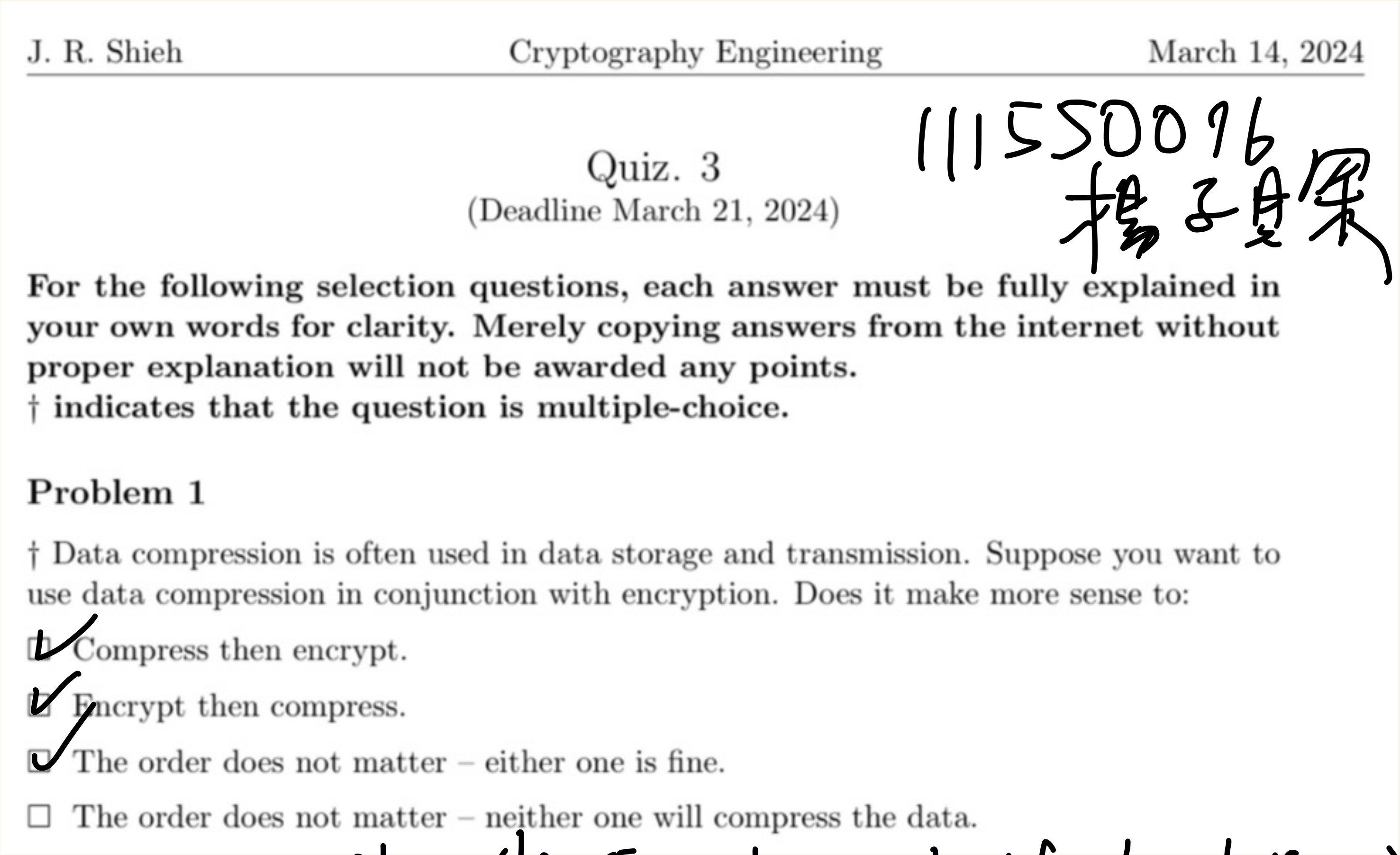


111550096
楊子賢

Quiz. 3

(Deadline March 21, 2024)

For the following selection questions, each answer must be fully explained in your own words for clarity. Merely copying answers from the internet without proper explanation will not be awarded any points.
† indicates that the question is multiple-choice.

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.

