AIAA5033 - Assignment#1 - Spring 2025

Name: Linxiao Cao; Student ID: 50031322

Problem 1

Using the English-language shift cipher, what is the encryption of "good" using the key "b"?

Solution.
Convert each letter to its numerical equivalent (0-based):

$$\begin{aligned} g &= 6 \\ o &= 14 \\ d &= 3 \end{aligned}$$

Apply the shift (1 position):

$$\begin{aligned} 6 + 1 &= 7 = h \\ 14 + 1 &= 15 = p \\ 3 + 1 &= 4 = e \end{aligned}$$

Combine the shifted letters: **hppe**

$$\begin{aligned} x &= 1 \\ y &= 2 \end{aligned}$$

Problem 2

Using the English-language shift cipher, what is the plaintext could correspond to ciphertext “AZC”? Please provide details for how you obtain the solution.

Solution.

To determine the plaintext corresponding to the ciphertext ”AZC” using a shift cipher, we need to consider all possible shift values (from 0 to 25) and see which one results in a meaningful English word. Shift of 0: AZC (not meaningful)

Shift of 1: ZYB (not meaningful)
Shift of 2: YXA (not meaningful)
Shift of 3: XWZ (not meaningful)
Shift of 4: WVY (not meaningful)
Shift of 5: VUX (not meaningful)
Shift of 6: UTW (not meaningful)
Shift of 7: TSV (not meaningful)
Shift of 8: SRU (not meaningful)
Shift of 9: RQT (not meaningful)
Shift of 10: QPS (not meaningful)
Shift of 11: POR (not meaningful)
Shift of 12: ONQ (not meaningful)
Shift of 13: NMP (not meaningful)
Shift of 14: MLO (not meaningful)
Shift of 15: LKN (not meaningful)
Shift of 16: KJM (not meaningful)
Shift of 17: JIL (not meaningful)
Shift of 18: IHK (not meaningful)
Shift of 19: HGJ (not meaningful)
Shift of 20: GFI (not meaningful)
Shift of 21: FEH (not meaningful)
Shift of 22: EDG (not meaningful)
Shift of 23: DCF (not meaningful)
Shift of 24: CBE (not meaningful)
Shift of 25: BAD (meaningful)

When we apply a shift of 25 to the ciphertext ”AZC”, we get the plaintext ”BAD”, which is a meaningful English word.

Answer: The plaintext corresponding to the ciphertext ”AZC” is ”BAD” when using a shift of 25 (or equivalently, a shift of -1).

Problem 3

Let $G(x) = x||parity(x)$. Which of the following proves that G is NOT a pseudo-random generator?

(a) G is not expanding

(b) Consider the following distinguisher D : $D(y)$ outputs 1 if the first bit of y is 1

(c) Consider the following distinguisher D : $D(y)$ outputs 1 if the parity of the first n bits of y is 0

(d) Consider the following distinguisher D :$D(y)$ outputs 1 if the parity of the first n bits of y is equal to the last bit of y

Solution. To determine which option proves that $G(x) = x||\text{parity}(x)$ is not a pseudo-random generator (PRG), we analyze each option:

Option (a): G is not expanding

- $G(x)$ takes an input x of length n and outputs x concatenated with its parity, resulting in an output of length $n + 1$. Since $n + 1 > n$, G is expanding. Thus, (a) is incorrect.

Option (b): Distinguisher $D(y)$ outputs 1 if the first bit of y is 1

- The first bit of $G(x)$ is uniformly random if x is random. Similarly, the first bit of a truly random string y is also uniformly random. This distinguisher cannot distinguish $G(x)$ from random, so (b) is incorrect.

Option (c): Distinguisher $D(y)$ outputs 1 if the parity of the first n bits of y is 0

- For $G(x)$, the parity of the first n bits (which is x) is the last bit. The probability that this parity is 0 is 1/2, same as for a random string. This distinguisher does not have an advantage, so (c) is incorrect.

