

**Problem 1):** We know that the expression

$$\Pr[M = m | C = c] = \Pr[M = m], \quad (1.1)$$

which holds for some encryption scheme  $(\text{Gen}, \text{Enc}, \text{Dec})$ . We apply Bayes' formula to the left-hand-side of expression 1.1 to yield

$$\Pr[M = m | C = c] = \frac{\Pr[M = m, C = c]}{\Pr[C = c]} \quad (1.2)$$

Since the  $\Pr[M = m, C = c]$  term in expression 1.2 represents a Joint Probability Distribution, we also have

$$\Pr[M = m, C = c] = \Pr[M = m] \Pr[C = c | M = m]$$

This allows the result in expression 1.2 to be equivalently expressed as

$$\Pr[M = m | C = c] = \frac{\Pr[M = m] \Pr[C = c | M = m]}{\Pr[C = c]}$$

which is divided by  $\Pr[M = m]$  to give

$$\frac{\Pr[M = m | C = c]}{\Pr[M = m]} = \frac{\Pr[C = c | M = m]}{\Pr[C = c]} \quad (1.3)$$

By noting that expression 1.1 implies

$$\frac{\Pr[M = m | C = c]}{\Pr[M = m]} = 1,$$

the result in expression 2.2 becomes

$$\frac{\Pr[C = c | M = m]}{\Pr[C = c]} = 1$$

Multiplying this result by  $\Pr[C = c]$  gives

$$\Pr[C = c | M = m] = \Pr[C = c], \quad (1.4)$$

thereby proving that  $\Pr[M = m | C = c] = \Pr[M = m]$  implies  $\Pr[C = c | M = m] = \Pr[C = c]$ .

□

**Problem 2):** For a message of length  $l \in \mathbb{Z}^+$ , a one-time pad will be associated with three separate set spaces and three different algorithms.

The set spaces of a one-time pad are the key space  $\mathcal{K}$ , the message space  $\mathcal{M}$ , and the cipher-text space  $\mathcal{C}$ . Additionally these spaces are such that

$$\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0, 1\}^l, \quad (2.1)$$

where  $\{0, 1\}^l$  is the set of all binary strings having length  $l$ . Formally, the set  $\{0, 1\}^l$  is defined

$\{0, 1\}^l \equiv \{n_1, n_2, \dots, n_l\}$  where  $\forall i \in [1, l], n_i \in [0, 1] \subset \mathbb{Z}$  (i.e. either  $n_i = 0$ , or  $n_i = 1$  for every element in the set)<sup>1</sup>.

A one-time pad is also associated with the algorithms

- The key-generation algorithm, denoted Gen
- The encryption algorithm, denoted Enc
- The decryption algorithm, denoted Dec

---

<sup>1</sup>This also holds for  $\mathcal{M}$  and  $\mathcal{C}$

The purpose of  $\text{Gen}$  is to generate a key for encrypting and decrypting our message, where  $k$  denote this key  $k$  and say that  $k \in \mathcal{K}$ . We say that  $\text{Gen}$  works by choosing a string from  $\mathcal{K}$  according to the uniform distribution. From this choice of distribution, it follows that each possible key will be chosen with probability  $2^{-l}$ .

Before we describe  $\text{Enc}$  and  $\text{Dec}$  algorithms we must define a bit-wise **XOR** on two binary strings of equal length. Let  $a$  and  $b$  be any two binary strings such that  $a, b \in \{0, 1\}^l$ . Additionally, let us express these strings by

$$a \equiv \{a_1, a_2, \dots, a_l\}$$

and

$$b \equiv \{b_1, b_2, \dots, b_l\},$$

respectively. The bit-wise **XOR** of  $a$  and  $b$  is denoted by  $a \oplus b$  and expressed as the binary string

$$a \oplus b \equiv \{a_1 \oplus b_1, a_2 \oplus b_2, \dots, a_l \oplus b_l\}$$

The elements of this binary string, the  $a_i \oplus b_i$ , are binary bits and are defined  $\forall i \in [1, l]$  to be the traditional bit-level **XOR** of  $a_i$  and  $b_i$ , denoted  $a_i \oplus b_i$ . We use the truth table

$a_i$	0	0	1	1
$b_i$	0	1	0	1
$a_i \oplus b_i$	0	1	1	0

to express the definition of the transitional bit-level **XOR** and continue to the definitions of the  $\text{Enc}$  and  $\text{Dec}$  algorithms.