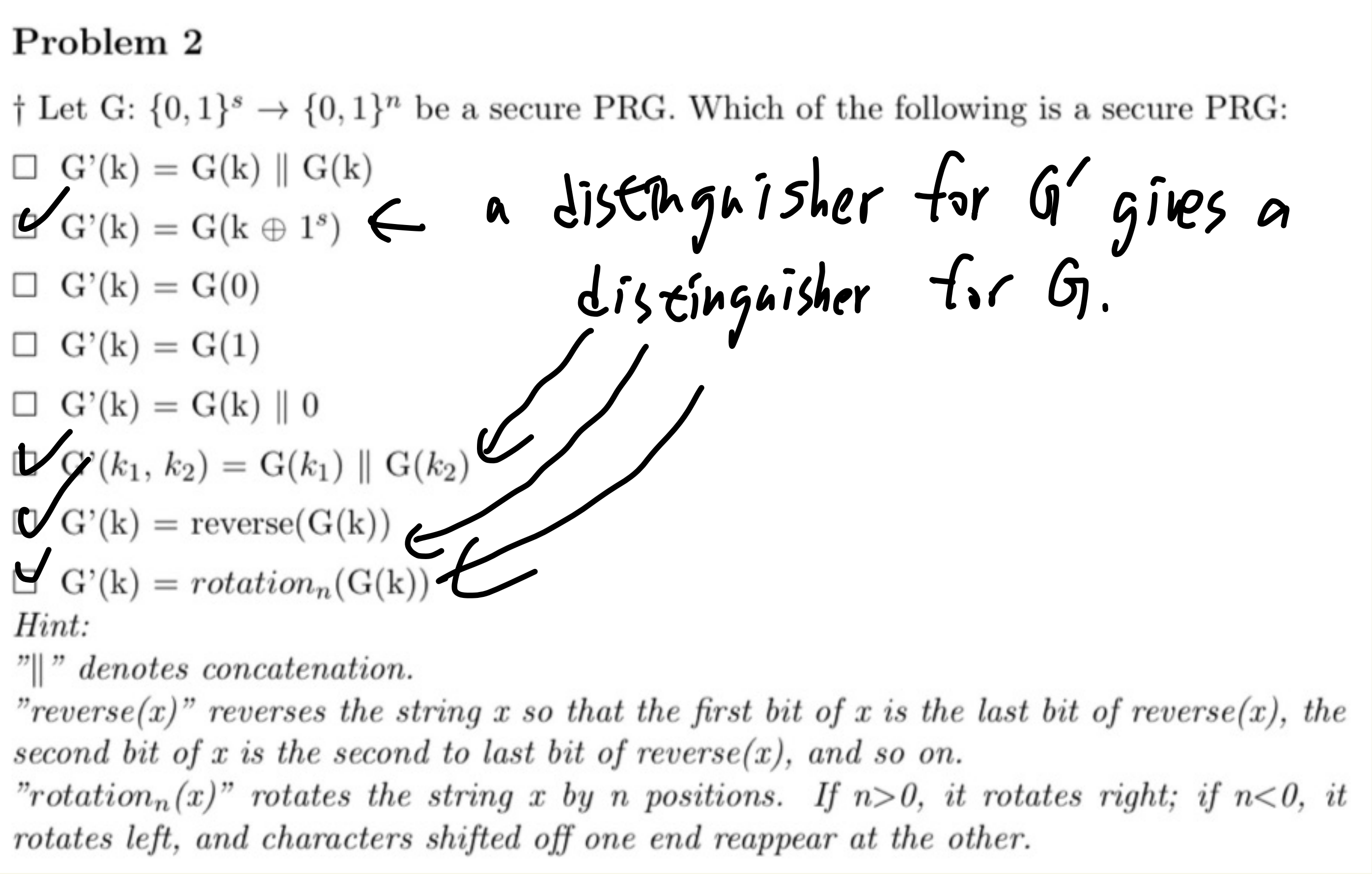
☑ Encrypt then compress.

☑ The order does not matter – either one is fine.

□ The order does not matter – neither one will compress the data.

1. Compress can reduce the size and entrophy of the data, ⇒ easier to encrypt.

2、avoid compression attack or preserve encryption metadata.



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) ‖ G(k)

☑ G'(k) = G(k ⊕ 1^s) ← a distinguisher for G′ gives a distinguisher for G.

