Problem 1

† Data compression is often used in data storage and transmission. Suppose you want to use data compression in conjunction with encryption. Does it make more sense to:

- ☑ Compress then encrypt.
- ☐ Encrypt then compress.
- ☐ The order does not matter – either one is fine.
- ☐ The order does not matter – neither one will compress the data.

The order does matter, if text is encrypte first, it is random the compression won't work perfectly if compressed and then encrypted, the compressio work.

Problem 2

† Let G: $\{0,1\}^s \to \{0,1\}^n$ be a secure PRG. Which of the following is a secure PRG:

- ☐ G'(k) = G(k) ‖ G(k) a distingushier will output not random but to the last n bits
- ☑ G'(k) = G(k ⊕ 1^s) a distinguisher for G' gives a distinguisher output
- ☐ G'(k) = G(0) a distinguisher will output not random bc to G(0)
- ☐ G'(k) = G(1) a distinguisher will output not random bc to G(0)
- ☐ G'(k) = G(k) ‖ 0 a distinguisher will output not random or to
- ☑ G'(k_1, k_2) = G(k_1) ‖ G(k_2) a distinguisher for G' gives a distinguish output
- ☑ G'(k) = reverse(G(k)) the output is distinguish
- ☐ G'(k) = $rotation_n$(G(k)) the output is not random

Hint:
"‖" denotes concatenation.
"reverse(x)" reverses the string x so that the first bit of x is the last bit of reverse(x), the second bit of x is the second to last bit of reverse(x), and so on.
"$rotation_n$(x)" rotates the string x by n positions. If n>0, it rotates right; if n<0, it rotates left, and characters shifted off one end reappear at the other.

Problem 3

Let (E, D) be a (one-time) semantically secure cipher with key space $K = \{0,1\}^k$. A bank wishes to split a decryption key k $\in \{0,1\}^k$ into two pieces p_1 and p_2 so that both are needed for decryption. The piece p_1 can be given to one executive and p_2 to another so that both must contribute their pieces for decryption to proceed.

The bank generates random k_1 in $\{0,1\}^k$ and sets k_1' $\leftarrow$ k $\oplus$ k_1. Note that $k_1 \oplus k_1$' = k. The bank can give k_1 to one executive and k_1' to another. Both must be present for decryption to proceed since, by itself, each piece contains no information about the secret key k (note that each piece is a one-time pad encryption of k).

Now, suppose the bank wants to split k into three pieces p_1, p_2, p_3 so that any two of the pieces enable decryption using k. This ensures that even if one executive is out sick, decryption can still succeed. To do so the bank generates two random pairs (k_1, k_1') and (k_2, k_2') as in the previous paragraph so that $k_1 \oplus k_1' = k_2 \oplus k_2' =$ k. How should the bank assign pieces so that any two pieces enable decryption using k, but no single piece can decrypt?

- ☐ $p_1 = (k_1, k_2)$, $p_2 = (k_1, k_2)$, $p_3 = (k_2')$
- ☐ $p_1 = (k_1, k_2)$, $p_2 = (k_1', k_2')$, $p_3 = (k_2')$
- ☑ $p_1 = (k_1, k_2)$, $p_2 = (k_1', k_2)$, $p_3 = (k_2')$
- ☐ $p_1 = (k_1, k_2)$, $p_2 = (k_2, k_2')$, $p_3 = (k_2')$
- ☐ $p_1 = (k_1, k_2)$, $p_2 = (k_1')$, $p_3 = (k_2')$

Combination 1, 2, 5 cannot decrypt when 2 people come together

combination 4 can decrypt only when p2 is present

Combination 3 is only solution

Problem 4

Let M = C = K = { 0, 1, 2, . . . , 255 } and consider the following cipher defined over (K, M, C):
E(k, m) = m + k (mod 256); D(k, c) = c - k (mod 256)
Does this cipher has perfect secrecy?

☐ No, there is a simple attack on this cipher.

☑ Yes

☐ No, only the One Time Pad has perfect secrecy.

as with the one-time pad there is exactly one key mapping a given message m to a given cipher c

Problem 5

† Let (E, D) be a (one-time) semantically secure cipher where the message and ciphertext space is $\{0,1\}^n$. Which of the following encryption schemes are (one-time) semantically secure?

