Quiz 3

111550057 資工15 莊婷馨

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 conventional method is "compress then encrypt", because it is more effective to compress plain data. After reducing redundancy of the data, it will be easier to encrypt (less calculation) and harder to crack.

However, there are still some advantages to "encrypt then compress", one is to avoid compression attacks like CRIME and BREACH attacks. The metadata of encryption can also be kept for further usage.

Every scenario has an order of encryption and compression that has the best performance. Although either one is fine, it is best to test and compare all combinations to obtain the optimal solution.

Problem 2

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

❌ $G'(k) = G(k) \parallel G(k)$ -> repeat patterns ∴ not random

✅ $G'(k) = G(k \oplus 1^s)$ -> inverse keyword will not effect randomness

❌ $G'(k) = G(0)$ -> the key is always 0 ∴ not random

❌ $G'(k) = G(1)$ -> the key is always 1 ∴ not random

❌ $G'(k) = G(k) \parallel 0$ -> the LSB of $G'(k)$ is always 0 ∴ not random

✅ $G'(k1, k2) = G(k1) \parallel G(k2)$ -> concatenating 2 random bit strings ∴ still random

✅ $G'(k) = \text{reverse}(G(k))$ -> reverse will not effect randomness

✅ $G'(k) = \text{rotation}_n(G(k))$ -> rotation will not effect randomness

Problem 3

❌ $p_1 = (k_1, k_2)$, $p_2 = (k_1, k_2)$, $p_3 = (k_2')$ -> cannot decrypt by P1 and P2

❌ $p_1 = (k_1, k_2)$, $p_2 = (k_1', k_2')$, $p_3 = (k_2')$ -> cannot decrypt by P2 and P3

✅ $p_1 = (k_1, k_2)$, $p_2 = (k_1', k_2)$, $p_3 = (k_2')$

❌ $p_1 = (k_1, k_2)$, $p_2 = (k_2, k_2')$, $p_3 = (k_2')$ -> p_2 can decrypt by itself

❌ $p_1 = (k_1, k_2)$, $p_2 = (k_1')$, $p_3 = (k_2')$ -> cannot decrypt by p_2 and p_3

Only the third option can decrypt with any two of the keys, but not only one key.

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.

$\forall m, c \in M, C$, if $E(k, m) = c$

$m + k \pmod{256} = c$

$\Rightarrow k = c - m \pmod{256}$

$\Rightarrow$ number of $\{k \in K, E(k, m) = c\} = 1 \rightarrow$ const.

$\therefore$ This cipher has perfect secrecy

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?

❌ $E'(k, m) = E(0^n, m)$

-> When attacker ask for encryption of 0^n and 1^n, EXP(0) and EXP(1) can be easily distinguished.

✅ $E'((k, k'), m) = E(k, m) \parallel E(k', m)$ -> attacking E' is the same as attacking E

❌ $E'(k, m) = E(k, m) \parallel MSB(m)$

-> When attacker ask for encryption of 0^n and $1^1 0^{n-1}$, EXP(0) and EXP(1) can be easily distinguished.

✅ $E'(k, m) = 0 \parallel E(k, m)$ -> attacking E' is the same as attacking E

❌ E'(k, m) = E(k, m) ‖ k -> the ciphertext reveals the secret key ∴ definitely not secure

✅ E'(k, m) = reverse(E(k, m)) -> attacking E' is the same as attacking E

✅ E'(k, m) = rotation$_n$(E(k, m)) -> attacking E' is the same as attacking 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?

