CS-3002: Information Security

Lecture # 8: Basic Key Exchange

Prof. Dr. Sufian Hameed
Department of Computer Science
FAST-NUCES

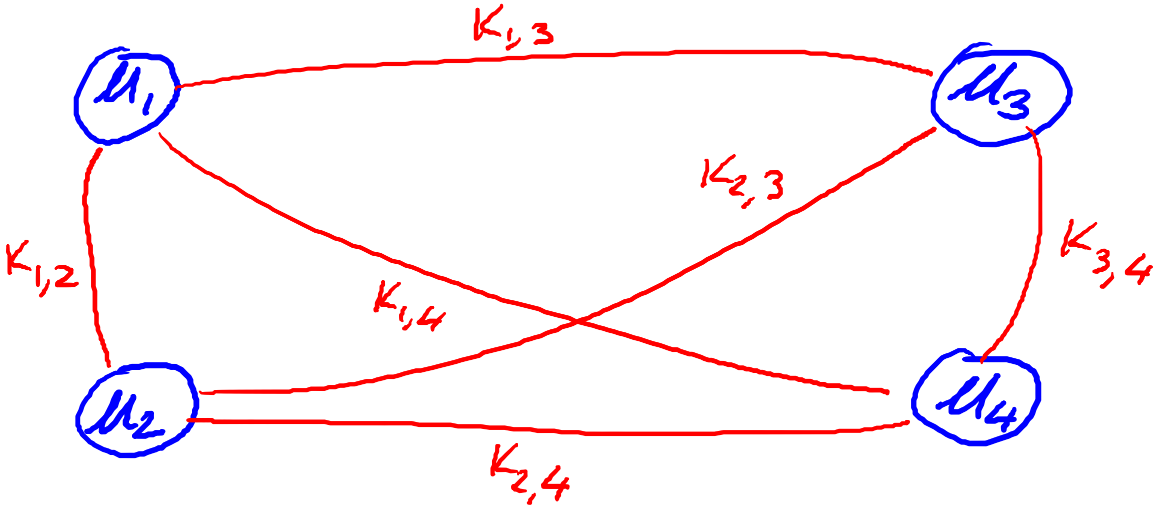
Overview

- *What will you learn today*
  - *Basic Key Exchange*
    - *Trusted 3rd party (introduce toy protocol)*
    - *Merkle Puzzle*
    - *The Diffie-Helmann Protocol*
    - *Public Key Encryption*

Trusted 3rd Parties

Key management

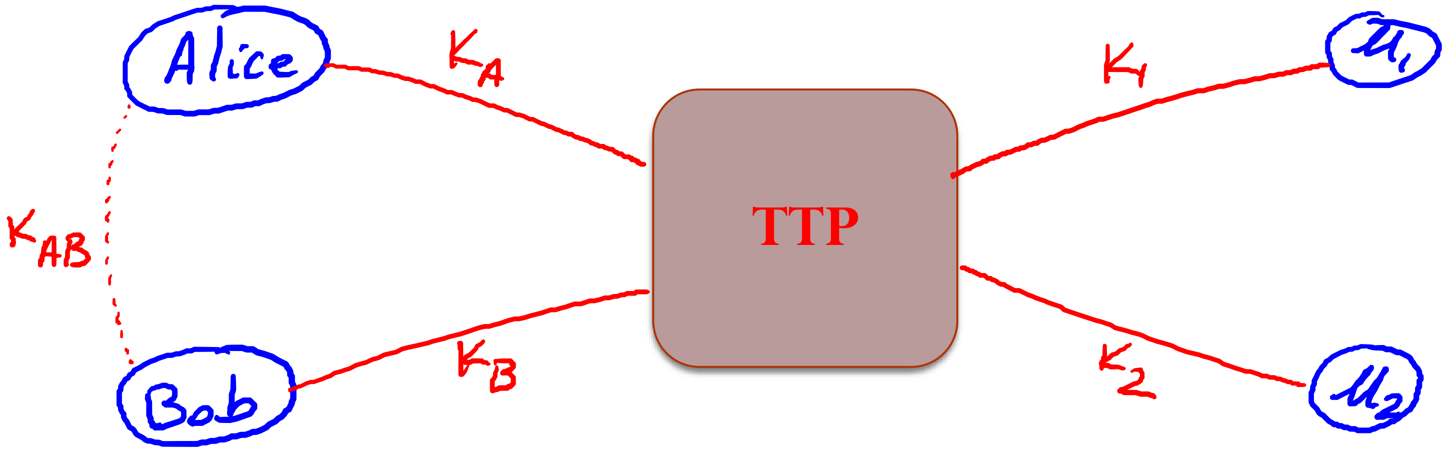
Problem: n users. Storing mutual secret keys is difficult



Total: O(n) keys per user

A better solution

Online Trusted 3rd Party (TTP)



Every user only remembers one key.