□ G'(k) = G(0)

□ G'(k) = G(1)

□ G'(k) = G(k) ‖ 0

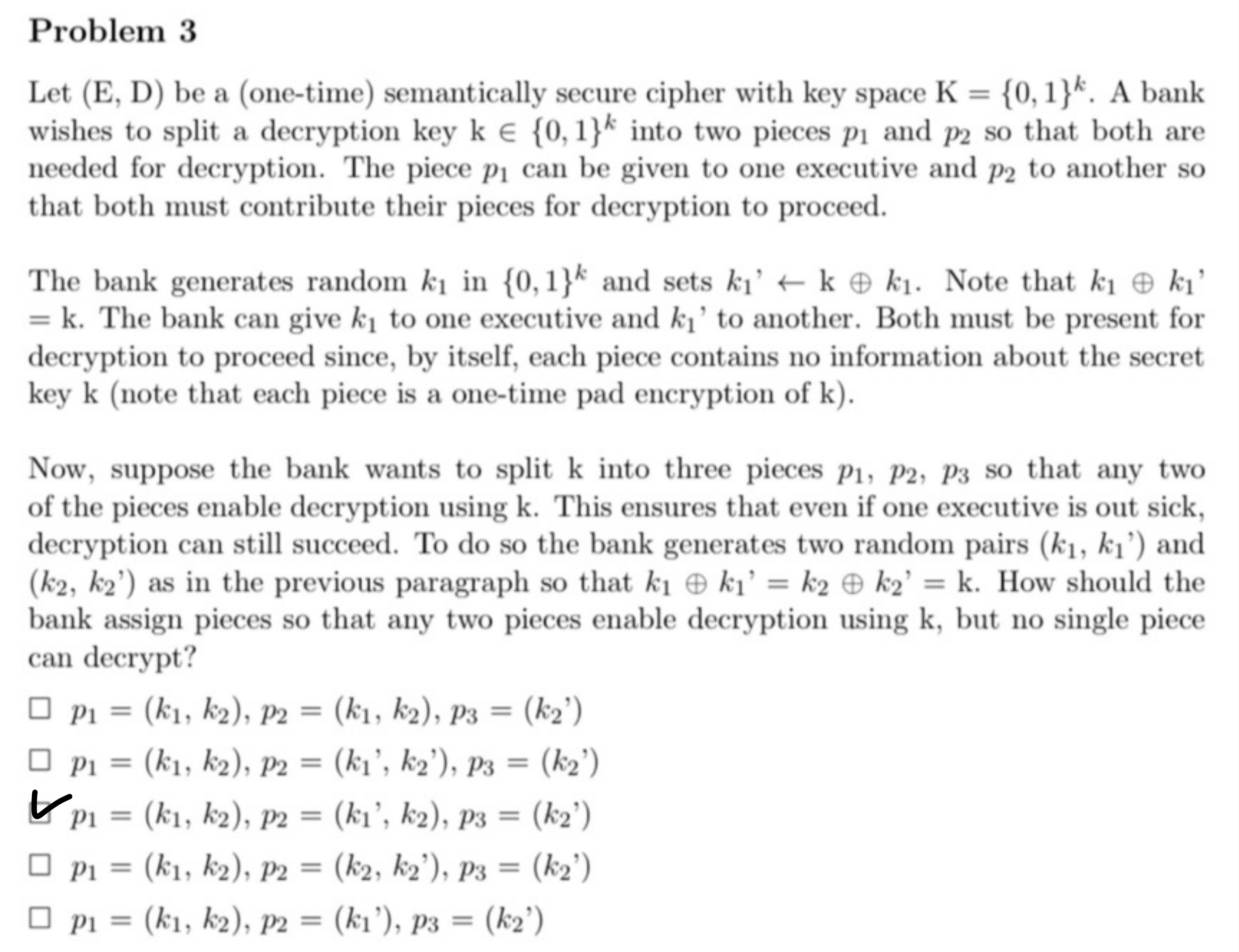
☑ G'(k_1, k_2) = G(k_1) ‖ G(k_2)

☑ G'(k) = reverse(G(k))

☑ G'(k) = $rotation_n$(G(k))

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.

The outputs of unmarked answer above will not be random.



