Modern Cryptography

2025 – 1Q

Wakaha Ogata

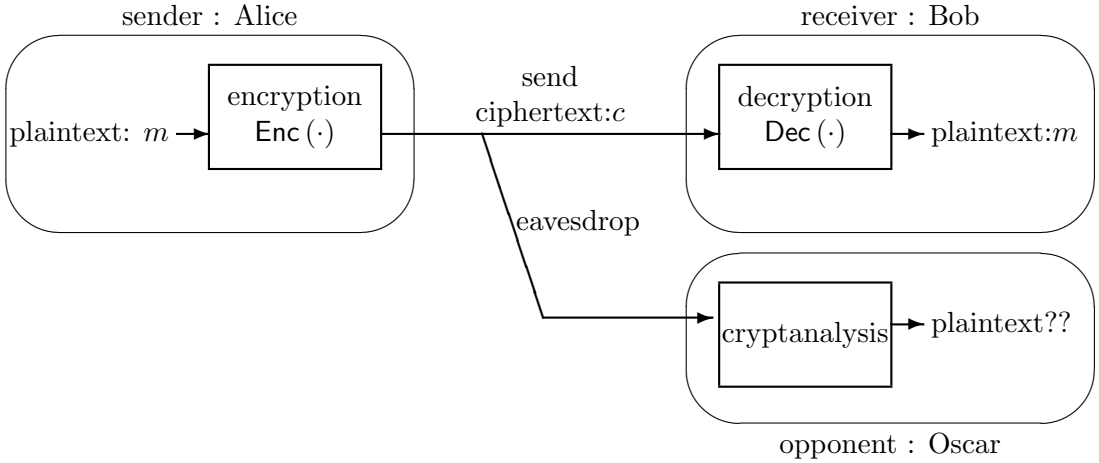
E-mail:ogata.w.aa@m.titech.ac.jp

Contents

Chapter 1

Private-key Cryptosystem

1.1 Basics of (symmetric) cryptosystem



Cryptosystem: $(\mathsf{Enc}, \mathsf{Dec})$ and ..
Keys: A "key" is used for encryption and decryption $(\mathsf{Enc}_k(m), \mathsf{Dec}_k(c))$.
How to share the key?

$\cdots$ We assume that they can share a key in some manner.

Example 1.1 (Caesar cipher/Shift cipher)
Shift k alphabet characters. (key=k)
for $k = 3$, $\mathsf{Enc}_3(A) = D$, $\mathsf{Enc}_3(B) = E$, $\mathsf{Enc}_3(X) = A$,
for $k = 5$, $\mathsf{Enc}_5(A) = F$, $\mathsf{Enc}_5(B) = G$

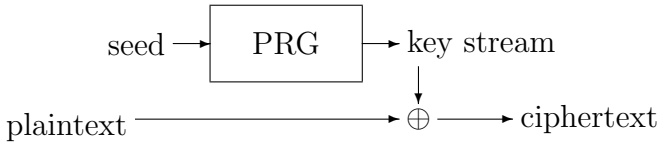
A B C D E F G $\cdots\cdots$ X Y Z A B $\cdots$

Example 1.2 (Permutation cipher)
Permute the order of characters. (key = permutation π)
for π=(3,4,1,5,2),
$\mathsf{Enc}_\pi(ABCDE) = CDAEB$
$\mathsf{Enc}_\pi(Alice) = icAel$

Example 1.3 (One-time pad)
Xor the key bit (string) and the plaintext bit (string).

$$\begin{array}{lcl} \textit{plaintext} & : & m_1m_2m_3\cdots \ (m_i = 0 \ \textit{or}\ 1) \\ \textit{key} & : & k_1k_2k_3\cdots \ (k_i = 0 \ \textit{or}\ 1) \\ \textit{ciphertext} : \mathsf{Enc}_{k_1k_2\cdots}(m_1m_2\cdots) & : & c_1c_2c_3\cdots \ (c_i = m_i \oplus k_i) \end{array}$$

The key stream is used and then discared. For practical use, a pseudo-random generator is used to generate a key stream from a short bit string. (stream cipher)



1.2 Security of cryptosystem / Adversary model

Secure = unbreakable against adversaries.
Adversary model =[who the adversary is] × [what he wants to do]

Who the adversary is / position: We always assume that Oscar knows the encryption algorithm. In addition, he can:

- wire-tap ... ciphertexts (Ciphertext only attack)
- obtain a plaintext published later
  ... some plaintexts and the corresponding ciphertexts
  (Known plaintext attack)
- access to the encryption machine (with an unknown key)
  ... plaintexts chosen by himself and the corresponding ciphertexts
  (Chosen plaintext attack)

- access to the decryption machine (with an unknown key)
  ... ciphertexts chosen by himself and the corresponding plaintexts
  (Chosen ciphertext attack)

What he wants to do / goal: Oscar wishes

- to find the key,
- to find the plaintext of a given ciphertext, (cryptanalysis)
- to distinguish ciphertexts (whether the plaintext is "yes" or "no").

