EE 720: Introduction to Number Theory and Cryptography (Autumn 2023)
Instructor: Saravanan Vijayakumaran
Indian Institute of Technology Bombay

Mid-semester Exam: 24 points   Date: September 22, 2023

1. [6 points] Prove that the one-time pad is
   (a) perfectly secret.
   (b) not CPA-secure.

   Recall the definition of the one-time pad. For an integer $n > 0$, set message space $\mathcal{M}$, key space $\mathcal{K}$, and ciphertext space $\mathcal{C}$ all equal to $\{0,1\}^n$. `Gen` chooses key k uniformly from $\mathcal{K}$. Given k and mesage $m \in \{0,1\}^n$, `Enc` computes $c := k \oplus m$. Given $k, c \in \{0,1\}^n$, `Dec` computes $m := k \oplus c$.

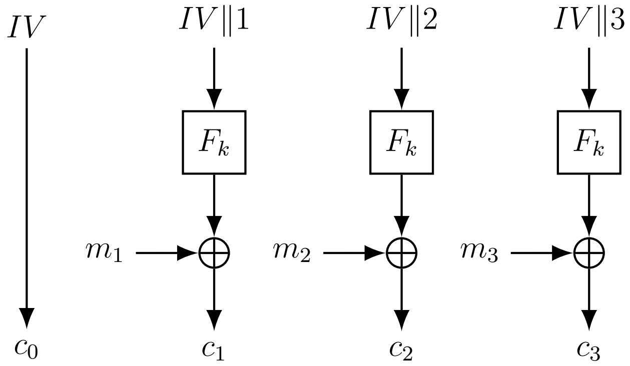
2. [6 points] Alice has a length-preserving pseudorandom function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$. She wants to encrypt messages of length $2n$. Let $m \in \{0,1\}^{2n}$ denote the message. Let $m_1 \in \{0,1\}^n$ denote the first n bits of m and let $m_2 \in \{0,1\}^n$ denote the last n bits of m. Alice uses the encryption scheme $\Pi = (\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$ where:

   - `Gen`: Key k is chosen uniformly from $\{0,1\}^n$.
   - `Enc`: The message space $\mathcal{M} = \{0,1\}^{2n}$. A string r is chosen uniformly from $\{0,1\}^{n-1}$ and the ciphertext $c \in \{0,1\}^{3n-1}$ corresponding to $m = (m_1, m_2) \in \{0,1\}^{2n}$ is given by
   $$c := \langle r, m_1 \oplus F_k(0\|r), m_2 \oplus F_k(1\|r) \rangle.$$
   Here $\|$ is the string concatenation operator.
   - `Dec`: Given key k and ciphertext $c = \langle r, c_1, c_2 \rangle \in \{0,1\}^{3n-1}$, the message $m = (m_1, m_2)$ is decrypted using $m_1 = c_1 \oplus F_k(0\|r)$ and $m_2 = c_2 \oplus F_k(1\|r)$.

   Prove that Alice's scheme is **CPA-secure**. You **cannot use** the CPA-security of CTR mode in your proof.

3. Recall that the PKCS #7 padding scheme is used to pad a message $\vec{x}$ having length some integral number of bytes into a *encoded data* $\vec{m}$ having length jL bytes where L is the block length in bytes. The number of bytes which are appended to $\vec{x}$ to get $\vec{m}$ is b where $1 \le b \le L$. Each of these padding bytes is equal to the byte representation of the integer b. Assume that $L < 256$ so b can fit in a single byte.

   Suppose the encoded data $\vec{m}$ has length $3L$ bytes, i.e. $\vec{m} = (m_1, m_2, m_3)$ where $|m_i| = L$ bytes for $i = 1, 2, 3$. Now suppose the encoded data is encrypted using CTR mode where F is a length-preserving pseudorandom function as shown below. The input and output lengths of F_k are both equal to $n = 8L$ bits. Here the value IV is uniformly chosen from $\{0,1\}^{\frac{3n}{4}}$.



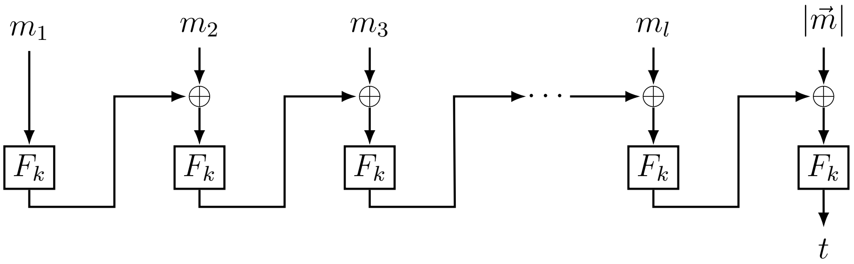
Suppose an adversary has access to a padding oracle. On input some ciphertext block $\vec{c} = (c'_0, c'_1, c'_2, c'_3)$, the padding oracle only returns a message from the set $\{\mathsf{ok}, \mathsf{padding\_error}\}$. The ok is returned when there is no padding error in the encoded data $\vec{m}'$ obtained from $\vec{c}$.

(a) [2 points] Describe a procedure by which the adversary can recover the **length** b of the padding in the encoded data $\vec{m}$.

(b) [2 points] Describe a procedure by which the adversary can recover the **last** byte in the encoded data block m_2. For example, if $L = 3$ and $m_2 =$ `0x01 0x07 0x20`, then `0x20` is the last byte of m_2.

(c) [2 points] Describe a procedure by which the adversary can recover the **first** byte in the encoded data block m_2. By first byte, we mean the most significant byte. For example, if $L = 3$ and $m_2 =$ `0x01 0x07 0x20`, then `0x01` is the first byte of m_2.

4. [6 points] Consider the modification of the CBC-MAC where the message length is appended to the *end of the message*. Let F be a length-preserving pseudorandom function of length n and let $\vec{m} = [m_1, m_2, \ldots, m_l]$ be a message of length ln where each $m_i \in \{0,1\}^n$. Assume that the the message length in bits satisfies $|\vec{m}| = ln < 2^n$. This allows us to represent $|\vec{m}|$ using n bits. The MAC tag t on $\vec{m}$ is calculated as shown in the below figure.



Show that this construction is an **insecure** MAC for **arbitrary-length** messages. By arbitrary-length messages, we mean that the sender and receiver do not fix l beforehand. The sender can send message-tag pairs $(\vec{m}, t)$ where the number of blocks l in $\vec{m}$ can vary.

Hint: Choose $m_1, m_2 \in \{0,1\}^n$ where $m_1 \neq m_2$. Query the MAC oracle on m_1 first and then on m_2. Then choose an $m_3 \in \{0,1\}^n$ and query the oracle on a message $m_1 \| n \| m_3$. Use the query responses to construct a tag t on a message $a\|b\|c \in \{0,1\}^{3n}$ which is not in the query set.