Problem 1

(O) Compress then encrypt

Explanation: As compression can reduce the size and entropy of the data, making easier to encrypt.

(O) Encrypt then compress

Explanation: As it reduces the storage to save the file, taking up less spaces, and make it easier to manage and access the data.

(O) The order does not matter – either one is fine

Explanation: As both have their benefit.

(X) The order does not matter – neither one will compress the data

Explanation: They both compress the data.

Problem 2

(X) G'(k) = G(k) || G(k)

Explanation: Suppose that

$$A(x) = \begin{cases} 1 & \text{if the first } \frac{n}{2} \text{ bits equals to the last } \frac{n}{2} \text{ bits} \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\begin{aligned} ADV_{PRG}[A, G'] &= |\underset{k \overset{R}{\leftarrow} \{0,1\}^s}{Pr}[A(G'(k)) = 1] - \underset{r \overset{R}{\leftarrow} \{0,1\}^n}{Pr}[A(r) = 1]| \\ &= |1 - \frac{1}{2^{\frac{n}{2}}}| \end{aligned}$$

Which is not negligible.

(O) G’(k) = G(k$\oplus 1^s$)

Explanation: Let k’= k$\oplus 1^s$, then G’(k) = G(k’). As k’ still lies within the set $\{0,1\}^s$, G(k’) is also a secure PRG.

(X) G’(k) = G(0)

Explanation: No matter what k is input, it always outputs a constant,

which is clearly not random.

(X) G’(k) = G(1)

Explanation: No matter what k is input, it always outputs a constant,

which is clearly not random.

(X) G’(k) = G(k) $||$ 0

Explanation: Suppose that

$$A(x) = \begin{cases} 1 & \text{if the last bit is 0} \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\begin{aligned} ADV_{PRG}[A, G'] &= |\underset{k \overset{R}{\leftarrow} \{0,1\}^s}{Pr}[A(G'(k)) = 1] - \underset{r \overset{R}{\leftarrow} \{0,1\}^n}{Pr}[A(r) = 1]| \\ &= |1 - \frac{1}{2}| \\ &= \frac{1}{2} \end{aligned}$$

Which is not negligible.

(O) G’(k1, k2) = G(k1) || G(k2)

Explanation: As G(k1) and G(k2) are distinct pseudo-random bits, we cannot observe any relationship or pattern between them, so this is a PRG.

(O) G’(k) = reverse(G(k))

Explanation: Define P(g(00)) is the probability of finding two consecutive zeroes in the output of G(k), and so does P(g(01)), P(g(10)) and P(g(11)). Similarly, we define P(g’(00)) is the probability of finding two consecutive zeroes in the output of G’(k). We can observe that P(g(00))=P(g’(00)), P(g(01))=P(g’(10)), P(g(10))=P(g’(01)) and P(g(11))=P(g’(11)). As the probability distribution among g(00), g(01), g(10) and g(11) have negligible difference compared to TRG, the probability distribution of G’ will also have negligible difference compared to TRG.

(O) $G’(k) = rotation_n\big(G(k)\big)$

Explanation: Using the thought above, rotation barely affect the probability distribution among g(00), g(01), g(10) and g(11), which only affects the first two bits and the last two bits. When the output bits are long enough, the effect is negligible.

Problem 3

(X) p1 = (k1, k2), p2 = (k1, k2), p3 = (k2’)

Explanation: They cannot decrypt the key if p3 is absent.

(X) p1 = (k1, k2), p2 = (k1’, k2’), p3 = (k2’)

Explanation: They cannot decrypt the key if p1 is absent.

(O) p1 = (k1, k2), p2 = (k1’, k2), p3 = (k2’)

Explanation: if p1 is absent, p2 and p3 can use (k2, k2’) to decrypt the key; if p2 is absent, p1 and p3 can also use (k2, k2’) to decrypt the key; if p3 is absent, p1 and p2 can use (k1, k1’) to decrypt the key.

(X) p1 = (k1, k2), p2 = (k2, k2’), p3 = (k2’)

Explanation: p2 can decrypt the key by himself.

(X) p1 = (k1, k2), p2 = (k1’), p3 = (k2’)

Explanation: They cannot decrypt the key if p1 is absent.

Problem 4

(O) Yes

Explanation: According to the definition, a cipher can achieve perfect secrecy if the length of key >= the length of plaintext and the key is chosen randomly to encrypt each plaintext. (as the plaintext has only one number)

Problem 5

(X) E’(k, m) = E(0^n, m)

Explanation: As the key is fixed, and by Kerckhoff’s principle, we assume the attacker knows the encryption algorithm, the attacker can encrypt the plaintext himself to tell E’(k, m_0) from E’(k, m_1).

(O) E’((k, k’), m) = E(k, m) || E(k’, m)

Explanation: it concatenates two semantically secure ciphertext encrypted with the different key, which is still secure as it does not reveal any relationship or information between the two encryptions. Therefore, the attacker cannot tell which ciphertext belongs to which plaintext.

(X) E’(k, m) = E(k, m) || MSB(m)

Explanation: Suppose the attacker finds a way to determine the MSB of plaintext from the ciphertext, then he tell two plaintexts correctly if they have different MSB.

(O) E’(k, m) = 0 $||$ E(k, m)

Explanation: As we prepend a constant, the attacker cannot find any extra information from the constant, which means E’(k, m) is as secure as E(k, m).

(X) E’(k, m) = E(k, m) $||$ k

Explanation: As the key is attached to the ciphertext, the attacker can use the key to encrypt the plaintext himself to tell E’(k, m_0) from E’(k, m_1).

(O) E’(k, m) = reverse(E(k, m))

Explanation: As it only reverses the sequence of a semantically secure ciphertext without changing its distribution, it is still semantically secure.

(O) E’(k, m) = $rotation_n$(E(k, m))

Explanation: As it rotates the same time regarding the sequence of E(k, m_0) and E(k, m_1), the attacker still cannot find any feature of m_0 and m_1, so E’(k, m) is as secure as E(k, m).

Problem 6

Key = “attack at dawn” $\oplus$ 6c73d5240a948c86981bc294814d

= 61747461636b206174206461776e $\oplus$ 6c73d5240a948c86981bc294814d

= 0d07a14569fface7ec3ba6f5f623

Ans = “defend at noon” $\oplus$ 0d07a14569fface7ec3ba6f5f623

= 646566656e64206174206e6f6f6e $\oplus$ 0d07a14569fface7ec3ba6f5f623

= 6962c720079b8c86981bc89a994d

Problem 7

If we traverse from the root node to node 25, we can find the path is root➔2➔5➔12➔25. To make player 25 not able to decrypt the DVD, we cannot select nodes whose children contains the nodes above. Moreover, we can only select 4 keys to encrypt, so we must maximize the DVD player covered in each selection. Therefore, we select 1 in the first layer (as opposed to 2), 6 in the second layer (as opposed to 5), 11 in the third layer (as opposed to 12) and 26 in the fourth layer (as opposed to 25 itself).

Ans: 1, 6, 11, 26

Extra Credit

Yes, they have the same security properties. However, there's one respect that truncated-SHA512 is better than SHA256: truncated SHA-512 is invulnerable to length extension attack and SHA-512 is faster on 64-bit processors without acceleration.