(Example) adversary model A : to find the key under ciphertext only attack.

$c_i := \mathsf{Enc}_k(m_i)$

$c_1, c_2, c_3, \ldots \longrightarrow \boxed{M_A} \longrightarrow k$

(Example) adversary model B : to cryptanalyze under chosen plaintext attack.

$\hat{c} := \mathsf{Enc}_k(\hat{m})$

Enc_k

$\hat{c} \longrightarrow \boxed{M_B} \longrightarrow \hat{m}$

In both cases, we assume that k is chosen randomly from a set $\mathcal{K}$ called key space, m_i and $\hat{m}$ are chosen randomly from a (sufficiently large) set $\mathcal{M}$.

Definition 1.1 (Security of cryptosystem (informal)) *For a fixed adversary model (specified by Oscar's ability and goal), a cryptosystem is said to be secure, if there is no efficient algorithm that achieves the goal by using the ability.*

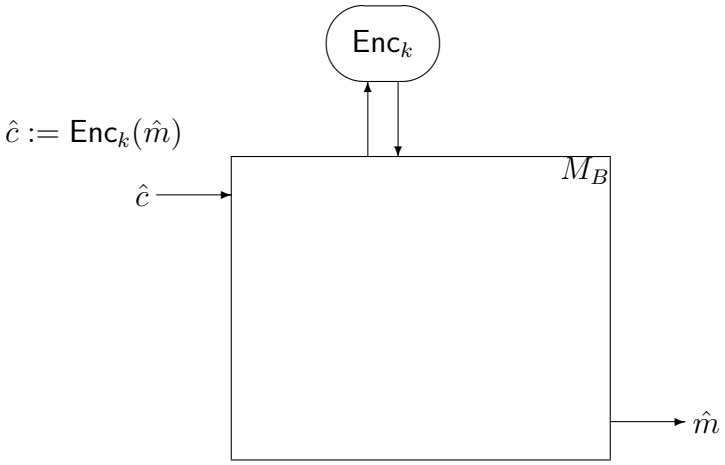
Which system is more secure?

· cryptosystem X that is secure in adversary model A.

· cryptosystem Y that is secure in adversary model B.

Theorem 1.1 (Relation between security notions (informal)) *If a cryptosystem is secure in adversary model B, then it is also secure in adversary model A.*

(Proof) It is sufficient to construct an efficient algorithm M_B that achieves the goal in adversary model B by using an efficient attack algorithm M_A that achieves the goal in adversary model A.



Concrete examples of attack:

- Classical attack (Oscar uses the knowledge that the plaintext is written in English)
  $\cdots$ cryptanalysis under ciphertext only attack
- Exhaustive search (Oscar searches over all the possible keys)
  $\cdots$ find the key under known plaintext attack
- Differential attack (Oscar prepares many pairs of plaintexts that have constant differential and collects their corresponding ciphertexts, and then guesses the key)
  $\cdots$ find the key under chosen plaintext attack

1.3 Another classification of security

- Unconditionally secure : one cannot break the cryptosystem even if he has infinite computational power.
- Computationally secure : one can break the cryptosystem if he has infinite computational power, but the attack is *infeasible*. (e.g., it takes 10,000 years even if he uses TSUBAME).

Security	Advantage	Disadvantage	Example
unconditional	absolute security	$\|\text{key}\| \geq \|\text{plaintext}\|$	One-time-pad
computational	a key is reusable	conditional security	AES, RSA

Cryptosystems in which a fixed length single key is used to encrypt many plaintexts cannot be unconditionally secure against known plaintext attack (but can be computationally secure).

When we can say "infeasible"?
Computational complexity $T(n)$: the number of steps needed to output a computation result when the input is n-bit. Usually it is expressed by using Big O notation $O(\cdot)$.