Ans: 6962C720079B8C86981BC89A994D

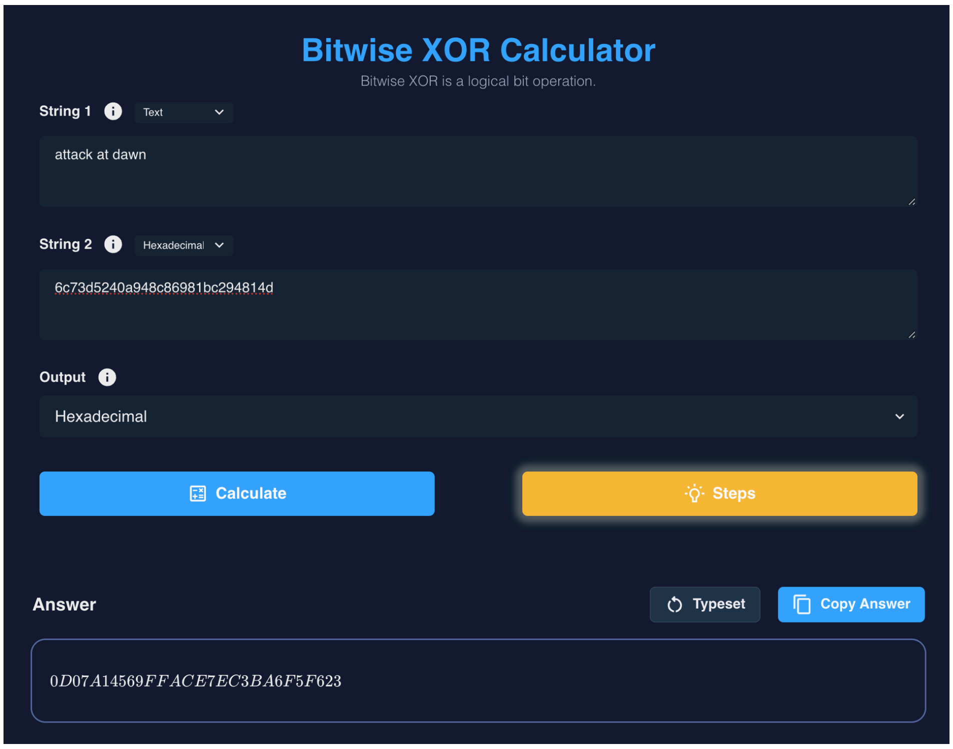
Let m = "attack at dawn" and c = 6c73d5240a948c86981bc294814d.

If $m \oplus k = c$, then

$$k = 0 \oplus k = m \oplus (m \oplus k) = m \oplus c$$

Therefore, we can get the key by $m \oplus c$: 0D07A14569FFACE7EC3BA6F5F623

(use online tool: Bitwise XOR Calculator)



String 1: Text

attack at dawn

String 2: Hexadecimal

6c73d5240a948c86981bc294814d

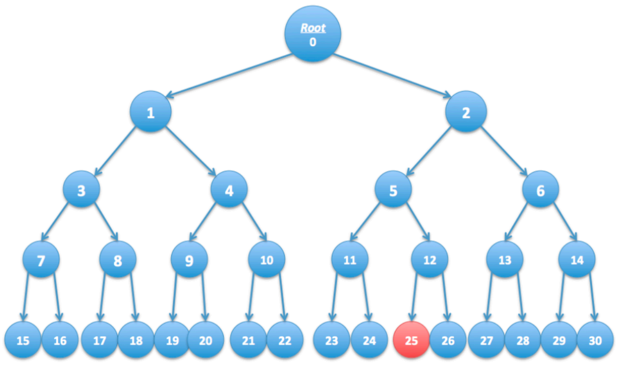
Output: Hexadecimal

Answer

0D07A14569FFACE7EC3BA6F5F623

Next, get the next ciphertext by key XOR "defend at noon" using online tool.

Problem 7



✅ 1, 6, 11, 26

❌ 21, 17, 5, 24

Since 25 is under 2, 5 and 12, we should choose the other node on the same level to avoid including 25, which are 1, 6 and 11. The only one node left is 26. These four nodes are the answers.

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-512-truncated-to-256-bits is more secure than SHA-256, because SHA-256 is under the threat of length extension attacks. What's more, SHA-512-truncated-to-256-bits runs faster on 64 bit platforms. Generally, SHA-512-truncated-to-256-bits is a better choice.