Authentication
Using Symmetric Keys

Haipeng Dai

haipengdai@nju.edu.cn
313 CS Building
Department of Computer Science and Technology
Nanjing University
Authentication Using Symmetric Keys

- **Assumption**
  - Without Trusted Third Party: each pair of parties share a secret key $K$
  - With Trusted Third Party: each party shares a secret key $K$ with the 3rd party

- **Threat model:**
  - Message injection
    - Inject a new message into a channel
  - Message modification
    - Modify a message in a channel
  - Message loss
    - Delete a message in a channel
  - Message replay
    - Replay an old message
Without Trusted Third Party – Version 1

- **Question:** What is wrong with this authentication protocol?
  - **Answer:** vulnerable to replay attack.
  - **How to fix this problem?**
Question: What is wrong with this authentication protocol?

- Answer: no session key is established, and no mutual authentication.
- How to fix this problem? - Add session key $k_{ab}$, and a nounce $m$ from Alice
Key management problems in authentication protocols without trusted third party:

1. Every pair of users need to have a shared secret key – too many keys.
2. Hard to manage: when a user joins a group, every one in the group needs to configure a new key with this person.

Solution: use a trusted third party – reduce $n^2$ keys to $n$ keys.
With Trusted Third Party – Version 1

- Version 1:

  - Question: Is this authentication protocol secure?
    - Answer: No. Vulnerable to man-in-the-middle attacks.
Attacks on Version 1 with 3rd Party

- Man-in-the-middle attack
  
  ![Diagram showing Man-in-the-middle attack]

- How to defend against this attack?
  - Solution: add principal name to prevent $\{n\}_{KA}$ from being reused by attacker.
Question: Is this authentication protocol secure?
   — Answer: No.
Attacks on Version 2 with 3rd Party

- How to defend against this attack?
  - Solution: add principal name into $\{n\}_{KB}$.
Question: Is this authentication protocol secure?
   — Answer: No.

Hint: message symmetry in authentication protocols is not good.
Attacks on Version 3 with 3rd Party

How to defend against this attack?

– Break symmetry.
With Trusted Third Party – Version 4

- We are almost there – except having mutual authentication.
With Trusted Third Party – Final Version

Alice

A

n

{n, B}_KA, m

{B, k_{ab}}_KA, {m}_{k_{ab}}

Bob

A

{A, {n, B}}_KA

{A, {n, B}}_KA

Authentication
Center

{B, k_{ab}}_KA, {m}_{k_{ab}}

{A, {n, B}}_KA

{A, {n, B}}_KA

{B, k_{ab}}_KA, {m}_{k_{ab}}
How to achieve secrecy and integrity?

Alice \[\{\text{data, MD(data)}\}_{K_{ab}}\] Bob
How to improve the performance?

- Change $\{\text{data}\}_{\text{PRA}}$ to $\{\text{MD(data)}\}_{\text{PRA}}$. 
Nonrepudiation – Version 2

How to further improve the performance?

- No need to encrypt \( \{\text{MD}(\text{data})\}_{PR_A} \) with \( K_{ab} \).
Nonrepudiation – Version 3

\{data\}_{K_{ab}}, \{\text{MD}(data)\}_{PR_A}