Computer Security

Cunsheng Ding, HKUST

COMP4631

Lecture 14: Several Security Protocols

Main Topics of this Lecture

1. Authentication protocols and their classification.
2. A protocol for authentication and nonrepudiation.
3. A protocol for authentication, confidentiality and nonrepudiation.

Part I: Authentication Protocols and their Classification

Authentication Aspects

- Verify that the received message has not been altered (i.e., data authentication, also data integrity).
- Verify that the alleged sender is the real one (sender authentication, also data origin authentication).
- Verify the timeliness of messages.

A Basic Model of Authentication

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

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

When B receives the message c, he partitions c into $c = c_1||c_2$ and check whether $f(c_1) = c_2$. If yes, he concludes that c is indeed the message from A and it was not modified during transmission.

The part $f(m)$ is called the **authenticator**, while f is referred to as the **authentication function**. Usually the length of $f(m)$ is fixed.

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

Question: What is the price paid in this system?

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

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 $f(m)$ varies with that of m.

For example, the encryption transformation of a one-key cipher.

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

For example, 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.

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

Authentication checking proceedure: Left to the reader.

Authentication level: Depends on the security of the one-key cipher. If a secret key is used only once, it offers perfect authentication.

Advantages and disadvantages: High-level authentication, 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.

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

When receiving the data c, Bob partitions c into $c_1||c_2$, where c_2 has the same length as $E_k[h(m)]$. Bob then compares $h(c_1)$ with $D_k(c_2)$.

Conclusion: It provides a certain degree of authentication of both sender and message, but no confidentiality for message. Why?

Remark: The function $E_k \circ h$ is in fact a keyed hash function.

Security of Authentication Protocol 2

The first attack on the protocol: Observing $m||E_k[h(m)]$, an enemy E then randomly picks up an m', then replaces $m||E_k[h(m)]$ with a forged message $m'||E_k[h(m)]$ and sends it to Bob. E wishes that Bob accept it as the message from Alice.

Success probability: $\Pr(h(m) = h(m'))$.

Security requirements: The size of the hash value should be long enough. The hash values should be more or less "uniformly distributed".

Security of Authentication Protocol 2 – Continued

The second attack on the protocol: Observing $m||E_k[h(m)]$, an enemy E then tries to find an m' such that $h(m) = h(m')$. E then replaces $m||E_k[h(m)]$ with a forged message $m'||E_k[h(m)]$ and sends it to Bob.

Security requirement: For a given m, it should be computationally infeasible to find an m' such that

$$h(m) = h(m').$$

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 II: A Protocol for Authentication and Nonrepudiation

Authentication with Nonrepudiation

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

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

When receiving the data c, Bob partitions c into $c_1||c_2$, where c_2 has the same length as $D_{k_d^{(A)}}[h(m)]$. Bob then compares $h(c_1)$ with $E_{k_e^{(A)}}(c_2)$.

Conclusion: It provides a certain degree of authentication & nonrepudiation, but no confidentiality. Why?

Security requirements: The same as those in Authentication Protocol 2.

Part III: A Protocol for Authentication, Confidentiality and Nonrepudiation

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.

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

Bob verifies the sender, message, and signature similarly.

Exercise: Give details of the verification process.

Conclusion: It provides a certain degree of authentication, nonrepudiation, and confidentiality. $\boxed{\text{Why?}}$

Online question: Can we relieve the design requirements for h?