6G7Z1009: Introduction to Computer Forensics and Security

Key management-I

Reading List

- N. Ferguson, B. Schneier, T. Kohno, Cryptography Engineering: Design Principles and Practical Applications, (1st Edition) 2010, John Wiley.   Chapter 17, 18, 19
- W. Stallings, Cryptography and Network Security: Principles and Practice (5th Edition), 2010, Printice Hall (Chapter 14)
- M. Stamp, Information Security. Principles and Practice (2nd Edition), 2011, John Wiley. (Chapter 9 and 10)
- Behrouz Forouzan, Cryptography and Network Security,   The McGraw-Hill Companies. (Chapter 15)

Key distribution

- In previous lectures, we talk about symmetric-key and asymmetric-key cryptography but how we distribute the key?

Symmetric-key Distribution

- Symmetric-key cryptography is more efficient than asymmetric-key cryptography for enciphering large messages.Symmetric-key cryptography, however, needs a shared secret key between two parties.
- The distribution of keys is another problem. We need an efficient and reliable (trusted) way to maintain and distribute

Symmetric-key Distribution

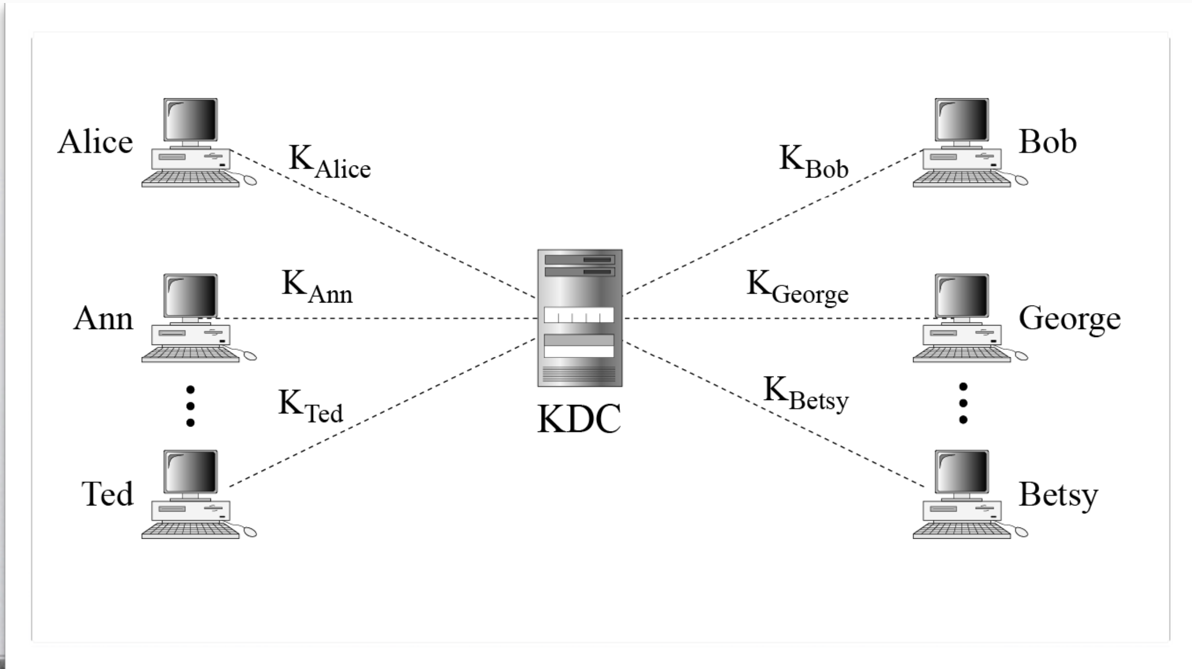
- Example:
- If Alice wants to exchange messages with N people, she needs N different symmetric (secret) keys. If N people need to communicate with each other, a total of N(N-1)/2 keys would be needed assuming a single key is used in both directions of communications between a pair of people. This is normally referred to as the N^2 problem.

Key Distribution Center

- In cryptography, a key distribution center (KDC) is part of a cryptosystem intended to reduce the risks inherent in exchanging keys.
- It consists of databases which hold every user's secret key. It involves users to request from a system to use services.