The  $\text{Enc}$  algorithm is used to encrypt the message into cipher-text based on the chosen key. We will denote the message to be encrypted by  $m$ , the cipher-text resulting from the encryption by  $c$ , and the

key by  $k$ . These will each be such that  $m \in \mathcal{M}$ ,  $c \in \mathcal{C}$ , and  $k \in \mathcal{K}$ . Using this notation and our definition for bitwise **XOR** from above, we define

$$c := m \oplus k$$

to be the expression used by  $\text{Enc}$  as it encrypts the message.

Inversely from the  $\text{Enc}$  algorithm, the  $\text{Dec}$  algorithm is used to decrypt the cipher-text back into the message based on the supplied key. Similarly to  $\text{Enc}$ , we define

$$m := c \oplus k$$

to be the expression used by  $\text{Dec}$  as it decrypts the message.

To prove the security of this one-time pad, consider any arbitrary message  $m$  and any arbitrary cipher-text  $c$ , where  $m \in \mathcal{M}$  and  $c \in \mathcal{C}$ . Now, we express the probability of finding a particular  $c$ , given a particular  $m$  by

$$\Pr[C = c \mid M = m] \tag{2.2}$$

Using the fact that  $c = m \oplus k$ , this expression may be rewritten as

$$\begin{aligned} \Pr[C = c \mid M = m] &= \Pr[M \oplus K = c \mid M = m] \\ &= \Pr[M \oplus K = c \mid M = m] = \Pr[m \oplus K = c] \end{aligned}$$

Next, we **XOR** the random variable term in the right-hand term of this expression by  $m$  to obtain

$$\Pr [m \oplus (m \oplus K) = m \oplus c] = \Pr [K = m \oplus c],$$

because  $a \oplus a = 0$  for any binary string  $a$ . Above, we defined the probability of choosing any  $k$  to be  $\Pr [K = k] = 2^{-l}$ . This allows us to finally obtain the relations

$$\begin{aligned} \Pr [C = c \mid M = m] &= \Pr [M \oplus K = c \mid M = m] \\ &= \Pr [m \oplus K = c] \\ &= \Pr [m \oplus (m \oplus K) = m \oplus c] \\ &= \Pr [K = m \oplus c] = \frac{1}{2^l} \end{aligned} \tag{2.3}$$

Since our choice of  $m$  in expression 2.2 was arbitrary, the result in expression 2.3 must hold for any  $m \in \mathcal{M}$ . This implies that, for any  $m_0, m_1 \in \mathcal{M}$ , we relation

$$\Pr [K = m_0 \oplus c] = \frac{1}{2^l} = \Pr [K = m_1 \oplus c]$$

holds. This satisfies *Lemma 2.3* from the text, thus the one-time pad is perfectly secure.

**Problem 3):** One-time pads are difficult to use in practice because the any key must be the same length as the message and each key can be used only once.

**Problem 4):** Begin by defining  $\Pi$  to be the encryption scheme  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  where

- The security parameter  $n \in \mathbb{Z}$  and  $\text{Gen}$  are used to generate the key  $k$  by running  $\text{Gen}(1^n) = k$
- The key  $k$ , message  $m$ , and  $\text{Enc}$  are used to produce cipher-text  $c$  by running  $\text{Enc}_k(m) = c$
- The key  $k$ , cipher-text  $c$ , and  $\text{Dec}$  are used to recover the message  $m$  by running  $\text{Dec}_k(c) = m$

Additionally, let  $m_0, m_1$  be messages of the same length and  $c$  be the cipher-text generated from one of the messages by running  $\text{Enc}_k(m_b) = c$ , where  $b = \{0, 1\}$ . The encryption scheme  $\Pi$  is considered to be **CPA** secure if the probability of a polynomial time-limited adversary  $\mathcal{A}$ , with access to  $m_0, m_1$  and  $c$ , determining which message was used to compute  $c$  is equal to the sum of  $1/2$  and any value that is negligible on the order of  $n$ .