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 ∈ $\{0,1\}^k$ 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' ← k ⊕ k_1. Note that k_1 ⊕ 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 ⊕ k_1' = k_2 ⊕ 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')

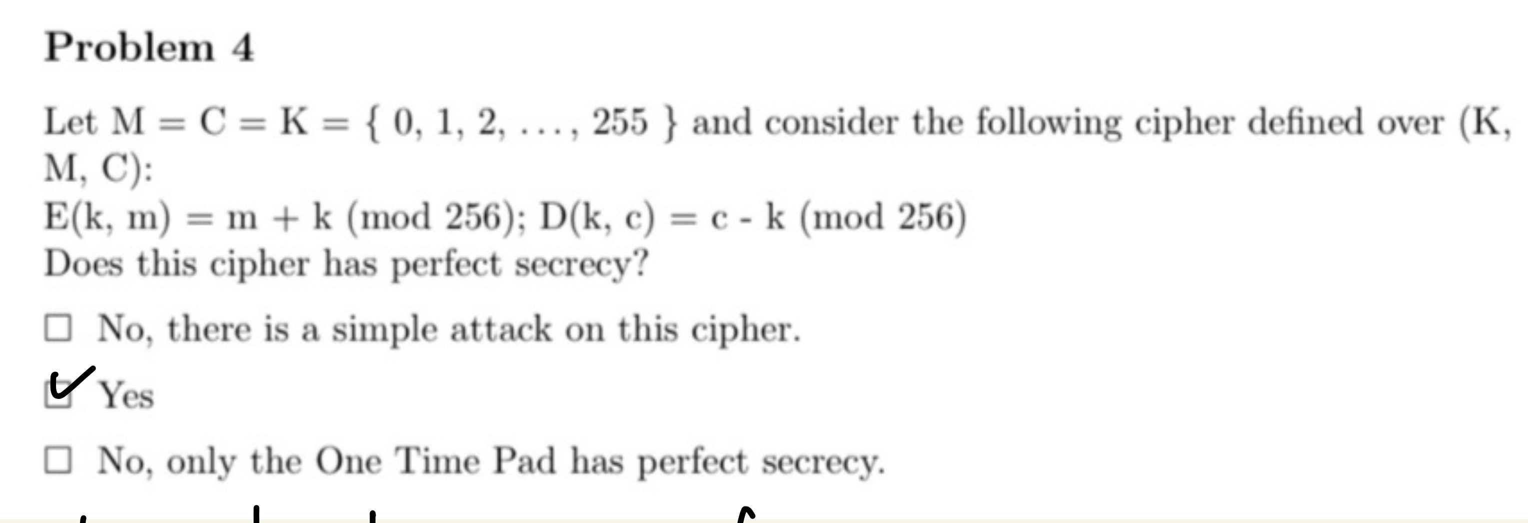
□ p_1 = (k_1, k_2), p_2 = (k_1', k_2'), p_3 = (k_2')

☑ 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')

This means that among the possible combinations of keys or people for decryption, combinations 1, 2, and 5 require more than two individuals to successfully decrypt the information. Combination 4 is unique in that it allows decryption with only person 2 present. Therefore, combination 3 is identified as the only solution where decryption can occur when any two individuals come together.