Generating keys: a toy protocol

Alice wants a shared key with Bob. Eavesdropping security only.

Bob (k_B) **Alice** (k_A) **TTP**

"Alice wants key with Bob" →

choose random k_{AB}

$E(k_A, \text{"A,B"} \| k_{AB})$ ←

$\text{ticket} \leftarrow E(k_B, \text{"A,B"} \| k_{AB})$

ticket ←

k_{AB} k_{AB}

(E,D) a CPA-secure cipher

Generating keys: a toy protocol

Alice wants a shared key with Bob. Eavesdropping security only.

Eavesdropper sees: $E(k_A, \text{“A, B”} \,\|\, k_{AB})$; $E(k_B, \text{“A, B”} \,\|\, k_{AB})$

(E,D) is CPA-secure $\Rightarrow$

eavesdropper learns nothing about k_{AB}

Note: TTP needed for every key exchange, knows all session keys.

(basis of Kerberos system)

Toy protocol: insecure against active attacks

Example: insecure against replay attacks

Attacker records session between Alice and merchant Bob

- For example a book order

Attacker replays session to Bob

- Bob thinks Alice is ordering another copy of book

Key question

Can we generate shared keys without an **online** trusted 3rd party?

Answer: yes!

Starting point of public-key cryptography:

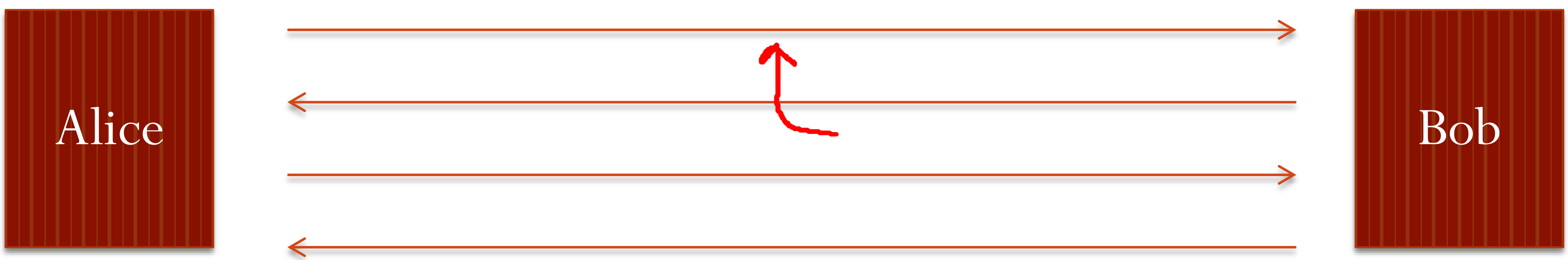
- Merkle (1974), Diffie-Hellman (1976), RSA (1977)
- More recently: ID-based enc. (BF 2001), Functional enc. (BSW 2011)

Merkle Puzzles

Key exchange without an online TTP?

Goal: Alice and Bob want shared key, unknown to eavesdropper

- For now: security against eavesdropping only (no tampering)



eavesdropper ??

Can this be done using generic symmetric crypto?

Merkle Puzzles (1974)

Answer: yes, but very inefficient

Main tool: puzzles

- Problems that can be solved with some effort
- Example: E(k,m) a symmetric cipher with $k \in \{0,1\}^{128}$
  - **puzzle(P) = E(P, "message")** where $P = 0^{96} \| b_1 \ldots b_{32}$
  - Goal: find P by trying all 2^{32} possibilities

Ralph Merkle design this a part of a seminar as an undergrad student.

Merkle puzzles

<u>Alice</u>: prepare 2^{32} puzzles

- For $i=1, \ldots, 2^{32}$ choose random $\mathbf{P_i} \in \{0,1\}^{32}$ and $\mathbf{x_i, k_i} \in \{0,1\}^{128}$

  set puzzle$_i$ $\leftarrow$ $E(\, 0^{96} \,\|\, \mathbf{P_i},$ **"Puzzle # x_i" ll k_i**)