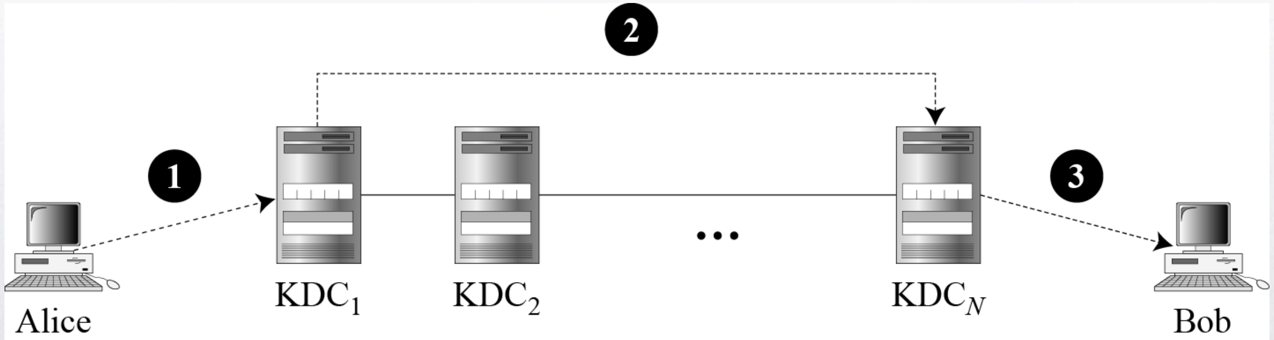
Key Distribution Center

- Each person establishes a shared key with the Key-distribution center (KDC).



