

Lecture 3: Cryptography Overview

Yuan Xue

I. SECURITY THREAT IN NETWORKS

As we have seen in the last two lectures, a network, especially one that grows to global scale, has many vulnerabilities. Threats are raised to against the key aspects of security: confidentiality, integrity, and availability.

- *Attacks against confidentiality*
 - eavesdropping
 - traffic flow analysis
- *Attacks against integrity*
 - IP spoof
 - Sequence number attack
 - Man-in-the-middle attack
- *Attacks against availability*
 - Denial of Service attack
 - Traffic redirection
- *Precursor to attack*
 - Port scan

II. SECURITY SERVICES AND MECHANISMS

To protect the network from various security threats, we study the security mechanisms and security services in this section. First, let us examine some related terms.

A. Terms

- *Vulnerability*: an aspect of the system that permits attackers to mount a successful attack, sometimes also called a “security hole”.
- *Weakness* a potential vulnerability, whose risk is not clear. Sometimes several weaknesses might combine to yield a full-fledged vulnerability.
- *Threat*: a circumstance or scenario with the potential to exploit a vulnerability, and cause harm to a system.
- *Attack*: A deliberate attempt to breach system security. Note that not all attacks are successful. An attack usually refers to a specific stratagem. A threat refers to a broader class of ways that things could go wrong. Attacks are usually classified into two types: (1) *Passive attack* refers to attack that does not result in a change to the system, and attempts to break the system solely based upon observed data. (2) *Active attack*, on the other hand, involves modifying, replaying, inserting, deleting, or blocking data.
- *Security Mechanism*: a mechanism that is designed to detect, prevent, or recover from a security attack.

- *Security Service* makes use of security mechanisms to counter security attacks.

ITU – T² Recommendation X.800, *Security Architecture for OSI* defines a systematic approach for security services and security mechanisms. In particular, X.800 divides the security services into five categories.

- *Authentication*: the assurance that the communicating entity is the one that it claims to be.
- *Access Control*: the prevention of unauthorized use of a resource.
- *Data confidentiality*: the protection of data from unauthorized disclosure.
- *Data integrity*: the assurance that data received are the same as send by an authorized entity.
- *Nonrepudiation*: provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.

Security mechanisms are used to implement the above security services. Each specific security mechanism and its corresponding security service is given in Table. I.

Security Service	Supporting Security Mechanisms
Peer entity authentication	encipherment, digital signature, authentication exchange
Data origin authentication	encipherment, digital signature
Access control	access control
Confidentiality	encipherment, routing control
Traffic flow confidentiality	encipherment, traffic padding, routing control
Data integrity	encipherment, digital signature, data integrity
Nonrepudiation	digital signature, data integrity, notarization
Availability	access control, authentication exchange

TABLE I
RELATIONSHIP BETWEEN SECURITY SERVICES AND MECHANISMS

III. CRYPTOGRAPHY OVERVIEW

Now we will study *cryptography*, which provides a strong tool against many kinds of security threats. In particular, we will examine the following issues in cryptography.

- *what encryption does*. We will introduce the basic concepts in encryption and study the encryption model.
- *how encryption works*. We will examine the basic principles to design encryption algorithms. Specifically, we will study two building blocks of encryption algorithms, and show how they can be expanded and improved to create stronger and practical encryption algorithms.
- *how encryption can fail*. We would also discuss cryptanalysis, and show how encryption algorithms can be broken.
- *how encryption can be applied*. We will further study how encryption algorithms can be used as building blocks with protocols and structures to perform security services.

IV. MODEL FOR ENCRYPTION

Before getting into the details of each encryption algorithm, we introduce the model and the basic concepts in encryption. We start with a simple example. Suppose you want to send a message to one of

your friends Bob. you wish the message not to be understood by others. One way to do this is to devise some “secret codes”, which substitute a letter for each letter in your original message. Such “secret codes” must be agreed by Bob so that he can understand your “secret message” once he receives it.

Caesar Cipher provides one feasible solution to such “secret codes”. Julius Caesar is said to be the first to use this scheme. In Caesar cipher, each letter is translated to a letter that is 3 places after it in the alphabet¹. So a full translation chart of the Caesar cipher is shown as follows².