- addition : $T_{\text{add}}(n) = O(n)$
- multiplication : $T_{\text{mult}}(n) = O(n^2)$
- AKS primality test : $T_{\text{AKS}}(n) \approx O(n^6)$
- exhaustive key search attack (ciphertext length $\approx$ key length $= n$) : $T_{\text{ex-attack}}(n) = O(2^n)$

Number of steps and computation time (10^{12} steps = 1 sec.)

n	$T(n) = n^2$	$T(n) = n^6$	$T(n) = 2^n$
1	1 $(1.0 \times 10^{-12}\text{sec})$	1 $(1.0 \times 10^{-12}\text{sec})$	2 $(2.0 \times 10^{-12}\text{sec})$
10	100 $(1.0 \times 10^{-10}\text{sec})$	10^6 $(1.0 \times 10^{-6}\text{sec})$	1024 $(1.0 \times 10^{-9}\text{sec})$
50	2500 $(2.5 \times 10^{-9}\text{sec})$	$(1.56 \times 10^{-2}\text{sec})$	$(1.1 \times 10^{3}\text{sec} \approx 19$ min)
100	10000 $(1.0 \times 10^{-8}\text{sec})$	(1.0 sec)	$(1.3 \times 10^{18}\text{sec} \approx 4.1 \times 10^{10}$ years)

Classification of problems. For a given problem,

- if there exists an algorithm which solves the problem with polynomial steps, i.e., $T(n) = O(n^d)$ for some d, then the problem is called **easy**.
- if there is no algorithm which solves the problem with polynomial steps, e.g., $T(n)$ is an exponential function, $O(2^n)$, for any algorithms, then the problem is called **difficult**.

For a computationally secure cryptosystem,

- encryption/decryption must be easy for legitimated users.
- cryptanalysis must be difficult for any attackers.

Consider a cryptosystem such that the computational complexities of encryption/decryption are $T(n) = n^2$, and that of the best cryptanalysis is $T(n) = 2^n$, where n is the key length. If we extend 128 bit key to 256 bit key,

- encryption/decryption time becomes $\frac{256^2}{128^2} = 4$ times longer,
- cryptanalysis time becomes $\frac{2^{256}}{2^{128}} = 2^{128}$ times longer,

$\rightarrow$ for the improvement of the computational power, we just need to increase the key length.

1.4 Three main subjects

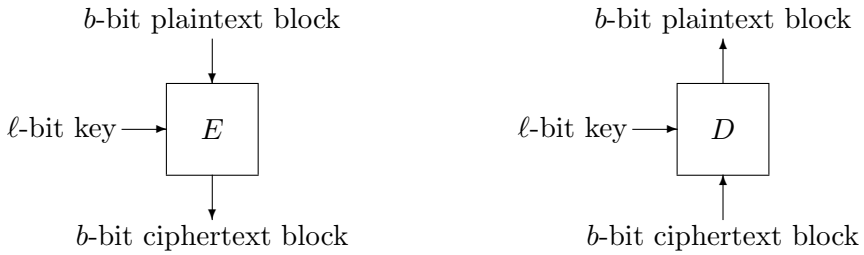
There are three subjects in the research of symmetric cryptosystem :

- **stream cipher** $\approx$ designing a pseudo-random generator.
- **block cipher**. A tool for encrypting fixed length plaintexts.
- **mode of operation**. Metholds to use block ciphers to encrypt arbitraly length plaintexts (and to authenticate messages).

1.5 Block cipher and mode of operation

1.5.1 Block cipher

Fixed length plaintext $\rightarrow$ the same length ciphertext



Block cipher : (E, D) with key length ℓ and block length b.
Correctness : for any key $k \in \{0,1\}^\ell$ and plaintext block $m \in \{0,1\}^b$, $D_k(E_k(m)) = m$ holds.

Definition 1.2 (Security of block cipher: Pseudo-random (informal)) *We say a block cipher is secure, if any efficient attacker cannot tell the difference between the block cipher and a random permutation.*

E_k returns $c_i := E_k(m_i)$ $\qquad$ R returns random c_i

1.5.2 Mode of operation

We assume that there exists a "secure" block cipher (E_k, D_k).
How to encrypt long messages with the block cipher ?

For simplicity, we assume that the length of plaintext $\mathbf{m}$ is multiple of block length b.

$$\mathbf{m} = m_1 \| m_2 \| \cdots \| m_n, \quad m_i \in \{0,1\}^b$$

ECB (Electric Code Book) mode.

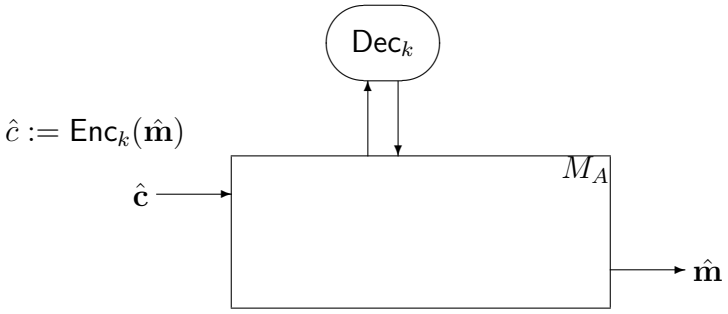
$$\begin{aligned}\mathsf{Enc}_k(\mathbf{m}) &= c_1\|c_2\|\cdots\|c_n\\ c_i &= E_k(m_i)\end{aligned}$$

Theorem 1.2 *ECB mode is insecure in the sense of distinguishing ciphertext under chosen plaintext attack.*

(Proof) Let E_k be an arbitrary block cipher, and m_1, m_2 be arbitrary plaintext-blocks different from each other. Clearly, $\mathsf{Enc}_k(m_1\|m_1) = E_k(m_1)\|E_k(m_1)$ and $\mathsf{Enc}_k(m_1\|m_2) = E_k(m_1)\|E_k(m_2)$ can be easily distinguished.