Now denote the experiment above as  $\text{Priv}_{\mathcal{A}, \Pi}^{\text{CPA}}(n)$ . Let this return 0 except when  $\mathcal{A}$  is able to determine which message was used to compute  $c$  then let  $\text{Priv}_{\mathcal{A}, \Pi}^{\text{CPA}}(n)$  return 1. Using this notation, our definition **CPA** security can be formally stated

$$\Pr \left[ \text{Priv}_{\mathcal{A}, \Pi}^{\text{CPA}}(n) = 1 \right] \leq \frac{1}{2} + \text{negl}(n) \quad (4.1)$$

where  $\text{negl}(n)$  is a negligible function of order  $n$ .

Finally, consider the case for the experiment  $\text{Priv}_{\mathcal{A}, \Pi}^{\text{CPA}}(n)$  where the messages  $m_0, m_1$  passed to the adversary  $\mathcal{A}$  are such that  $m_0 = m_1$  and the result of  $\text{Enc}_k(m_i) = c_i$  is fixed each  $m_i$  in the message space. That is to say, for any fixed  $k$ , that each  $c_i \in \mathcal{C}$  is determined by the result of  $\text{Enc}_k(m_i)$  for only one  $m_i \in \mathcal{M}$ . In this case the result of  $\text{Priv}_{\mathcal{A}, \Pi}^{\text{CPA}}(n)$  will always be 1 because the cipher-texts  $c_0 = \text{Enc}_k(m_0)$  and  $c_1 = \text{Enc}_k(m_1)$  are always equal thereby allowing  $\mathcal{A}$  *always* to succeed every time this is case. In this case,  $\Pi$  does not satisfy the definition given in expression 4.1 and is therefore not **CPA** secure. Thus, we must impose an additional requirement on **CPA** secure encryption schemes.

The problem arises from the case when  $m_0 = m_1$  and the when, for each fixed  $m_i \in \mathcal{M}$ , the function  $\text{Enc}_k(m_i)$  always returns the the same  $c_i$ . That is to say that the operation  $\text{Enc}_k(m_i)$  on each  $m_i \in \mathcal{M}$  always determines single, unique corresponding  $c_i \in \mathcal{C}$ . With this in mind, we refine our definition of **CPA** security to also include the requirement that, given a fixed key  $k$ , the Dec algorithm be non-deterministic on  $m \in \mathcal{M}$ . This is equivalent to requiring that the Dec algorithm be such that any passed  $m_i \in \mathcal{M}$  can return any  $c \in \mathcal{C}$  with some non-zero probability, thereby making Dec probabilistic instead of deterministic.

**Problem 5):** The **ECB** mode of operation is defined, in terms of the expression for the resulting cipher-text  $c$ , according to

$$c = \{F_k(m_1), F_k(m_2), \dots, F_k(m_i), \dots, F_k(m_l)\} \quad (5.1)$$

where  $F_k(m_i)$  is a pseudo random permutation function with key  $k$  and the  $m_i$  are the blocks of the message. Since each block is directly encrypted by  $F_k(m_i)$ , any  $m_i \in \mathcal{M}$  can result in only one unique  $c_i \in \mathcal{C}$  when passed to  $F_k(m_i)$  this mode is deterministic and therefore *not* **CPA** secure. Since this mode is not **CPA** secure, it *cannot* be CCA secure. This follows from the fact that CCA security of an encryption scheme  $\Pi$  implies the CPA security of  $\Pi$ .