Option (d): Distinguisher $D(y)$ outputs 1 if the parity of the first n bits of y is equal to the last bit

- For $G(x)$, the last bit is exactly the parity of the first n bits, so $D(y)$ outputs 1 with probability 1. For a random string, the parity of the first n bits and the last bit are independent, so the probability they are equal is 1/2. This gives the distinguisher an advantage of 1/2, proving G is not a PRG. **Thus, (d) is correct.**

Problem 4

Consider encryption scheme $\prod$ that encrypts a 2n-bit message using an n-bit key via $Enc_k(m_a|m_b) = k \oplus m_a|k \oplus m_b$. Which of the following could be the start of a proof that $\prod$ is not EAV-secure?

(a) Consider an attacker A who outputs $m_0 = 0^n0^n and m_1 = 1^n1^n$...

(b) Consider an attacker A who outputs $m_0 = 0^n0^n and m_1 = 0^n1^n$...

(c) Consider an attacker A who outputs $m_0 = 0^n1^n and m_1 = 1^n0^n$...

(d) $\prod$ is EAV-secure, since it uses the one-time pad

Please provide proper justification and analysis for all of items and provide your answer to the question.

Solution. To determine which option could be the start of a proof that the encryption scheme $\prod$ is not EAV-secure, we analyze each option in detail.

Encryption Scheme Analysis: The encryption scheme $\prod$ is defined as:

$$Enc_k(m_a \mid m_b) = (k \oplus m_a) \mid (k \oplus m_b)$$

where: - k is an n-bit key. - m_a and m_b are each n-bit message halves. - $\oplus$ denotes bitwise XOR.

For an attacker to break EAV-security (semantic security against eavesdroppers), they must construct a distinguishing attack: given ciphertexts, they should determine which plaintext message was encrypted.

Option (a): $m_0 = 0^n0^n$ and $m_1 = 1^n1^n$

- Encryption of m_0:

$$Enc_k(0^n \mid 0^n) = (k \oplus 0^n) \mid (k \oplus 0^n) = k \mid k$$

- Encryption of m_1:

$$Enc_k(1^n \mid 1^n) = (k \oplus 1^n) \mid (k \oplus 1^n)$$

- Here, both encryptions result in two identical halves of the ciphertext. Since this structure remains the same for both messages, an attacker cannot distinguish between them. Thus, this choice is not a good starting point for proving insecurity.

Option (b): $m_0 = 0^n0^n$ and $m_1 = 0^n1^n$

- Encryption of m_0:

$$Enc_k(0^n \mid 0^n) = k \mid k$$

- Encryption of m_1:

$$Enc_k(0^n \mid 1^n) = k \mid (k \oplus 1^n)$$

- Here, the ciphertext halves for m_0 are equal $(k \mid k)$, while for m_1, the halves are different

$(k \mid k \oplus 1^n)$.

- This provides a clear distinguishing attack: the attacker can simply check whether the two halves of the ciphertext are equal. If they are, the plaintext was m_0; otherwise, it was m_1. Since this method successfully distinguishes between two chosen messages, it proves that the scheme is not EAV-secure.

⇒ **(b) is a correct starting point for proving insecurity.**

Option (c): $m_0 = 0^n 1^n$ and $m_1 = 1^n 0^n$

- Encryption of m_0:

$$Enc_k(0^n \mid 1^n) = k \mid (k \oplus 1^n)$$

- Encryption of m_1:

$$Enc_k(1^n \mid 0^n) = (k \oplus 1^n) \mid k$$

- Here, the ciphertext halves are different in both cases. However, distinguishing between these cases requires deeper analysis, and the structure does not immediately expose an obvious attack like in option (b). While this option may still lead to insecurity, it is not as clear and immediate as (b).

Option (d): Claiming $\prod$ is EAV-secure because it "uses the one-time pad"

This claim is incorrect because: - A true one-time pad encrypts each message with a fresh, random key. Here, the same key k is reused for multiple messages, violating the principles of one-time pad security. - The encryption function exhibits structure that allows an attacker to distinguish messages (as shown in option (b)). - The scheme fails indistinguishability, meaning it is not EAV-secure. Thus, (d) is incorrect.

Final Answer Since **option (b)** provides a clear distinguishing attack, it is the best choice for starting a proof that $\prod$ is not EAV-secure