Theorem 1.3 *ECB mode is insecure in the sence of cryptoanalysis under chosen ciphertext attack.*

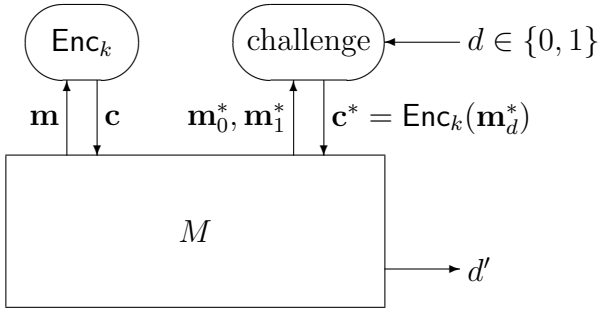
(Proof) It is sufficient to consrtruct attack algorithm M_A.



CBC (Cipher Block Chaining) mode.

$$\begin{aligned}\mathsf{Enc}_k(\mathbf{m}) &= c_0\|c_1\|c_2\|\cdots\|c_n\\ c_0 &= IV \leftarrow\!\!\$\ \{0,1\}^b\\ c_i &= E_k(m_i \oplus c_{i-1})\end{aligned}$$

Theorem 1.4 *If random permutation is used as the underlying block cipher and block length b is long enough (i.e.* $1/2^b$ *is negligibly small), then CBC mode is secure in the sense of indistinguishable against chosen plaintext attak.*



(Sketch of proof) For simplicity, we assume that M asks Encryption oracle only once. Let

$$\begin{aligned} \mathbf{m} &= m_1 \| \cdots \| m_n, & \mathbf{m}_d^* &= m_1^* \| \cdots \| m_{n^*}^*, \\ \mathbf{c} &= IV \| c_1 \| \cdots \| c_n, & \mathbf{c}_d^* &= IV^* \| c_1^* \| \cdots \| c_{n^*}^*, \\ in_i &:= m_i \oplus c_{i-1}, & in_i^* &:= m_i^* \oplus c_{i-1}^*. \end{aligned}$$

First, assume that $(in_1, \ldots, in_n, in_1^*, \ldots, in_{n^*}^*)$ are all different each other. In this case, $IV, c_1, \ldots, c_n, IV^*, c_1^*, \ldots, c_{n^*}^*$ are randomly and independently distributed regardless of bit d, because E is assumed to be random permutation. This means that M cannot guess d at all.

Next, we estimate the probability that this assumption holds. By using the facts that IV and IV^* are chosen randomly and independently, and E is assumed to be random permutation, we can show

$$\Pr[(in_1, \ldots, in_n, in_1^*, \ldots, in_{n^*}^*) \text{ are all different}] = 1 - \frac{(n+n^*)(n+n^*-1)}{2^{b+1}}.$$

Since $1/2^b$ is negligibly small, this probaility is almost 1.
Consequently, M cannot guess d.

Counter mode. (Stream cipher)

$$\mathsf{Enc}_k(\mathbf{m}) = ctr \| c_1 \| c_2 \| \cdots \| c_n$$
$$ctr : \text{counter}$$
$$c_i = E_k(ctr + i) \oplus m_i$$

Note that E_k is not necessary to be a permutation, because $D_k = E_k^{-1}$ is not used.

Theorem 1.5 *If random function is used as the underlying block cipher, then CTR mode is secure in the sense of indistinguishable against chosen plaintext attack.*

The proof is similar to that of Theorem 1.4, but

$$\Pr[(in_1, \ldots, in_n, in_1^*, \ldots, in_{n^*}^*) \text{ are all different}] = 1$$

in this case.

1.5.3 AES

(Advanced Encryption Standard)
An standard selected by the U.S. National Institute of Standards and Technology (NIST) in 2001.

- The block size is 128-bit. The key size can be 128, 192, or 256.
- SPN (Substitution Premutation Network) structure.
- Number of rounds (number of S layer) depends on the key size.

Key size	Number of rounds
128	10
192	12
256	14