Assignment 2

Instructions

This assignment consists of 4 problems. Problem 1 must be solved and submitted via the provided Google Form, while Problems 2, 3, and 4 should be solved as handwritten solutions and submitted in class on due date. For the handwritten problems, write your name, roll number, and ensure clarity with arguments and neatness in your solutions. Late submissions will not be accepted. All steps of your calculations or reasoning must be shown. Failure to follow these instructions may result in a deduction of marks. If you have any questions, comment on following thread.

Problem 1: Multi time xor

The ciphertext is like previous assignment mapped to your serial number. You are provided with a ciphertext stored in the file ciphertext_num.txt on given github link.

The ciphertext is generated as follows:

The plaintext is appended with a random string and converted to UTF-8 bytes. It is encrypted using a secret key similar to a One-Time Pad (OTP), but with a Vigenère-like approach, where the same short key is reused for the entire plaintext, using XOR as the operation. Finally, the ciphertext is converted from bytes to hexadecimal format.

Your task is to find following.

1. Key Length Guessing: Identify the most likely lengths of the encryption key.

2. Key Recovery: Recover the encryption key. Decryption: Use the recovered key to decrypt the ciphertext and retrieve the original plaintext.

3. Random String Extraction: Extract the 25-character random string appended to the plaintext.

Fill the given google form and submit the guessed key length, recovered key and extracted random string.

Problem 2: Perfect Security

A deterministic symmetric encryption scheme E specifies a pair of algorithms $E = (\text{Enc}, \text{Dec})$ with three associated sets: the key space K, the message space M, and the ciphertext space C.

- The deterministic encryption algorithm $\text{Enc} : K \times M \to C$ takes as input a key $k \in K$ and a message $m \in M$, and produces a ciphertext $c \in C$.
- The deterministic decryption algorithm $\text{Dec} : K \times C \to M$ takes as input a key $k \in K$ and a ciphertext $c \in C$, and returns a message $m \in M$.

We require that E satisfies the *decryption correctness* property, meaning that any ciphertext produced by Enc is correctly decryptable using Dec. Formally, for all keys $k \in K$ and for all messages $m \in M$, it holds that:

$$\text{Dec}(k, \text{Enc}(k, m)) = m.$$

We say that encryption scheme E is *perfectly secure* if for all $m_0, m_1 \in M$, and all $c \in C$, we have:

$$\Pr[\text{Enc}(k, m_0) = c] = \Pr[\text{Enc}(k, m_1) = c],$$

where k is a random variable uniformly distributed over K. Intuitively, this means that an E ciphertext leaks no information about the encrypted message. An alternative definition of perfect security (for one-time pads) was provided in class.

(a) Devise an encryption scheme E_{90} such that:

- (1) Given an encryption of any message, an adversary can figure out 90% of the secret key (i.e., the ciphertext leaks this information), but
- (2) The scheme is still perfectly secure, despite 90% of the key being revealed.

Prove that the scheme is secure and that it is correct. When constructing your encryption scheme, you should define algorithms Enc and Dec as well as the associated sets K, M, and C.

(b) Devise an encryption scheme E_{broken} such that:

- (1) Given an encryption of any message, an adversary learns nothing about the secret key, but
- (2) The scheme is completely broken (as in, given the ciphertext, an adversary can completely recover the plaintext).

(c) Build an encryption scheme E_1 such that an adversary can recover the first bit of any message $m \in M$ from its encryption, but E_1 is nonetheless perfectly secure.
Now, in addition to the above, let $M = \{0,1\}^n$ for any $n \geq 1$; show that any encryption scheme E with message space M is not perfectly secure (if the first bit of any message $m \in M$ can be recovered from its encryption, as per above).

(d) Let E be an encryption scheme such that $|K| \neq 0$ and $C = \{\text{Enc}(k,m) : k \in K, m \in M\}$. Show that if E is perfectly secure then $|M| \leq |C| \leq |K|$. Explain why some assumptions about K and C (such as stated above) are necessary to prove this claim.

Problem 3: PRGs

Let $\ell, n \in \mathbb{N}$ such that $\ell < n$. Let $G : \{0,1\}^\ell \to \{0,1\}^n$ be a secure PRG (pseudorandom generator). For each of the new constructions below, determine if it is also a secure PRG. If you believe that a derived generator is not secure, then provide an attack. If you believe that a derived generator is secure, then try to justify that as formally as possible (if you have seen proofs by reduction, that is a useful technique here). In the following, s, s_1, and s_2 are strings in $\{0,1\}^\ell$, and $\|$ denotes string concatenation. The bitwise XOR operation is denoted by $\oplus$. The bitwise AND operation is denoted by $\wedge$. Some of the derived generators have domains or ranges that differ from that of G.

(a) $G_1(s) := G(s) \oplus 1^n$, where 1^n is the bit-string consisting of n 1s, for example $1^4 = 1111$.

(b) $G_2(s) := G(s)[0 \ldots n-2]$. Here we treat the output string $G(s)$ as an array and use vector notation to indicate that we truncate the result by removing the last bit. For example, $\text{abcd}[0 \ldots 2] = \text{abc}$.

(c) $G_3(s) := G(s) \| G(s)$. Note that the range of G_3 is $\{0,1\}^{2n}$.

(d) $G_4(s_1 \| s_2) := s_1 \| G(s_2)$. Note that G_4 is a function from $\{0,1\}^{2\ell}$ to $\{0,1\}^{\ell} \times \{0,1\}^{n}$.

(e) $G_5(s) := G(s) \| G(G(s)[0 \ldots \ell - 1])$. Note that the range of G_5 is $\{0,1\}^{\ell} \times \{0,1\}^{n}$.

Problem 4: Encryption with a Deck of Cards

Alice, Bob, and Eve are playing a card game. Alice shuffles a deck of cards and deals it all out to herself and Bob (each gets half of the 52 distinct cards). Alice now wishes to send a secret message m to Bob by saying something aloud. Everybody is in the same room, and eavesdropper Eve is listening in: she hears everything Alice says (but Eve cannot see the face of Alice's and Bob's cards).

(a) Suppose Alice's message m is a string of 48 bits. Describe how Alice can communicate m to Bob in such a way that Eve will have no information about what m is.
Note: Alice and Bob are allowed to devise a public strategy together before the cards are dealt.

(b) Now suppose that Alice's message m is 49 bits. Show that there exists no protocol that allows Alice to communicate m to Bob in such a way that Eve will have no information about m.