Cryptography and Security

Cunsheng DING

HKUST, Hong Kong

Version 3

Lecture 13: Protocols for Security Services

Main Topics of This Lecture

1. Passive and active attacks.
2. Authentication protocols and their classification.
3. A protocol for authentication and nonrepudiation.
4. A protocol for authentication, confidentiality and nonrepudiation.
5. Merkel's protocol and a man-in-the-middle attack.
6. The Needham-Schröder protocol.

Passive and Active Attacks

Passive and active attacks

Passive attacks: Any attack on a security system under the assumption that the attacker can only intercept messages exchanged over a communication channel is called a **passive attack**.

Active attacks: Any attack on a security system under the assumption that the attacker can stop, intercept, delete, modify, and replay messages exchanged over a communication channel or insert his/her messages into the channel is called a **active attack**. In such a scenario, we say that the attacker has **full control** over the communication channel.

Part II: Authentication Protocols and their Classification

Authentication Aspects

- Verify that a received message is not a forged or modified one (i.e., data authentication, data integrity).
- Verify that an alleged sender is the real one (sender authentication).
- Verify that the alleged creator of a message is the real one (data origin authentication).
- Verify that a received message is a current one (i.e., not a replayed earlier one).

A Basic Model of Authentication

A wants to send messages to B. They share a secret function f. A sends B:

$$m||f(m).$$

When B receives a text c, he "partitions" c into $c = c_1||c_2$, where c_2 has the same length as $f(m)$, and then checks whether $f(c_1) = c_2$. If yes, he concludes that c is indeed the message created by A and was not modified during transmission.

Such protocol provides **data origin authentication** and **data integrity** to certain degree if f is designed **"properly"** and also **sender authentication** if m contains a timestamp.

$f(m)$ is called the **authenticator**, and f the **authentication function**.

Remark: It uses a preshared secret, where the two parties trust each other.

A Basic Model of Authentication

A wants to send messages to B. They share a secret function f. A sends B:

$$m||f(m).$$

Conclusion: Such a protocol can provide several security services.

Natural Law: If you want to gain, you have to pay.

Question: What is the price paid in this authentication system?

Authentication Functions

Question: How to design the authentication function f in the basic model?

Design consideration: The receiver should be able to partition the received message for authentication checking.

Approach 1: The length of the authenticator $f(m)$ is proportional to that of m.

For example, f is the encryption function of a one-key cipher.

Approach 2: The length of the authenticator $f(m)$ is the same for all m.

For example, f is a keyed hash function h_k.

Authentication Protocol 1

The protocol: Suppose that Alice and Bob share a secret key k for a one-key cipher and no third party possesses k. Assume that the cipher text $E_k(m)$ has always the same length as that of the message m.

$$\text{Alice} \quad \longrightarrow \quad m||E_k(m) \quad \longrightarrow \quad \text{Bob}$$

Remark: E_k is the authentication function f in the basic authentication model. The length of $E_k[m]$ is the same as that of m if E_k is the encryption function of AES and m is padded properly.

Authentication and integrity level: Depends on the security of the one-key cipher.

Advantages and disadvantages: High-level security, but very expensive.

Authentication Protocol 2

Protocol: Let h be a hash function. Assume that Alice and Bob share a secret key k of a one-key cipher. No third party possesses k.

$$\text{Alice} \quad \longrightarrow \quad m||E_k[h(m)] \quad \longrightarrow \quad \text{Bob}$$

Remark: $E_k \circ h$ is the authentication function f in the basic authentication model, and is the second keyed hash function in Lecture 11.

Design requirements: It provides a certain degree of authentication of both data origin and message if the following hold (see Lecture 11):

- The one-key cipher is computationally secure.
- The hash function has the weak collision resistance property.

A Classification of Authentication Protocols

Type 1: Those based on a preshared secret. For example, Authentication Protocol 1 and Authentication Protocol 2 in this lecture.

Type 2: Those do not need a preshared secret. For example, the following is for mutual authentication:

1. A sends $E_{k_e^{(B)}}[N_1||ID_A]$ to B, where N_1 is a nonce used to identify this transaction uniquely, and is generated by A.