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

TABLE II
CAESAR CIPHER

Using this encryption, the message “meet me after the party” would be translated as

Original	m	e	e	t	m	e	a	f	t	e	r	t	h	e	p	a	r	t	y
Translated	P	H	H	W	P	H	D	I	W	H	U	W	K	H	S	D	U	W	B

TABLE III
MESSAGE ENCRYPTED BY CAESAR CIPHER

The simple Caesar cipher is easy to memorize and implement. However, once the patten of “shifting 3 places” is recognized by an interceptor, he is able to read all the subsequent messages between you and Bob. The general Caesar Cipher addresses this problem and provides a more secure encryption scheme. In this scheme, the translated letter can be the one that is shifted by K places in the alphabet. For $K = 1$, the message would be translated as follows. To communicate with Bob using general Caesar cipher, you need to agree with him on the value of K .

Original	m	e	e	t	m	e	a	f	t	e	r	t	h	e	p	a	r	t	y
Translated	N	F	F	U	N	F	B	G	U	F	S	U	I	F	Q	B	S	U	Z

TABLE IV
MESSAGE ENCRYPTED BY GENERAL CAESAR CIPHER (K=1)

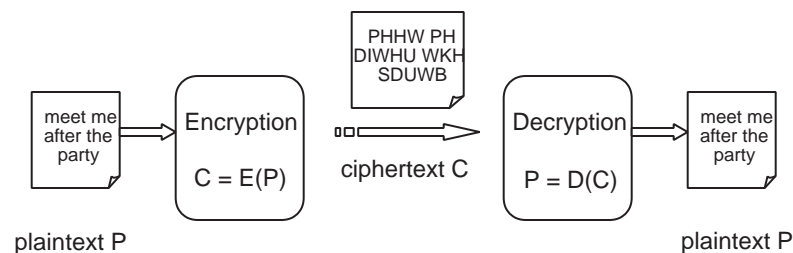


Fig. 1. Encryption Model.

¹Note that the alphabet is wrapped around.

²The following conventions are used for letters in our class: the original text (plaintext) is in lower-case; the translated text (ciphertext) is in uppercase; key values are in italicized lowercase.

From the above examples, we observe the following components in an encryption system as shown in Fig. 1.

- *Plaintext*: the original message. Formally, we use $P = \langle p_1, p_2, \dots, p_n \rangle$ to denote the plaintext. In the above examples, $P = \langle m, e, e, t, m, e, a, f, t, e, r, t, h, e, p, a, r, t, y \rangle$.
- *Ciphertext*: the translated or encrypted message, denoted as $C = \langle c_1, c_2, \dots, c_m \rangle$. In the first example, the ciphertext $C = \langle P, H, H, W, P, H, D, I, W, H, U, W, K, H, S, D, U, W, B \rangle$.
- *Encryption (enciphering)*: the process of encoding a message so that its meaning is not obvious; the transformation from plaintext to ciphertext. Formally, encryption is denoted using $C = E(P)$, where C is the ciphertext and P is the plaintext. In the Caesar cipher, $C = E(P) = (P + 3) \bmod (26)^3$; in the general Caesar cipher, $C = E(P) = (P + K) \bmod (26)$.
- *Decryption (deciphering)*: the reverse process of encryption; the transformation from ciphertext to plaintext, formally denoted as $P = D(C)$. In the Caesar cipher, $P = D(C) = (C - 3) \bmod (26)$; in the general Caesar cipher, $P = D(C) = (C - K) \bmod (26)$.
- *Cryptosystem*: a system for encryption and decryption. What we seek is a cryptosystem for which $P = D(E(P))$.
- *Key*: an input to the encryption and decryption algorithm. The encryption algorithm will produce a different ciphertext depending on the specific key being used. The corresponding key is needed to decrypt the ciphertext to plaintext. A key gives us flexibility in using an encryption algorithm, and provides additional security: if the encryption algorithm falls into the interceptor's hand, future messages can still be kept secret because the interceptor does not know the key value.

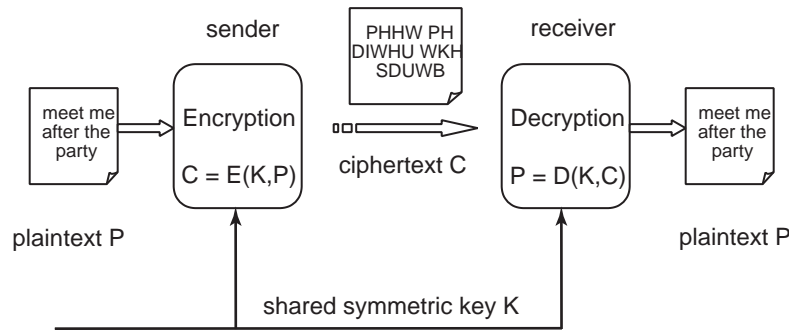


Fig. 2. Symmetric Cryptosystem Model.