We also define the **CTR** mode of operation in terms of the expression for resulting cipher-text  $c$ . For this mode of operation we have

$$c = \{c_0, c_1, c_2, \dots, c_i, \dots, c_l\} \quad (5.2)$$

with  $c_0 = \text{ctr}$  and the remaining  $c_i$  defined as  $c_i = r_i \oplus m_i$ . Here the  $m_i$  are the blocks of the message and the  $r_i$  are defined, in terms of their index  $i$ , some random initial counter value  $\text{ctr}$ , and the keyed pseudo random permutation function  $F_k(r_i)$ , according to

$$r_i = F_k(\text{ctr} + i) \quad (5.3)$$

Since  $r_i = F_k(\text{ctr} + i)$  and  $\text{ctr}$  is chosen at random, the set of all  $r_i$ ,  $r = \{r_1, r_2, \dots, r_i, \dots, r_l\}$  represents a pseudo random sequence with the same length as the message. This implies that result  $c_i = r_i \oplus m_i$  for each block of the message  $m_i$  depends on both on  $m_i$  and  $r_i$  instead of only  $m_i$  and the keyed pseudo random permutation function  $F(m_i)$ . By extension, any arbitrary message  $m \in \mathcal{M}$  can be encrypted into any cipher-text  $c \in \mathcal{C}$  with some non-zero probability, thereby making the **CTR** mode of operation probabilistic and thus **CPA** secure. Since the first block of the message,  $c_0$ , holds the value of  $\text{ctr}$  in the clear, the cipher-text resulting from this mode of operation is deterministic on the value of  $\text{ctr}$ .

This enables an adversary to employ a **CCA** attack by sending  $m_0 = 0^n$  and  $m_1 = 1^n$  to the encryption oracle, flipping the first bit of  $c_1$  in the cipher-text  $c$  returned by the encryption oracle to obtain  $c'$ , and then sending  $c'$  to the decryption oracle to obtain either  $10^{n-1}$  or  $01^{n-1}$ . The possible results from the decryption oracle respectively imply that either  $m_0$  was enciphered into  $c$  or  $m_1$  was enciphered into  $c$  thus giving the adversary two messages, cipher-text associated with each message, and the value of ctr used for encrypting both message. This information allows the adversary to eventually to recover the pseudo random permutation function  $F_k$  and its associated key used in to encrypt these messages. This attack is made possible because only the first bit in the cipher-text for  $m$  was changed and this first block of cipher-text is *directly* dependent on only the corresponding message block  $m_1$  and the key  $k$  when the the value of ctr is known. The **CCA** attack exploits the fact that encryption in **CTR** mode becomes deterministic the on the values of ctr and some cipher-text are known.

**Problem 6):** Since any **CPA** secure encryption scheme requires than any message  $m \in \mathcal{M}$  can be encrypted into any  $c \in \mathcal{C}$  with some non-zero probability, any cipher-text  $c$  can correspond to the encryption of several different messages, thereby preventing an eavesdropper from learning anything about plain-text by comparing two cipher-texts for equivalence.

**Problem 7):** Each block in a cipher-text from an encryption, under **CBC** mode of operation, is dependent on either the cipher-text from the block before it or the value of IV, they must be encrypted or decrypted serially. Therefore, under the **CBC** mode of operation, there is no speed increase, in either encryption and decryption, available from parallel processing.

**Problem 14):** Define the cipher-texts  $c$  and  $c'$ , encrypted on a one-time pad with the same key, from messages  $m$  and  $m'$ , respectively, as  $c = m \oplus k$  and  $c' = m' \oplus k$ . Now, **XOR** these expressions together to obtain

$$c \oplus c' = (m \oplus k) \oplus (m' \oplus k)$$



Since  $k \oplus k = 0$ , our result simplifies to

$$c \oplus c' = m \oplus m'$$

thereby giving an adversary a relation between cipher and message texts.