# Quiz. 3

## 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:

Answer ) Compress then encrypt.

Explanation
The order is important; if the text is encrypted first, it will appear random, and compression will not operate properly. If compressed and then encrypted, the compression will work.

Compressing data before encrypting it is efficient because it reduces data size by eliminating redundancy, and the compressed data is subsequently encrypted. Encrypting before compressing is useless because encryption conceals data patterns, making it appear random and making it harder for compression algorithms to discover redundancies and reduce size. To be as efficient as possible, always compress before encrypting.

## Problem 2

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

Answer)
$G'(k) = G(k \oplus 1^s)$
$G'(k_1, k_2) = G(k_1) \| G(k_2)$
$G'(k) = \text{reverse}(G(k))$

Explanation
$G'(k) = G(k \oplus 1^s)$ is a secure PRG because XORing $k$ with a string of 1's and then applying $G$ does not affect the security properties of $G$.

$G'(k_1, k_2) = G(k_1) \| G(k_2)$ is secure, this would hold true particularly if $k1$ and $k2$ are independent and uniformly distributed, and $G$ maintains its pseudorandom properties when its output is extended in this way.

$G'(k) = \text{reverse}(G(k))$ ：In this construction, **G'** is defined to be the reverse of the output of **G** given a seed **k**. Since the output of **G** is pseudorandom and reversing a string does not introduce any pattern or predictability, this construction should maintain the pseudorandomness. Therefore, **G'** should also be a secure PRG.

*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.*
*"$\text{rotation}_n(x)$" 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 ∈ $\{0, 1\}^k$ 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 piece so that any two pieces enable decryption using k, but no single piece can decrypt?

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

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

*Answer)* $p_1 = (k_1, k_2), p_2 = (k_1', k_2), p_3 = (k_2')$

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

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

Explanation

When 2 persons come together, combinations 1, 2, and 5 cannot be decrypted. Combination 4 decrypts when only p2 is present. Thus, combination three is the 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.
Answer )Yes

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

Explanation

This code has a perfect secrecy because it adheres to Shannon's Criteria: The cipher satisfies Shannon's perfect secrecy as the ciphertext does not reveal any information about the plaintext without the key.

## 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)

Answer 1) E' ((k, k'), m) = E(k, m) || E(k', m)

□ E' (k, m) = E (k, m) || MSB(m)

Answer 2) E' (k, m) = 0 || E (k, m) (i.e. prepend 0 to the ciphertext)

□ E' (k, m) = E (k, m) || k

Answer3) E' (k, m) = reverse (E(k, m))

□ E' (k, m) = $rotation_n$(E(k, m))

Explanation

1) To break semantic security, an attacker would request the encryption of $0^n$ and $1^n$ and could readily distinguish EXP (0) from EXP (1) because it knows the secret key, $0^n$.
2) An attack on E' produces an attack on E.
3) To break semantic security, an attacker would request the encryption of $0^n$ and $0^{n-1}1$ and be able distinguish between EXP (0) and EXP (1).
4) an attack on E' produces an attack on E.
5) To break semantic security, an attacker would read the secret key from the challenge ciphertext and use it to decrypt it. Basically, any ciphertext contains the secret key
6) an attack on E' produces an attack on E.
7) This scheme rotates the ciphertext by n positions.

## 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?

Answer) 6c73d5240a948c86981bc2808548L

Explanation

Given the original message and encoded cypher, we can determine that key=d07a14569fface7ec3ba6f5f623L. XORing the key with the new message gets the right 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
$\in [0, n-1]$. 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

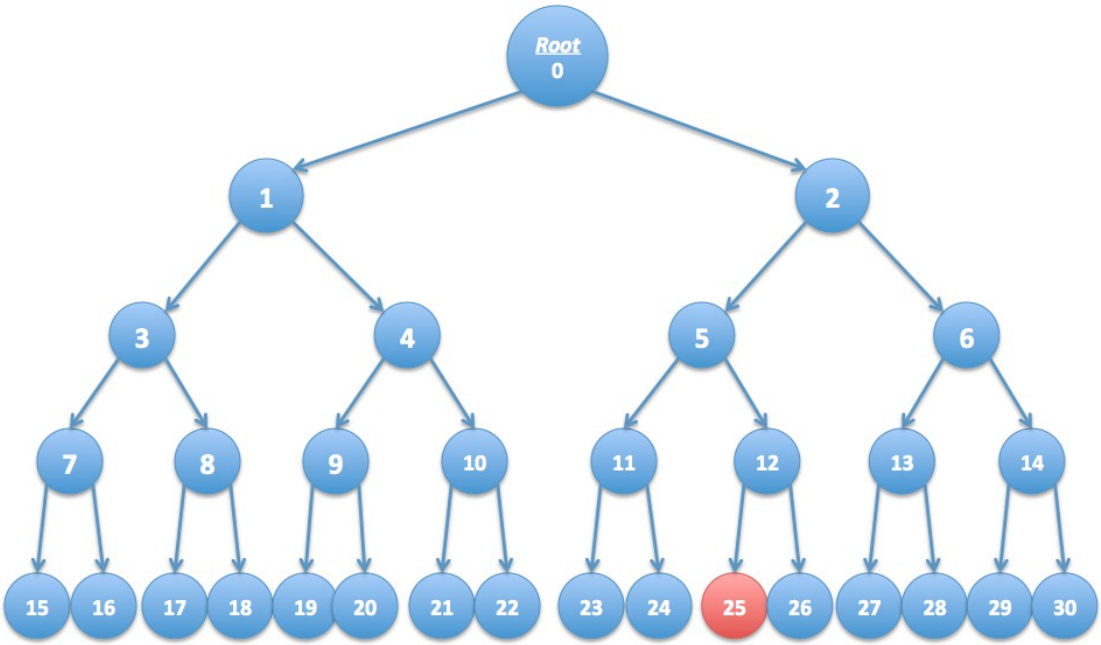
$$E(k_{root}, k) \| E(k, m)$$

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 pub- lished 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

Answer )26
Answer )6
Answer )1
Answer )11

- ☐ 24

Explanation
Because key 25 is to the right of key 0, we can safely include every element under key one. With the same logic (but a different parent), we can include 6 and 11. We simply need to include 26 in the remaining leaves.

## 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.

For most practical purposes, SHA-256 and SHA-512-truncated-to-256-bits have similar security standards because they generate a 256-bit hash. However, SHA-512-truncated-to-256-bits may have a slight security advantage because to its larger internal state and resistance to certain attacks. On 64-bit systems, SHA-512 can outperform SHA-256, however on 32-bit platforms, SHA-256 is usually preferred for its efficiency.