When the keys used for encryption and decryption are the same as shown in Fig. 2, they are called *symmetric key*, or *secret key*, and are denoted as K . The encryption (symmetric-key encryption or secret-key encryption) process can be denoted as $C = E(K, P)$; and the decryption process is denoted as $P = D(K, C)$. The cryptosystem is called *symmetric cryptosystem* or *conventional cryptosystem*, and needs to satisfy $P = D(K, E(K, P))$.

The keys for encryption and decryption can also be different as shown in Fig. 3. In this case, the encryption key is denoted as K_E , and the decryption key is denoted as K_D . The encryption (asymmetric-key encryption, public-key encryption) process is formally denoted as $C = E(K_E, P)$, and the decryption process is denoted as $P = D(K_D, C)$. Such a cryptosystem is called *asymmetric cryptosystem*. Similar to symmetric cryptosystem, the following requirement needs to be satisfied: $P = D(K_D, E(K_E, P))$.

³Here we assign a numerical equivalent to each letter (e.g., a = 0, ..., z = 25).

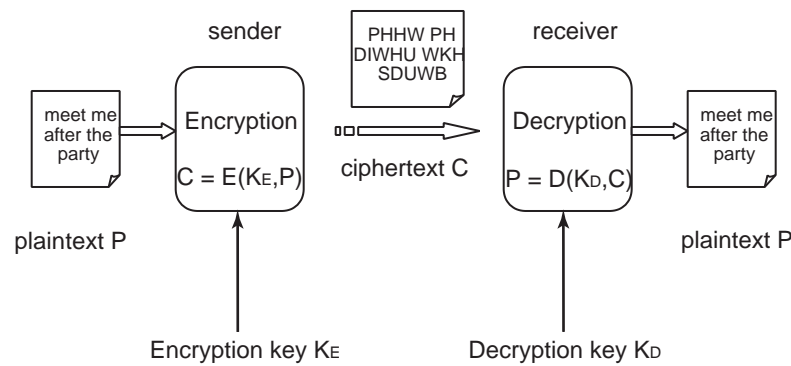


Fig. 3. Asymmetric Cryptosystem Model.

Cryptography studies the encryption and decryption schemes. *Cryptanalysis* studies how to “break the code”, i.e., how to decrypt the ciphertext without the knowledge of the encryption details. The areas of *cryptography* and *cryptanalysis* together are called *cryptology*.

We will first study *symmetric cryptosystem* in this lecture.

Before we get into the details of symmetric encryption schemes, let’s first exam how these schemes may fail. There are two general approaches to attacking a symmetric encryption scheme.

- *Brute-force attack* tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained.
- *Cryptanalysis* exploits the characteristics of the algorithm and the traces of structure or pattern in the plaintext that survive encryption, to attempt to break a single message, or to deduce the key in order to break the subsequent messages. A cryptanalyst can work with a variety of pieces of information and tools, such as statistical tools, and properties of languages. Based on the information known to the cryptanalyst, the cryptanalytic attacks are classified as follows.
 - In *ciphertext only attack*, encryption algorithm and ciphertext are known to the cryptanalyst.
 - In *known plaintext attack*, information known includes: encryption algorithm, ciphertext, and one or more plaintext-ciphertext pairs formed with the secret key.
 - In *chosen plaintext attack*, information known includes: encryption algorithm, ciphertext, and chosen plaintext and its corresponding ciphertext generated with the secret key.
 - In *chosen ciphertext attack*, information known includes: encryption algorithm, ciphertext, and chosen ciphertext and its corresponding decrypted plaintext with the secret key.
 - In *chosen text attack*, information known in both chosen plaintext attack and chosen ciphertext attack is available to the cryptanalyst.

Generally, an encryption algorithm is designed to withstand a *known plaintext attack*. An encryption algorithm is called “*unconditionally secure*”, if the ciphertext generated by the algorithm does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available, and how much time an opponents has. An encryption algorithm is “*computationally secure*”, if (1) the cost of breaking the cipher exceeds the value of the encrypted information; (2) the time required to break the cipher exceeds the useful lifetime of the information. There are two important issues when considering breaking an encryption algorithm. First, the cryptanalyst can not be expected to try only the hard, long way (e.g. timing attack). Second, estimations of “computational security” are based on

current technology. With the advance of computer technology, many encryption schemes which were once regarded “secure” are breakable now.