1 ☐ E'(k, m) = E(0^n, m)

2 ☑ E'((k, k'), m) = E(k, m) ‖ E(k', m)

3 ☐ E'(k, m) = E(k, m) ‖ MSB(m)

4 ☐ E'(k, m) = 0 ‖ E(k, m) (i.e. prepend 0 to the ciphertext)

5 ☐ E'(k, m) = E(k, m) ‖ k

6 ☑ E'(k, m) = reverse(E(k, m))

7 ☑ E'(k, m) = $rotation_n$(E(k, m))

1 To break semantic security an attacker would ask for the encryption can easily distinguish Exp(0) from Exp(1)

2 an attack on E' gives an attack on E

3 To break semantic security, an attacker would ask for the cypher can easily distinguish Exp(0) from Exp(1)

4 an attack on E' gives an attack on E

5 To break semantic security, an attacker would need the security and use it to decrypt the challenge ciphertext

b an attack on E' gives an attack on E

n an attack on E' gives an attack on E

Problem 6

Suppose you are told that the one time pad encryption of the message "attack at dawn" is 6c73d5240a948c86981bc294814d (the plaintext letters are encoded as 8-bit ASCII and the given ciphertext is written in hex). What would be the one time pad encryption of the message "defend at noon" under the same OTP key?

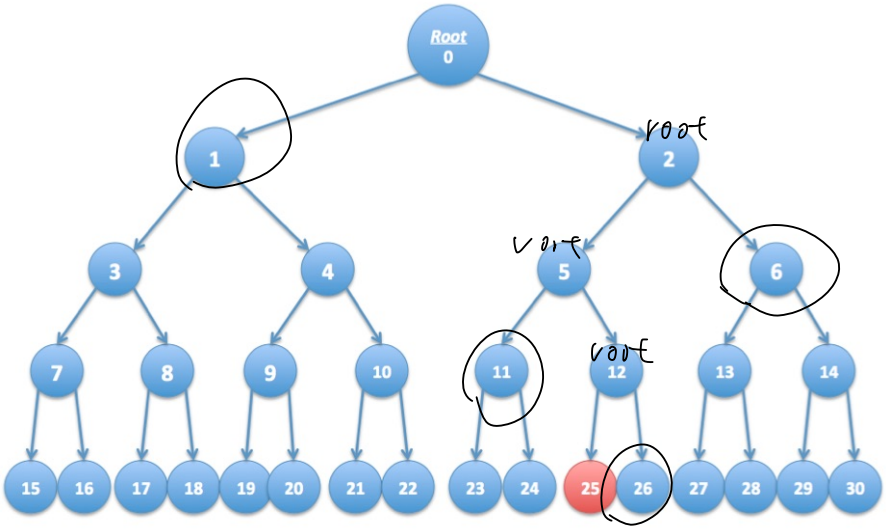
hex:

6c 73 d5 24 0a 94 8c 86 98 1b c2 94 81 4d

given the origina message and encoded cypher, we can recover that key After Xoring the key with the her message, we get correct answer

Problem 7

† The movie industry wants to protect digital content distributed on DVD's. We develop a variant of a method used to protect Blu-ray disks called AACS.



- ☐ 21
- ☐ 17
- ☐ 5
- ☑ 26
- ☑ 6
- ☑ 1
- ☑ 11
- ☐ 24

25 is to the right of the key 1

we can safely include all elements unders key 1, same logic we choose 6, 11, 26

Extra Credit

Did SHA-256 and SHA-512-truncated-to-256-bits have the same security properties? Which one is better? Please explain in detail.

SHA-256 and SHA-512-truncated-to-256-bits are both secure cryptographic hash functions. SHA-256 is faster and widely supported, making it a practical choice for most applications. However, SHA-512-truncated-to-256-bits provides a larger security margin due to its origin from SHA-512. If performance is critical, use SHA-256. For a higher security margin, especially in sensitive applications, consider SHA-512-truncated-to-256-bits.