Problem 4

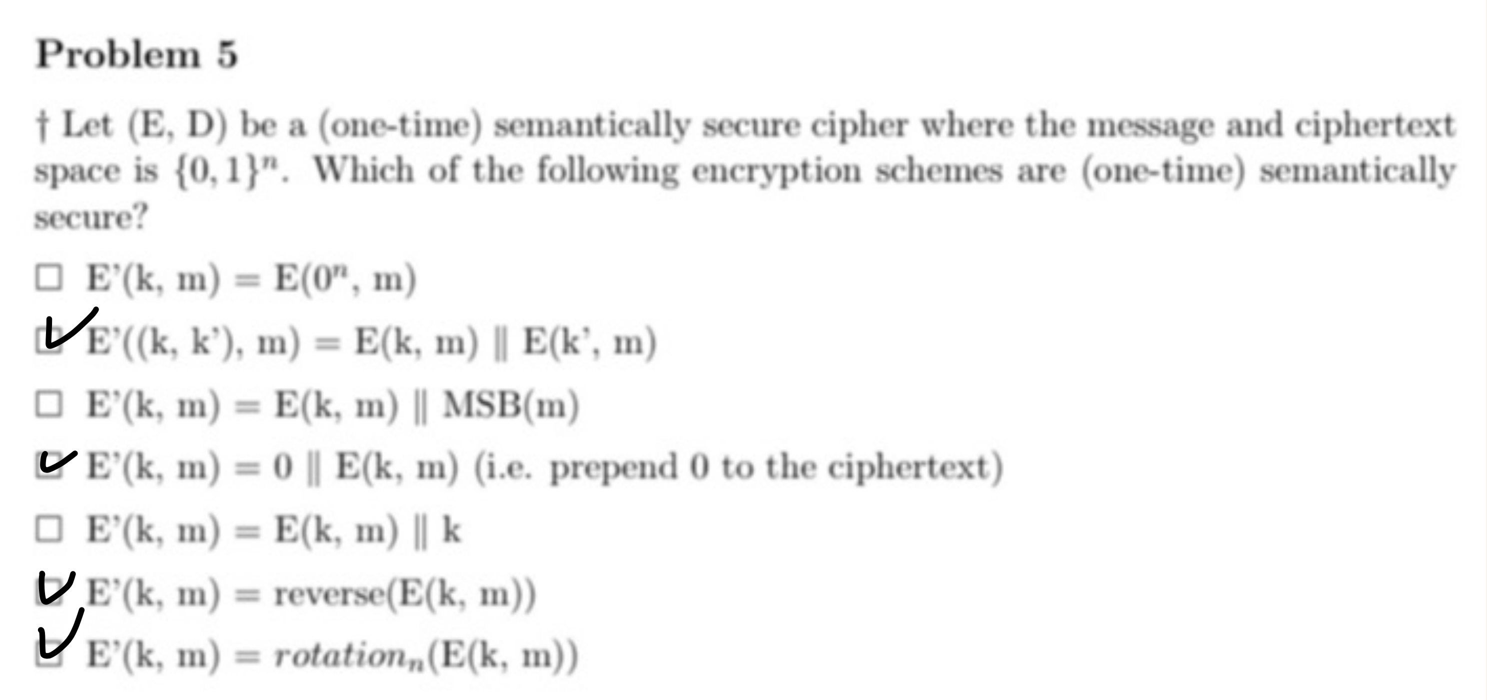
Let M = C = K = { 0, 1, 2, . . . , 255 } and consider the following cipher defined over (K, M, C):
E(k, m) = m + k (mod 256); D(k, c) = c - k (mod 256)
Does this cipher has perfect secrecy?

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

☑ Yes

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

The code has a 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)

☑ E'((k, k'), m) = E(k, m) ‖ E(k', m)

□ E'(k, m) = E(k, m) ‖ MSB(m)

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

□ E'(k, m) = E(k, m) ‖ k

☑ E'(k, m) = reverse(E(k, m))

☑ E'(k, m) = $rotation_n$(E(k, m))

1: To compromise semantic security, an attacker would request the encryption of sequences consisting entirely of zeros (0^n) and entirely of ones (1^n). The attacker can easily distinguish between EXP(0) and EXP(1) because they know the secret key, which in this context is assumed to be 0^n.

2: An attack on the modified encryption scheme E' implies a feasible attack on the original encryption scheme E.

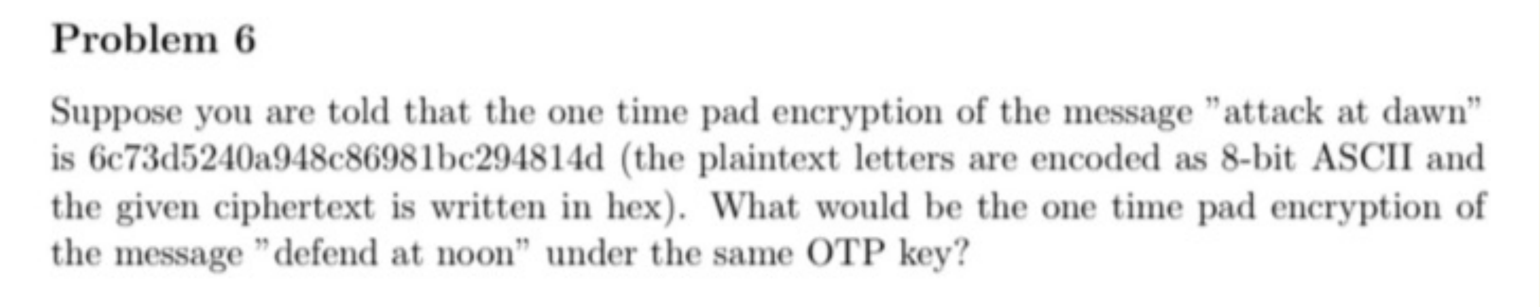
3: To undermine semantic security, an attacker would request the encryption of the sequence 0^n and the sequence followed by all zeros except for the last bit being one (0^n-11). The attacker can then differentiate between EXP(0) and EXP(1).