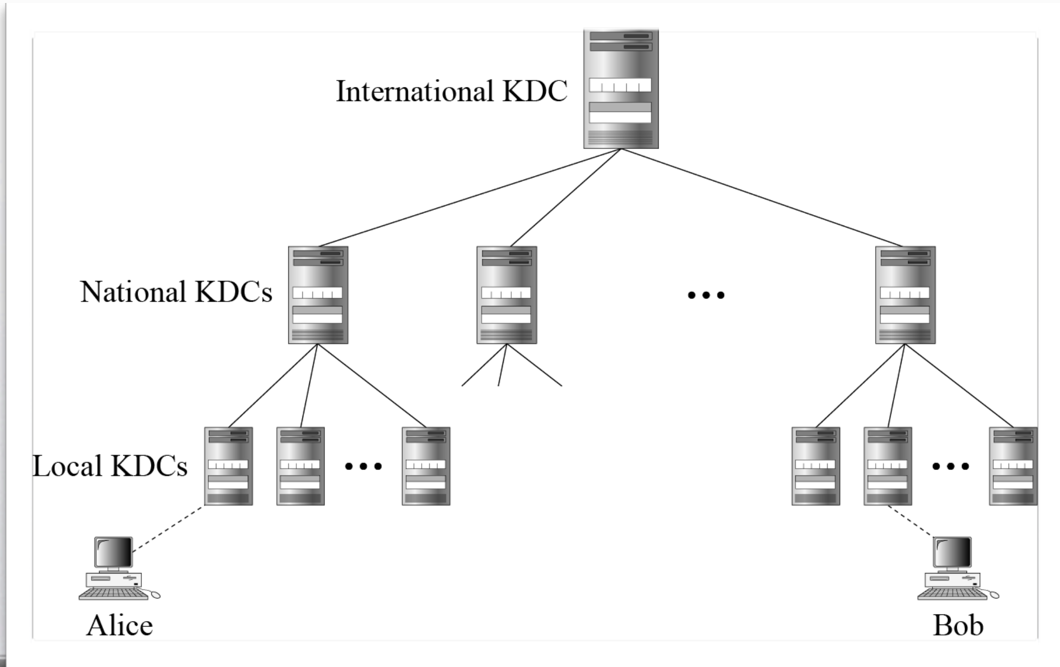
Key Distribution Center

- Types of key distributions:
  - Flat Multiple KDCs



Key Distribution Center

- Types of key distributions:
    - Hierarchical Multiple KDCs

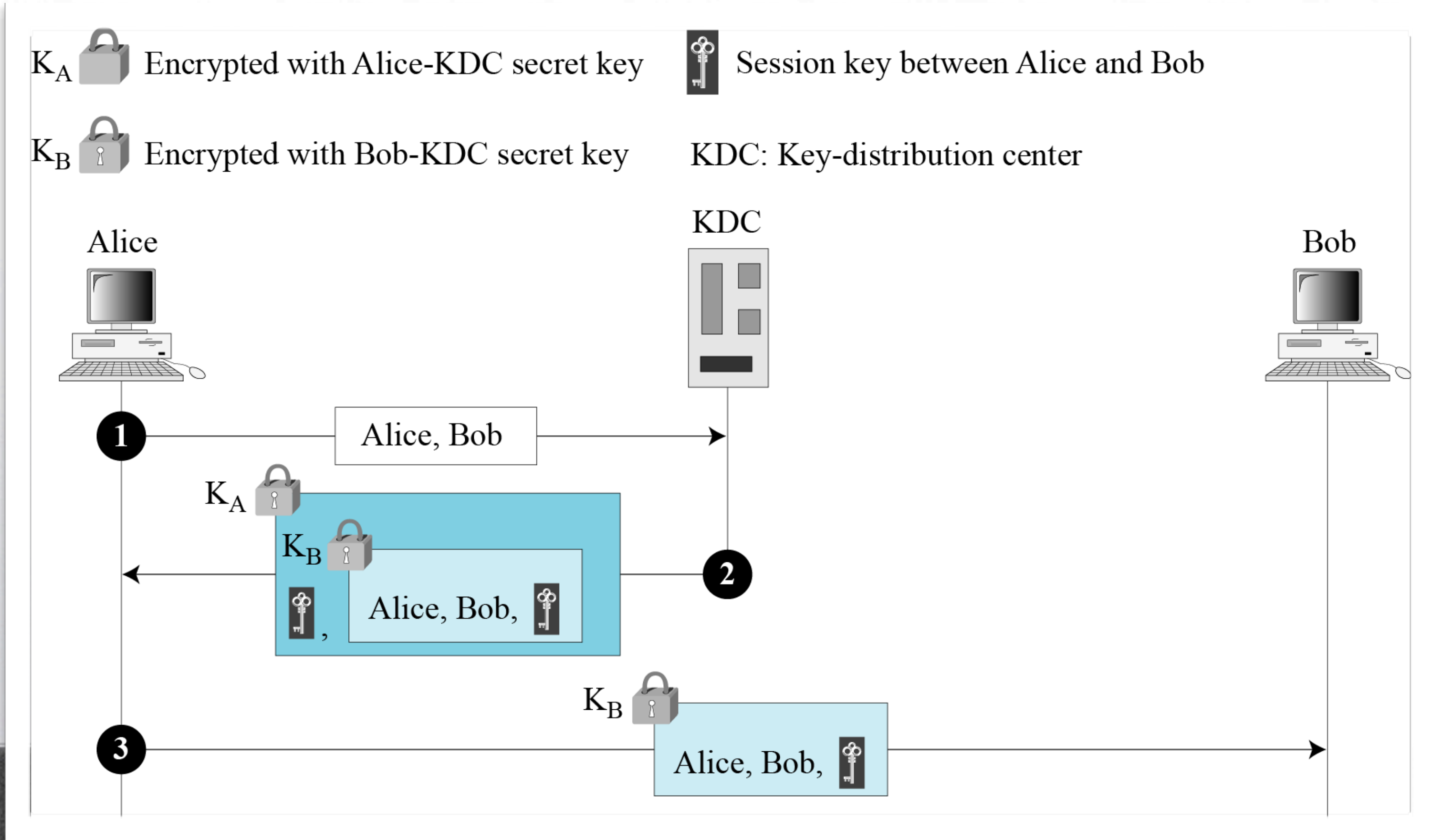


Session Keys

- A KDC creates a secret key for each member. This secret key can be used only between the member and the KDC, not between two members.
- A session symmetric key between two parties is used only once.

