A Review of Major Security Primitives

Cunsheng DING, HKUST

COMP5631

The objective of this review lecture

The objective of this lecture is to answer the following questions:

1. What are the security services covered in this course so far?
2. How to provide the data confidentiality service?
3. How to provide the data origin authentication and data integrity services?
4. How to provide the mutual authentication service?
5. How to provide the anti-replay service?
6. How to establish a common secret key?

The motivation of reviewing the security primitives

The first part (i.e., cryptography) of this course covers security primitives for providing specific security services.

Most of them are used in real-world security systems such as PGP and S/MIME, IP Security, SSL/TLS, VPNs and the Secure Shell, which will be covered in the second part of this course.

To better understand these real-world security systems, we need recall some basic security primitives before studying real-world security systems.

Passive and active attacks

Question: What are passive and active attacks?

Question: What is the replay attack?

Question: Is the replay attack a passive attack?

Question: Could the replay attack be a serious security problem?

Question: How is the anti-replay service provided?

A summary of the security services covered so far

- Data confidentiality
- Sender authentication, receiver authentication, mutual authentication
- Data integrity (data authentication)
- Signer nonrepudiation
- Anti-replay, data origin authentication
- Key generation, key distribution, key establishment

Comment: You have to fully understand these security services and know how to provide these security services.

The data confidentiality service

A cipher is used to encrypt a piece of data m. The ciphertext $E_k(m)$ may be in storage or in transmission:

$$\text{Alice} \ \rightarrow E_k(m) \rightarrow \ \text{Bob}$$

Encryption is usually done in the CBC mode.

Question: Does this protocol provide other security services? If yes, what are these security services?

Answer: It depends on if m contains a timestamp, ID of sender or if m has redundancy.

Data origin authentication & data integrity (1)

Question: How to provide the two security services **simultaneously**?

Answer: In real-world security systems, two protocols are used.

Data origin authentication & data integrity (2)

In PGP and S/MIME, the two security services are provided in the form:

$$\text{Alice } \rightarrow m||\text{Alice' digital signature on } m \rightarrow \text{Bob}.$$

Question: What should Bob do after receiving the text from Alice?

Question: Does this protocol provide other security services?

Data origin authentication & data integrity (3)

In most real-world security systems, the two security services are provided simultaneously as follows:

$$\text{Alice} \rightarrow m||h_k(m) \rightarrow \text{Bob},$$

where a hash function h and an authentication key are used in the HMAC mode for obtaining a keyed hash function h_k. The HMAC approach was covered earlier.

The value $h_k(m)$ is called the **message authentication code** (MAC).

Question: What should Bob do after receiving the text from Alice?

Question: Does this protocol provide other security services?

Providing the mutual authentication (1)

Question: How to provide the mutual authentication service?

Answer: In real-world security systems, two protocols are used. One is a Type-1 authentication protocol, the other is a Type-2 authentication protocol.

Providing the mutual authentication (2)

Type-1 authentication protocol (a Kerberos-like protocol or Niederheim-Schroeder-like protocol),

$$\text{Alice} \rightarrow E_k(ID_A||ID_B||timestamp) \rightarrow \text{Bob}$$

$$\text{Alice} \leftarrow E_k(ID_B||ID_A||timestamp) \leftarrow \text{Bob}$$

where a pre-shared secret key k is used.

Question: Does this protocol provide other security services?

Question: What are the two purposes of adding the timestamp in this protocol? (hint: anty-? and sender ??)

Providing the mutual authentication (3)

Type-2 authentication protocol (a challenge-response protocol),

$$\begin{aligned} \text{Alice} &\rightarrow E_{k_e^B}(N_1 || ID_A) \rightarrow \text{Bob} \\ \text{Alice} &\leftarrow E_{k_e^A}(ID_B || N_1 || N_2) \leftarrow \text{Bob} \\ \text{Alice} &\rightarrow E_{k_e^B}(N_2 || ID_A) \rightarrow \text{Bob} \end{aligned}$$

This is allows Alice and Bob to authenticate each other.

Question: Does this protocol provide other security services?

Providing the signer nonrepudiation service

Protocols: In real-world security systems, the following two digital signature schemes are supported:

- The RSA public-key cipher and a hash function.
- The Digital Signature Standard (also called DSA).

Establishing a common secret key (1)

Recall: We talked about key generation, key distribution, key exchange.

Remark: Key management includes, key generation, key distribution, key exchange, key storage, key destruction, etc.

Comment: "Key management" is the most complicated building block in real-world security systems!

Information: Two protocols for establishing a common secret key are supported in real-world security systems.

Establishing a common secret key (2)

The first one is the digital-envelop protocol,

$$\text{Alice} \rightarrow E_{k_e^B}(k) \rightarrow \text{Bob},$$

where we assume that all public keys are in the public domain.

Question: Is this protocol secure with respect to man-in-the-middle attacks.

Answer: No. Eva can replace $E_{k_e^B}(k)$ with $E_{k_e^B}(k')$ in the middle of transmission and pretends that she is Alice. In this way, Eva will be able to establish a secret key k' with Bob and pretends that she is Alice.

Question: Why is it used in real-world security systems?

Establishing a common secret key (3)

The second one is the Diffie-Hellman key exchange protocol (see Lecture 8).

Question: Is this protocol secure with respect to man-in-the-middle attacks.

Question: Why is it used in real-world security systems?

Diffie-Hellman group: a specific pair (p, α) used in the Diffie-Hellman protocol.