2. B generates a new nonce N_2, and sends $E_{k_e^{(A)}}[N_1||N_2||ID_B]$ to A. After decryption A gets N_1, and is sure that the responder is B.

3. A sends $E_{k_e^{(B)}}[N_2||ID_A]$ to B.

Part III: A Protocol for Authentication and Nonrepudiation

Remark: This protocol is used in PGP and S/MIME.

Authentication with Nonrepudiation

Protocol: Let h be a hash function. Assume that Alice and Bob have exchanged their public keys.

$$\text{Alice} \quad \longrightarrow \quad m||D_{k_d^{(A)}}[h(m)] \quad \longrightarrow \quad \text{Bob}$$

Conclusion: It provides a certain degree of signer nonrepudiation, data origin authentication, data integrity, but no data confidentiality. It also provides sender authentication if m contains a timestamp.

Remark: Signer nonrepudation implies bother data origin authentication and data integrity.

Security requirements: See Lecture 11 .

- The public-key cipher should be computationally secure.
- h should have the weak collision resistance and one-way property.

Part IV: A Protocol for Authentication, Confidentiality and Nonrepudiation

Remark: This protocol is used in PGP and S/MIME.

Authentication + Nonrepudiation + Confidentiality

Protocol: Let h be a hash function. Assume that Alice and Bob share a secret key k of a one-key cipher, and have exchanged their public keys.

$$\text{Alice} \quad \longrightarrow \quad E_k\left(m||D_{k_d^{(A)}}[h(m)]\right) \quad \longrightarrow \quad \text{Bob}$$

Exercise: Give details of the verification process by Bob.

Conclusion: It provides a certain degree of signer nonrepudiation, data origin authentication, data integrity, data confidentiality. It also provides sender authentication if m contains a timestamp. $\boxed{\text{Why?}}$

Question: Can we relieve the design requirements for h?

Part V: Key Distribution Protocols and a Man-in-the-middle Attack

Two Key Distribution Protocols with a PKC

Comments: Public key ciphers are usually not used for encrypting data of large size due to their poor performance. They are used for distributing secret keys of one-key ciphers and/or for signing messages (see Lecture 7).

The digital envelop protocol: It was introduced in Lecture 7, where we assumed that Alice and Bob exchanged their public keys beforehand.

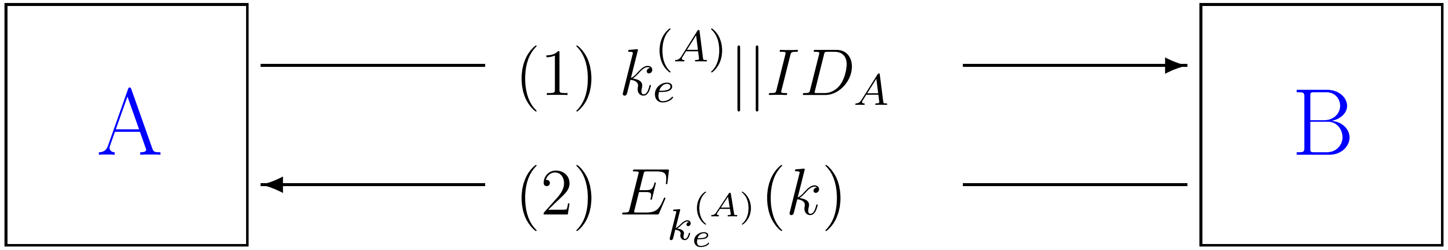
A variant of the digital envelop protocol: Merkel's protocol, where we assume that Alice and Bob do not know each other's public key.

Merkel's Key Distribution Protocol

Scenario: A and B want to establish a session key.