4: An attack on the modified encryption scheme E' translates into an attack on the original encryption scheme E.

5: To break semantic security, an attacker would extract the secret key from the challenge ciphertext itself and use this key to decrypt the challenge ciphertext. Essentially, this implies that any ciphertext reveals the secret key.

6: An attack on the modified encryption scheme E' leads to an attack on the original encryption scheme E.

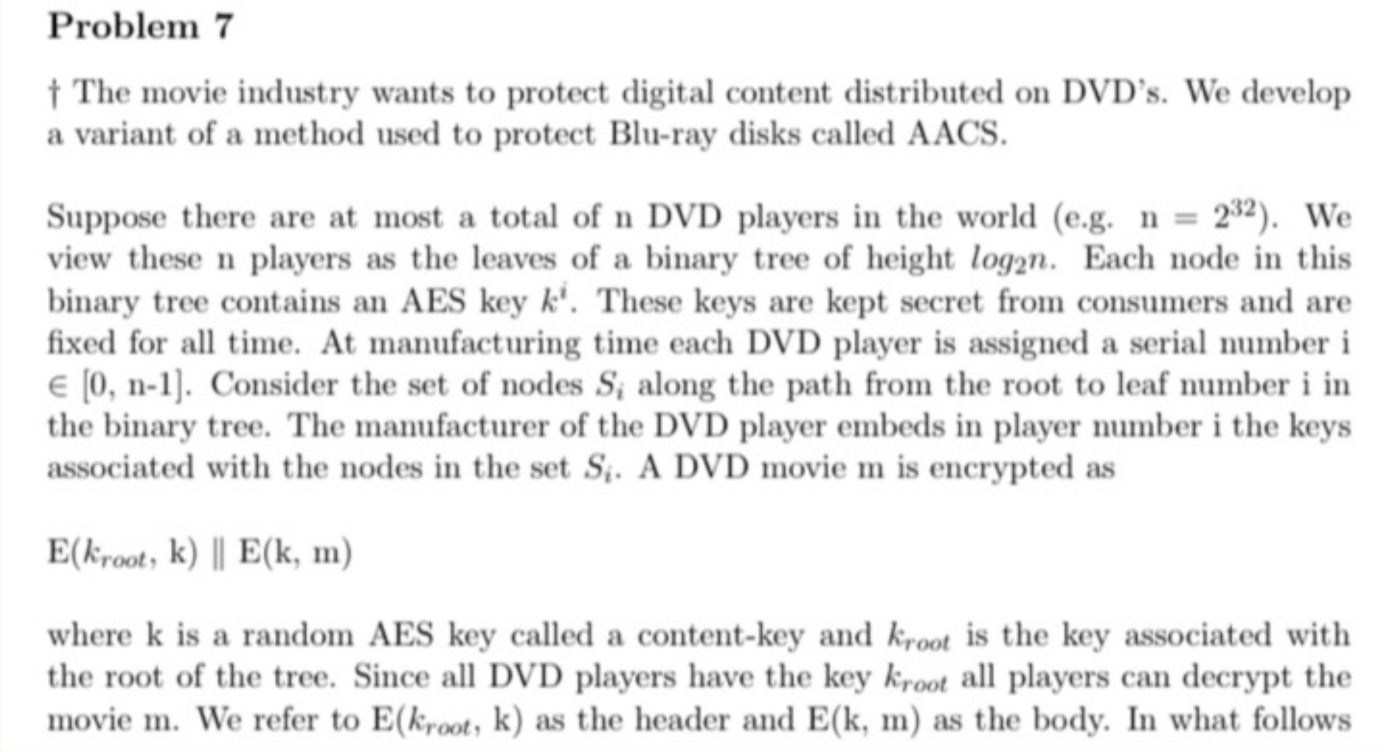
7: E'(k, m) = rotation_n(E(k, m)) is considered semantically secure if E is secure, as rotation doesn't compromise the ciphertext's security.



