密碼工程 Quiz3

學號:112550090 姓名:曾士珍

Problem1

† 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:

選項	explain
■ Compress then encrypt.	將資料壓縮後,減少了需要加密的資料量,從而提高了加密和解密的速度,特別是處理大量資料時。
■ Encrypt then compress.	通常在對安全性要求較高的環境下使用,先加密數據,然後再壓縮,可以確保即使壓縮後的數據在傳輸過程中被竊取,也無法被解密。
■ The order does not matter– either one is fine.	
□ The order does not matter– neither one will compress the data.	

Problem2

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

選項	explain
□ G'(k) = G(k) ‖ G(k)	都用 k 來生成,最後面的 n 碼不會是 random 的
■ G'(k) = G(k ⊕ 1s)	先把 plaintext 跟 1^s 做 xor 只是將原本 0 的值變 1,1 的值變 0 不影響其隨機性
□ G'(k) = G(0)	因其不依賴輸入的密鑰 k,只會產生相同的輸出 G(0),不符合 PRG 的隨機性
□ G'(k) = G(1)	因其不依賴輸入的密鑰 k,只會產生相同的輸出 G(1),不符合 PRG 的隨機性
□ G'(k) = G(k) ‖ 0	連接最後一位是 0 時,會輸出非隨機的結果,G'(k)的最後一位會是 0

■ G’(k1, k2) = G(k1) ‖ G(k2)	兩個隨機的結果接在一起還是隨機的
■ G’(k) = reverse(G(k))	把產生出來隨機的結果倒過來還是隨機的
■ G’(k) = rotationn(G(k))	把產生出來隨機的結果的其中 n 位倒過來還是隨機的

Problem3

選項	explain
□ p1 =(k1, k2), p2 = (k1, k2), p3 = (k2’)	題目要求任兩個人在時要可以解鎖，只有選項三符合。情況如下 當 p1, p2 時：k1 ⊕ k1’ = k 當 p2, p3 時：k2 ⊕ k2’ = k 當 p1, p3 時：k2 ⊕ k2’ = k
□ p1 =(k1, k2), p2 = (k1’, k2’), p3 = (k2’)	
■ p1 =(k1, k2), p2 = (k1’, k2), p3 = (k2’)	
□ p1 =(k1, k2), p2 = (k2, k2’), p3 = (k2’)	
□ p1 =(k1, k2), p2 = (k1’), p3 = (k2’)	

Problem4

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?

選項	Explain
□ No, there is a simple attack on this cipher.	E(k, m) = m + k (mod 256)，對於所有明文和密鑰，生成任意密文 c 的概率都是 1/256。 對於給定的密文 c 和密鑰 k，解密函數 D(k, c) = c - k (mod 256) 將唯一恢復出明文 m。 因此，這個密碼具有完美的保密性，因為無論明文是什麼，每個密文生成的概率都是相同的，且每個明文都可以使用密鑰從其對應的密文中恢復出來。
■ Yes	
□ No, only the One Time Pad has perfect secrecy.	

Problem5

† 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?

語義安全是指即使攻擊者擁有任意數量的密文和相應的明文，也無法從中推斷出有關明文的任何有用資訊，除非他們知道加密所使用的密鑰。

選項	explain
□ E'(k, m) = E(0n, m)	攻擊者請求對 0^n 和 1^n 的加密，可輕鬆區分 EXP(0)和 EXP(1)。
■ E'((k, k'), m) = E(k, m) ‖ E(k', m)	E(k, m) 和 E(k', m) 都是語義安全的，並且使用金鑰 k 和 k' 對 m 進行加密是獨立的，因此將它們串接起來也是語義安全的
□ E'(k, m) = E(k, m) ‖ MSB(m)	攻擊者請求對 0^n 和 10^n-1 進行加密，由於最高有效位不一樣，即使明文只有一位的差異，攻擊者也可以從密文中區分出 EXP(0) 和 EXP(1)。
■ E'(k, m) = 0 ‖ E(k, m)	將密文的第一位與 0 串接，不會讓其更容易被破解，會維持語意安全
□ E'(k, m) = E(k, m) ‖ k	攻擊者可能在密文中讀出金鑰，然後使用它來解密密文。
■ E'(k, m) = reverse(E(k, m))	將語意安全密文倒過來不會影響語意安全的特性
■ E'(k, m) = rotation^n(E(k, m))	將語意安全密文的其中 n 位倒過來不會影響語意安全的特性

Problem6

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?

”attack at dawn” are encoded as 0x61747461636b206174206461776e

61747461636b206174206461776e ⊕ Key = 6c73d5240a948c86981bc294814d

Key = 6c73d5240a948c86981bc294814d ⊕ 61747461636b206174206461776e

= d07a14569fface7ec3ba6f5f623

”defend at noon” are encoded as 0x646566656e64206174206e6f6f6e

New_cipher = Key ⊕ 646566656e64206174206e6f6f6e
= **6962c720079b8c86981bc89a994d**

Problem7

As shown below, consider a tree with n = 16 leaves. Suppose the leaf node labeled 25 corresponds to an exposed DVD player key. Check the set of keys below under which to encrypt the key k so that every player other than player 25 can decrypt the DVD. Only four keys are needed.

若加密的金鑰中含有該葉節點的祖先，其可以解密該金鑰，而後播放器就可以解密電影。 由於不希望解密播放器編號 25，因此需要避免擁有 0、2、5、12 金鑰。

選項	explain
□ 21	最佳的選法是，只需要 four keys 26：可以解密自己 6：為 27-30 台播放器的共同祖先 1：為 15-22 台播放器的共同祖先 11：為 23-24 台播放器的共同祖先
□ 17	
□ 5	
■ 26	
■ 6	
■ 1	
■ 11	
□ 24	

Extra Credit

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

SHA-256 和 SHA-512-truncated-to-256-bits 都是 SHA-2 系列的加密哈希函数的變體

	SHA-256	SHA-512-truncated-to-256-bits
輸出 hash 長度	產生 256 位（32 字節）	產生 512 位（64 字節），但截斷為 256 位以匹配 SHA-256 的輸出長度

碰撞抗性	有高水準的碰撞抗性	有較長的輸出長度，理論上提供更高水準的碰撞抗性
前像抗性	提供強大的前像抗性	提供強大的前像抗性
性能	比較快，因其在 32 位字上運作	比較慢，因其在 64 位字上運作
加密強度	供強大的安全性質	供強大的安全性質

結論：由於 SHA-256 更快的性能和廣泛的應用，通常在大多數實際應用中更喜歡使用 SHA-256。除非有對更長哈希長度的特定要求，否則 SHA-256 通常是密碼哈希的首選。