A Simple Protocol Using a KDC

- A Simple Protocol Using a KDC



A Simple Protocol Using a KDC

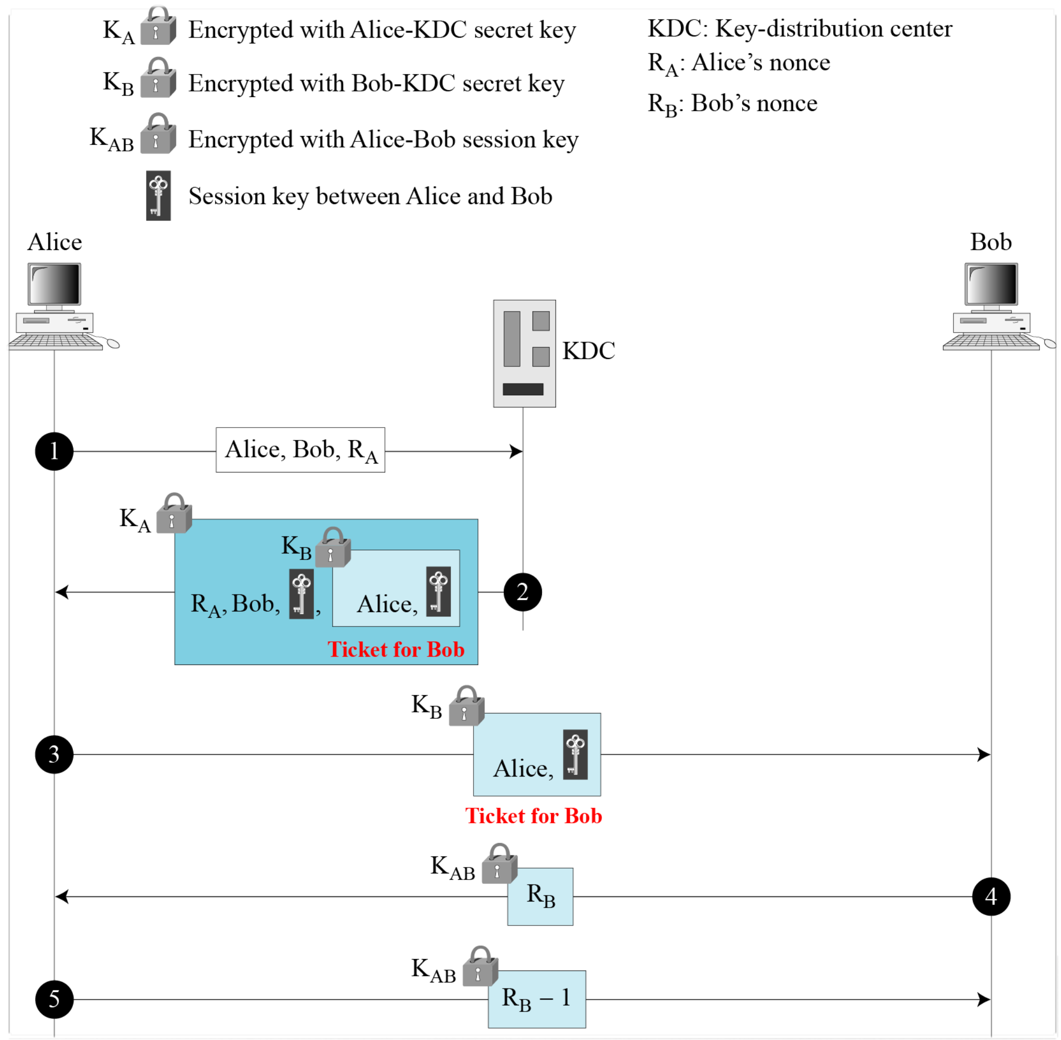
- Alice sends a plaintext message to KDC to request a symmetric session key between herself and Bob.
- The KDC creates a ticket encrypted using Bob's key K_B containing the session key. The ticket and the session
- key are sent to Alice in a message encrypted using Alice's key K_A. Alice decrypts the message and retrieves the session key and Bob's ticket.

A Simple Protocol Using a KDC

- Alice sends the ticket to Bob who decrypts (opens) the tickets and obtains the value of the session key
- This simple protocol is prone to replay attacks. An adversary can save the message (ticket) in step 3 and replay it later.

Needham Schroeder Protocol

- Needham Schroeder Protocol



Needham Schroeder Protocol

- Alice sends a message to KDC that includes her nonce R_A.
- The KDC sends an encrypted message to Alice that includes Alice's nonce, the session key, and an encrypted ticket to B that includes the session key. The ticket is encrypted using Bob's key and the whole message is encrypted using Alice's key.

Needham Schroeder Protocol

- Alice sends the ticket to Bob. Bob decrypts the ticket and sends his challenge R_B to Alice encrypted with the session key.
- Alice responds by sending to Bob the encrypted value R_{B-1} (rather than R_B to prevent replay attacks).