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?

The one-time pad encryption of the message "defend at noon" under the same OTP key would be 6962c720079b8c86981bc89a994d (in hex). After XORing the key with the new message, we get the correct answer



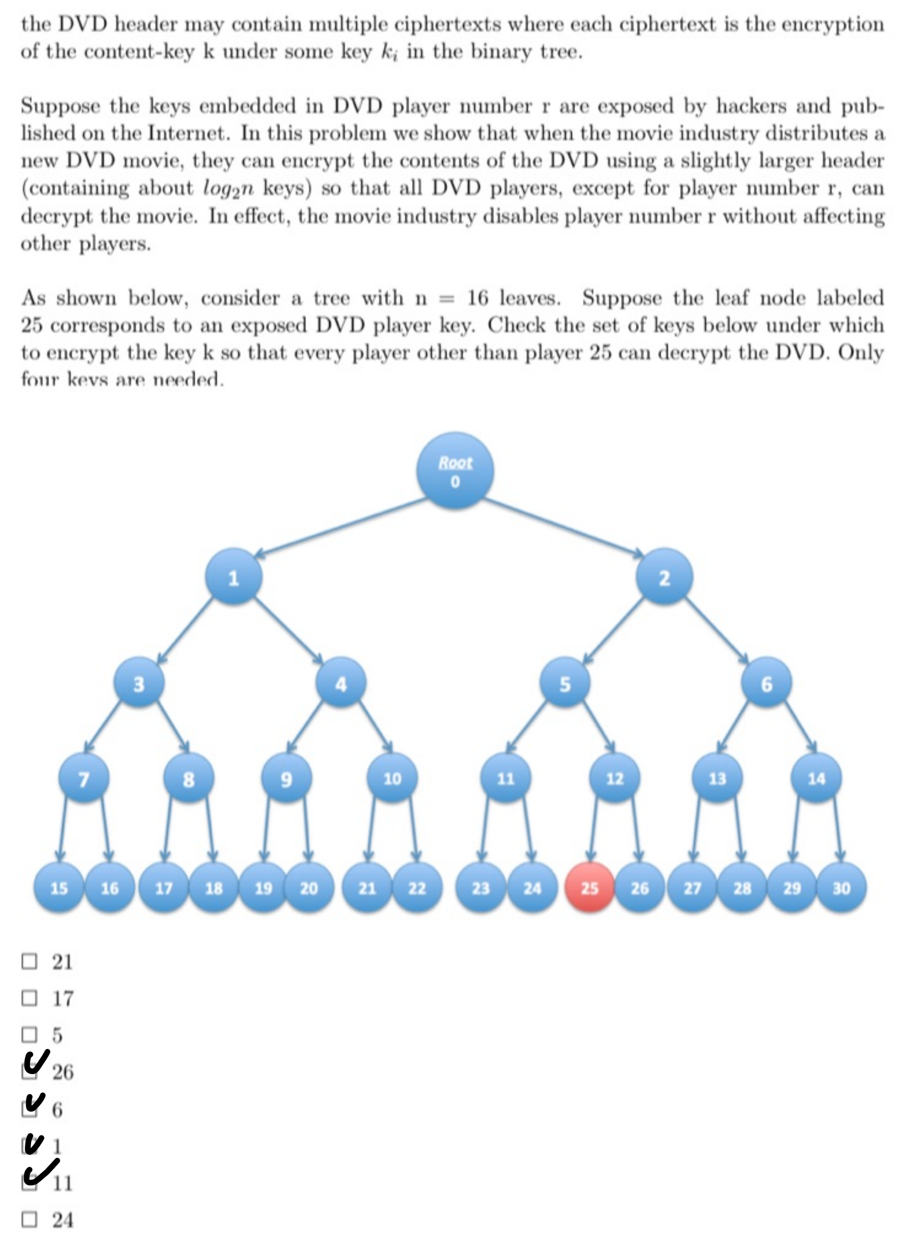
Problem 7

† 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.

