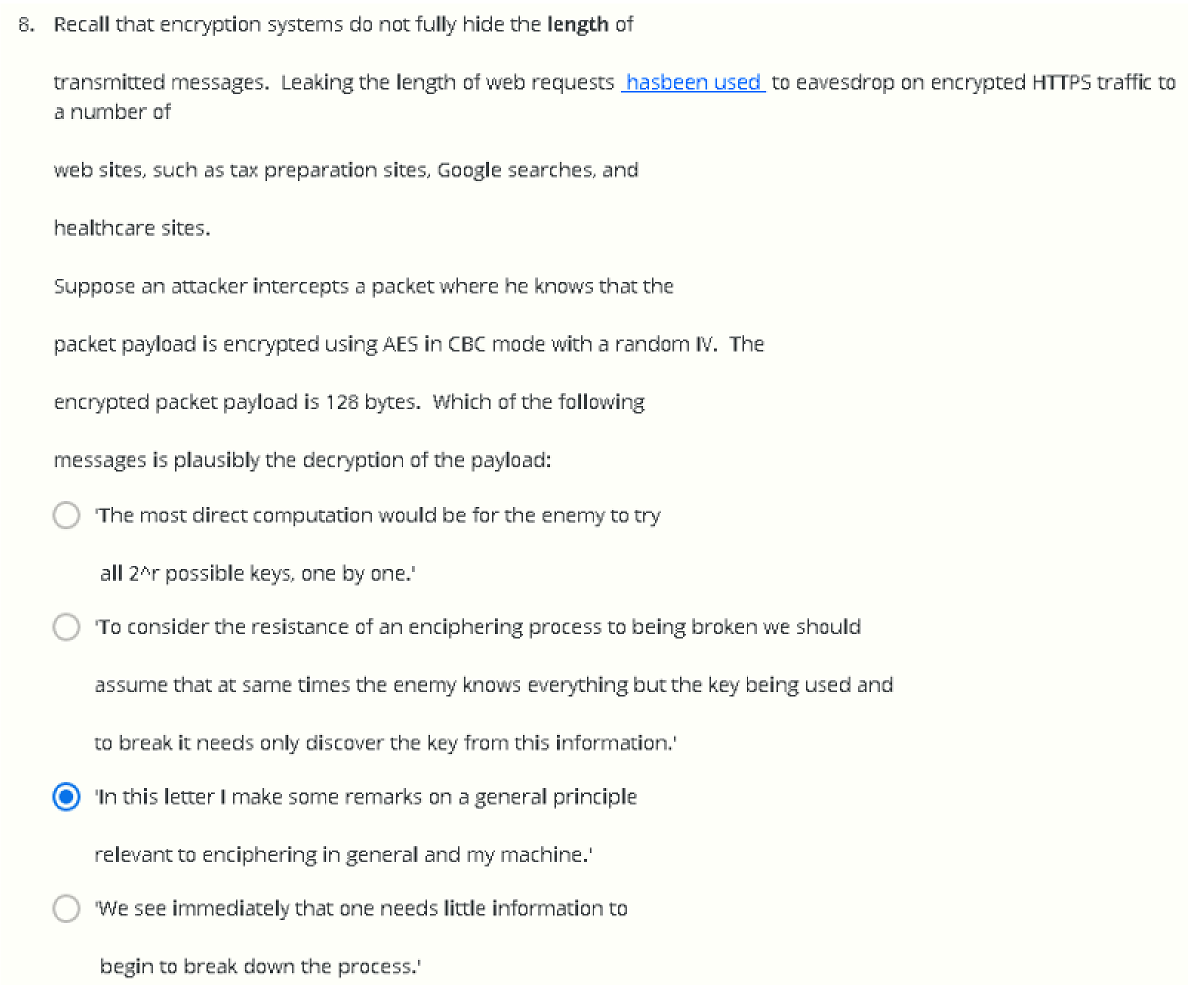
○ 3

○ $1+\ell/2$

问：Alice向Bob传输AES的加密块，使用CTR模式加密，若某一块在传输过程中出错，其他块均正确接收，Bob解密时有多少明文块会解密出错

分析：由于CTR模式特性可知，CTR模式每块加解密均独立，因此只有一块受影响

Q8



8. Recall that encryption systems do not fully hide the **length** of

transmitted messages. Leaking the length of web requests hasbeen used to eavesdrop on encrypted HTTPS traffic to a number of

web sites, such as tax preparation sites, Google searches, and

healthcare sites.

Suppose an attacker intercepts a packet where he knows that the

packet payload is encrypted using AES in CBC mode with a random IV. The

encrypted packet payload is 128 bytes. Which of the following

messages is plausibly the decryption of the payload:

○ 'The most direct computation would be for the enemy to try

all 2^r possible keys, one by one.'

○ 'To consider the resistance of an enciphering process to being broken we should

assume that at same times the enemy knows everything but the key being used and

to break it needs only discover the key from this information.'

● 'In this letter I make some remarks on a general principle

relevant to enciphering in general and my machine.'