1. A generates a key pair $\left(k_e^{(A)}, k_d^{(A)}\right)$, and sends $k_e^{(A)} || ID_A$ to B, where ID_A is an identifier of A.
2. B generates a secret key k, and sends $E_{k_e^{(A)}}(k)$ to A.
3. A computes $D_{k_d^{(A)}}\left[E_{k_e^{(A)}}(k)\right] = k$.
4. A discards $\left(k_e^{(A)}, k_d^{(A)}\right)$, and B discards $k_e^{(A)}$.

Merkel Key Distribution Protocol: Pictorial



Remark: This is a variant of the **digital envelop protocol**, here we assume that A and B did not exchange their public keys before.

Comment: This protocol is **secure** with respect to **passive attacks**, provided that the public-key cipher is secure.

Comment: This protocol is **vulnerable** to an **active attack**. If an enemy E has control of the **intervening** communication channel, then E can **"compromise"** the communication without being detected.

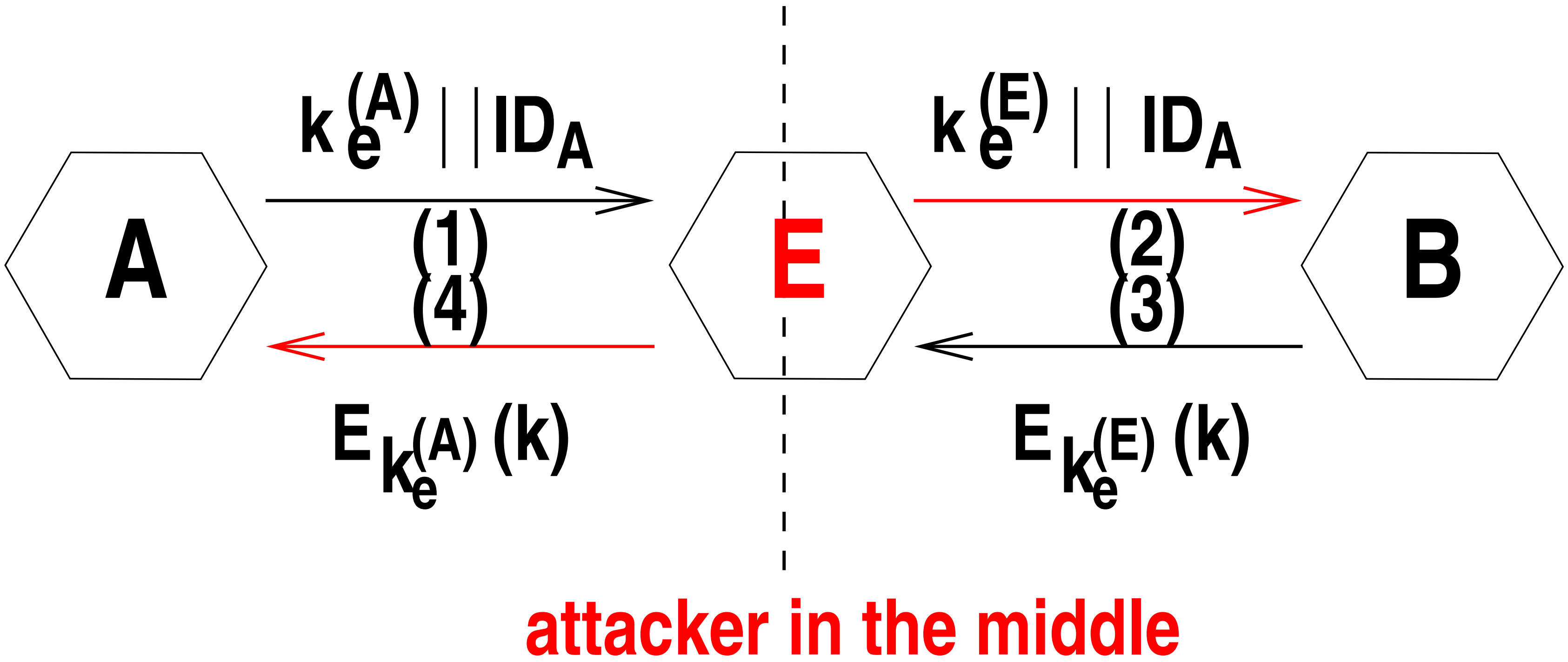
Question: What is the **active** attack?