Suppose there are at most a total of n DVD players in the world (e.g. n = 2^{32}). We view these n players as the leaves of a binary tree of height $log_2 n$. Each node in this binary tree contains an AES key k^i. These keys are kept secret from consumers and are fixed for all time. At manufacturing time each DVD player is assigned a serial number i ∈ [0, n-1]. Consider the set of nodes S_i along the path from the root to leaf number i in the binary tree. The manufacturer of the DVD player embeds in player number i the keys associated with the nodes in the set S_i. A DVD movie m is encrypted as

E(k_{root}, k) ‖ E(k, m)

where k is a random AES key called a content-key and k_{root} is the key associated with the root of the tree. Since all DVD players have the key k_{root} all players can decrypt the movie m. We refer to E(k_{root}, k) as the header and E(k, m) as the body. In what follows



the DVD header may contain multiple ciphertexts where each ciphertext is the encryption of the content-key k under some key k_i in the binary tree.

Suppose the keys embedded in DVD player number r are exposed by hackers and published on the Internet. In this problem we show that when the movie industry distributes a new DVD movie, they can encrypt the contents of the DVD using a slightly larger header (containing about $log_2 n$ keys) so that all DVD players, except for player number r, can decrypt the movie. In effect, the movie industry disables player number r without affecting other players.

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.

☐ 21

☐ 17

☐ 5

☑ 26

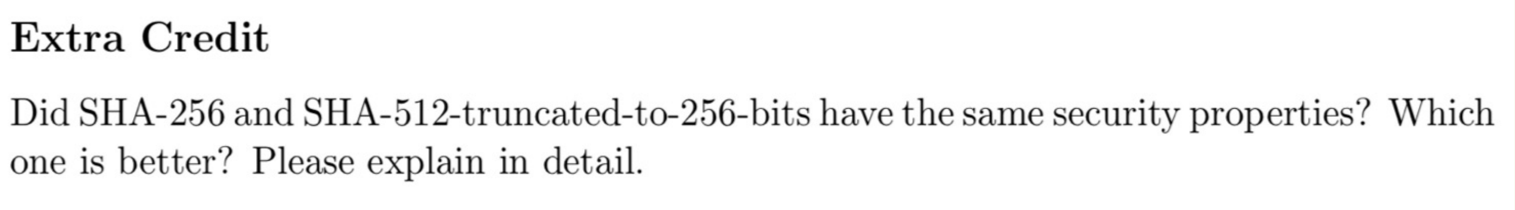
☑ 6

☑ 1

☑ 11

☐ 24

In other words, because key 25 is to the right of key 0, we can safely include all elements under key 1. Applying the same logic, albeit with a different parent, we can include keys 6 and 11. For the remaining leaf nodes, the only element we need to include is key 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 (often denoted as SHA-512/256) are both members of the SHA-2 family of cryptographic hash functions. While they have similar underlying structures, they differ in their output size and compression function.

1. Security Properties:

- Both SHA-256 and SHA-512/256 are designed to be collision-resistant, preimage-resistant, and second-preimage resistant, which are fundamental security properties for cryptographic hash functions.
- Collision resistance: This property ensures that it is computationally infeasible to find two different inputs that produce the same hash output.
- Preimage resistance: This property ensures that given a hash output, it is computationally infeasible to find any input that hashes to that output.
- Second-preimage resistance: This property ensures that given an input, it is computationally infeasible to find another input that hashes to the same output.

2. Output Size:

- SHA-256 produces a 256-bit hash output.
- SHA-512-truncated-to-256-bits (SHA-512/256) produces a 256-bit hash output, but it is derived from SHA-512, which produces a 512-bit hash output.

3. Performance:

- SHA-256 may have better performance in some scenarios due to its smaller output size. It requires fewer computations and may be faster on platforms where 64-bit arithmetic operations are not efficient.
- SHA-512/256 may have better performance in other scenarios, especially on platforms where 64-bit arithmetic operations are efficient, as it utilizes the SHA-512 compression function.

4. Recommendation:

- In terms of security, both SHA-256 and SHA-512/256 are considered to have strong security properties. They are widely used and trusted in practice.
- However, SHA-512/256 provides a higher level of security margin due to its larger internal state and longer compression function, which may offer better resistance against potential future attacks, especially in scenarios where quantum computing becomes a threat.
- Therefore, if performance is not a critical factor and there are no constraints on output size, SHA-512/256 may be preferred for its potential higher security margin.

In summary, both SHA-256 and SHA-512/256 offer strong security properties, but SHA-512/256 may provide a higher security margin due to its larger internal state. The choice between them should consider factors such as performance requirements, compatibility, and the desired level of security assurance.