Needham Schroeder Protocol

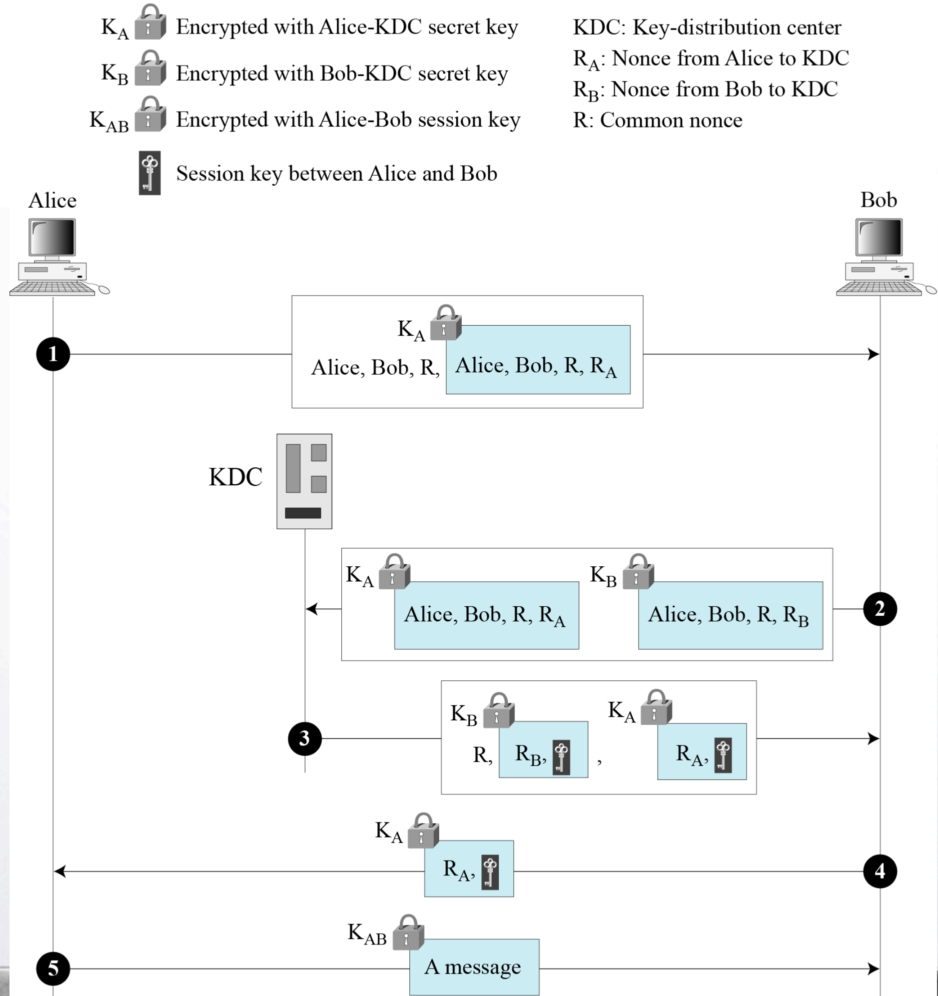
- What's vulnerability in Needham-Shroeder protocol?

Needham Schroeder Protocol

- If session key between A and B is compromised and the ticket to B is recorded, an intruder can impersonate A by carrying out last 3 steps.
- The weakness can be remedied by adding a timestamp to message 3, so that it becomes: A->B: $K_B\{A, t, K_{AB}\}$. B decrypts this message and checks that it is recent. This is the solution adopted in Kerberos

Otway-Rees Protocol

- Otway-Rees Protocol



Otway-Rees Protocol

- Alice sends a message to Bob that includes a common nonce R and her challenge R_A and a ticket to the KDC containing both R and R_A. The ticket is encrypted with Alice's secret key.
- Bob creates a similar ticket but with his own nounce R_B. Bob sends both tickets to KDC
- The KDC creates a message that contains R, a ticket for Alice with nounce R_A and a ticket for Bob with nonce R_B. The tickets contain the session key. The KDC sends the message to Bob

Otway-Rees Protocol

- Bob sends Alice her ticket
- Alice sends a short (hello) message encrypted with the session key to Bob