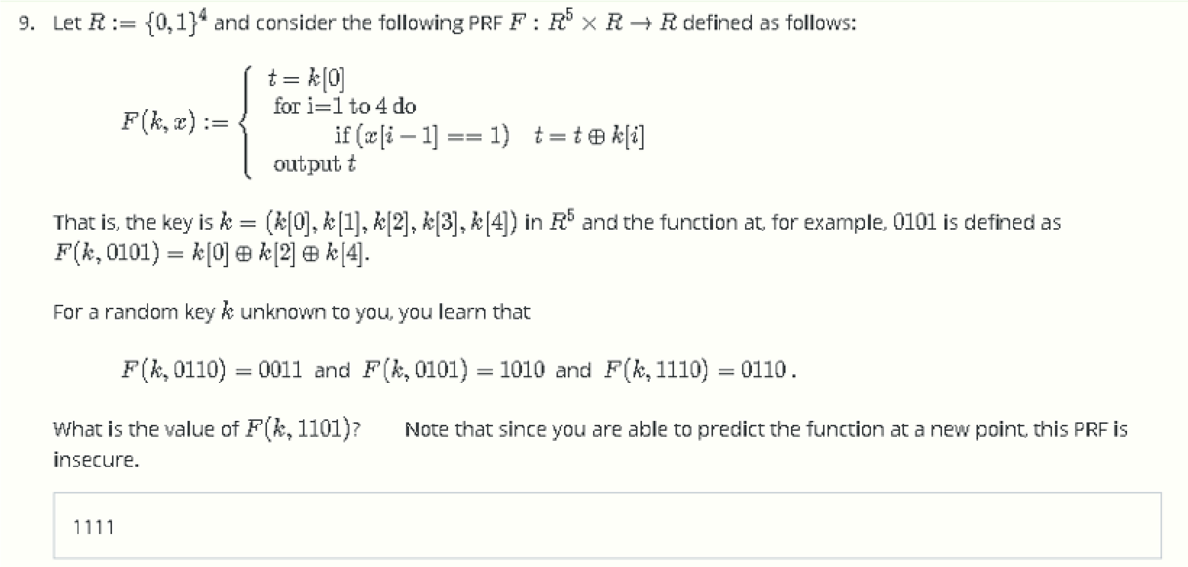
○ 'We see immediately that one needs little information to

begin to break down the process.'

问：懒得翻译

分析：数字母，第三个选项共有107个字节，填充后共有112字节，附加上IV的16字节共计128字节，满足题意

Q9



9. Let $R := \{0,1\}^4$ and consider the following PRF $F : R^5 \times R \to R$ defined as follows:

$$F(k,x) := \begin{cases} t = k[0] \\ \text{for i=1 to 4 do} \\ \qquad \text{if } (x[i-1] == 1) \quad t = t \oplus k[i] \\ \text{output } t \end{cases}$$

That is, the key is $k = (k[0], k[1], k[2], k[3], k[4])$ in R^5 and the function at, for example, 0101 is defined as $F(k, 0101) = k[0] \oplus k[2] \oplus k[4]$.

For a random key k unknown to you, you learn that

$F(k, 0110) = 0011$ and $F(k, 0101) = 1010$ and $F(k, 1110) = 0110$.

What is the value of $F(k, 1101)$? Note that since you are able to predict the function at a new point, this PRF is insecure.

1111

问:

分析：k_4一定参与xor，三个例子分别为

1. k_1 xor k_2 xor k_4 =0011
2. k_0 xor k_2 xor k_4 =1010
3. k_1 xor k_2 xor k_3 xor k_4 =0110
4. 1 和3 xor得到k_3 = 0101

设问可转化为k_0 xor k_2 xor k_3 xor k_4，即上述式2 xor k_3 =1111

In this project you will implement two encryption/decryption systems, one using AES in CBC mode and another using AES in counter mode (CTR). In both cases the 16-byte encryption IV is chosen at random and is prepended to the ciphertext.

For CBC encryption we use the PKCS5 padding scheme discussed in the lecture (14:04). While we ask that you implement both encryption and decryption, we will only test the decryption function. In the following questions you are given an AES key and a ciphertext (both are hex encoded) and your goal is to recover the plaintext and enter it in the input boxes provided below.

For an implementation of AES you may use an existing crypto library such as PyCrypto (Python), Crypto++ (C++), or any other. While it is fine to use the built-in AES functions, we ask that as a learning experience you implement CBC and CTR modes yourself.

1.

CBC key: 140b41b22a29beb4061bda66b6747e14

CBC Ciphertext 1:
4ca00ff4c898d61e1edbf1800618fb2828a226d160dad07883d04e008a7897ee2e4b7465d5290d0c0e6c6822236e1daafb94ffe0c5da05d9476be028ad7c1d81

明文：Basic CBC mode encryption needs padding.

2.

CBC key: 140b41b22a29beb4061bda66b6747e14

CBC Ciphertext 2:
5b68629feb8606f9a6667670b75b38a5b4832d0f26e1ab7da33249de7d4afc48e713ac646ace36e872ad5fb8a512428a6e21364b0c374df45503473c5242a253

明文：Our implementation uses rand. IV

3.

CTR key: 36f18357be4dbd77f050515c73fcf9f2

CTR Ciphertext 1:
69dda8455c7dd4254bf353b773304eec0ec7702330098ce7f7520d1cbbb20fc388d1b0adb5054dbd7370849dbf0b88d393f252e764f1f5f7ad97ef79d59ce29f5f51eeca32eabedd9afa9329

明文：CTR mode lets you build a stream cipher from a block cipher.

4.

CTR key: 36f18357be4dbd77f050515c73fcf9f2

CTR Ciphertext 2:
770b80259ec33beb2561358a9f2dc617e46218c0a53cbeca695ae45faa8952aa0e311bde9d4e01726d3184c34451

明文：Always avoid the two time pad!