Active Attack on the Merkel Protocol

1. A generates a key pair $\left(k_e^{(A)}, k_d^{(A)}\right)$, and sends $k_e^{(A)}||ID_A$ intended for B, where ID_A is an identifier of A.
2. E intercepts the message, creates its own key pair $\left(k_e^{(E)}, k_d^{(E)}\right)$, and sends $k_e^{(E)}||ID_A$ to B.
3. B generates a secret key k, and sends $E_{k_e^{(E)}}(k)$ (intended for A).
4. E intercepts the message, decrypts it to get k; then he computes and sends $E_{k_e^{(A)}}(k)$ to A.

Comment: A and B are unaware that E has got k.

The Intruder-in-the-Middle Attack: Pictorial

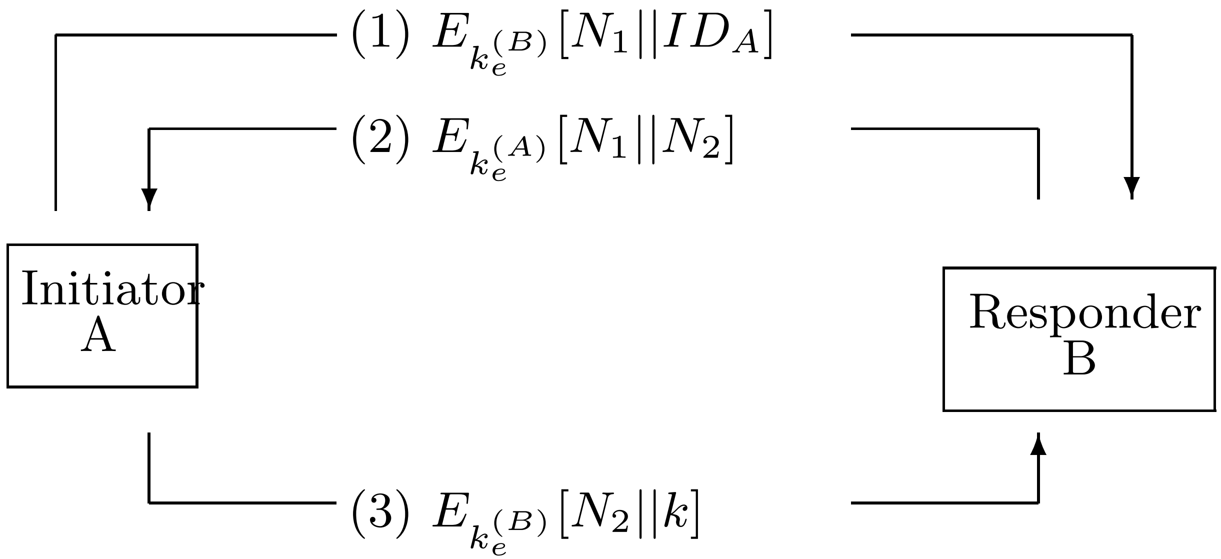


Active attack on the Merkel Protocol

The Needham-Schröder Protocol

For both confidentiality and authentication:

Assume that A and B have exchanged their public keys with some method.



Remarks: Nonce N_1 is to identify this transaction uniquely and is the challenge to B.

The Needham-Schröder Protocol

1. A sends $E_{k_e^{(B)}}[N_1||ID_A]$ to B, where N_1 is a nonce used to identify this transaction uniquely, and is generated by A.

2. B generates a new nonce N_2, and sends $E_{k_e^{(A)}}[N_1||N_2]$ to A. After decryption A gets N_1, and is sure that the responder is B.

3. A selects a secret key k and sends $E_{k_e^{(B)}}[N_2||k]$ to B.

   (Encryption with B's public key ensures confidentiality)

4. After decryption B gets N_2 and k, and is sure that its correspondent is A.

Remarks: This is a challenge-response protocol, which is a combination of a mutual authentication protocol and the digital envelop protocol.