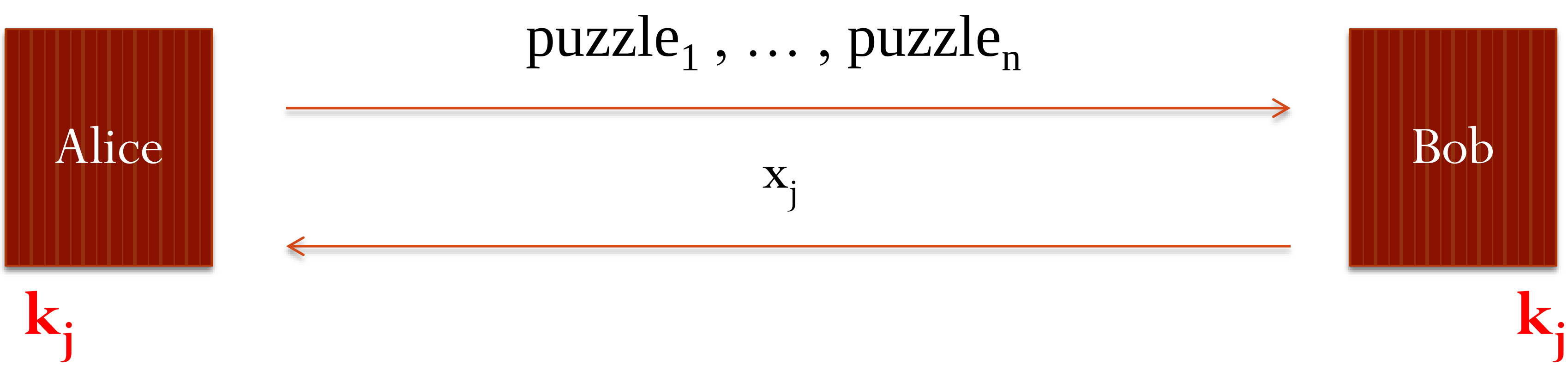
- Send puzzle$_1$, … , puzzle$_{2^{32}}$ to Bob

<u>Bob</u>:choose a random puzzle$_j$ and solve it. Obtain (x_j, k_j) .

- Send x_j to Alice

<u>Alice</u>: lookup puzzle with number x_j . Use k_j as shared secret

In a figure



Alice's work: $O(n)$ (prepare n puzzles)

Bob's work: $O(n)$ (solve one puzzle)

Eavesdropper's work: $O(n^2)$ (e.g. 2^{64} time)

Impossibility Result

Can we achieve a better gap using a general symmetric cipher?

Answer: unknown

But: roughly speaking,

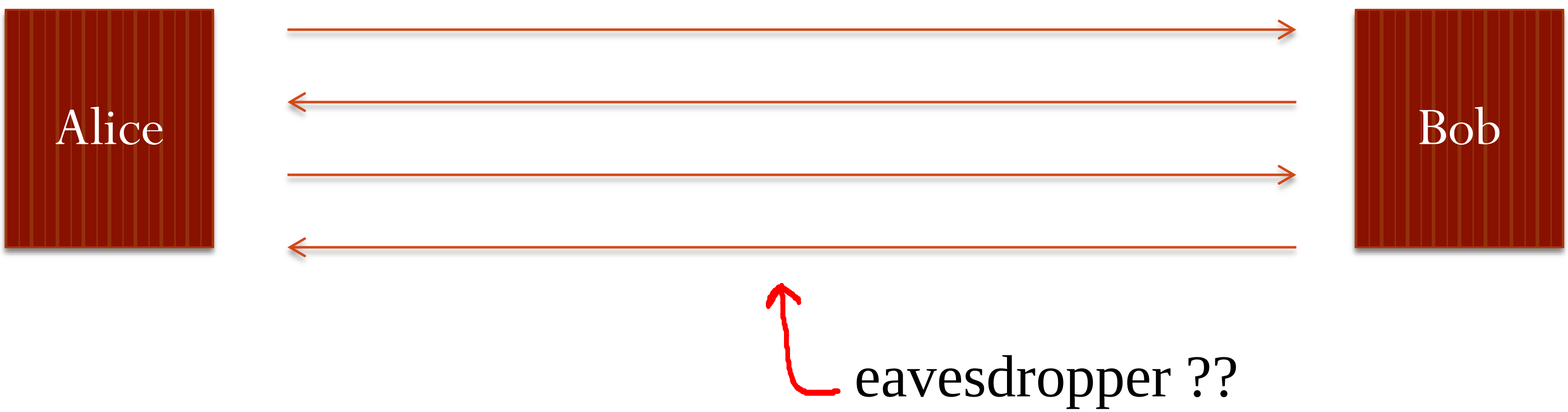
quadratic gap is best possible if we treat cipher as a black box oracle [IR’89, BM’09]

The Diffie-Hellman Protocol

Key exchange without an online TTP?

Goal: Alice and Bob want shared secret, unknown to eavesdropper

- For now: security against eavesdropping only (no tampering)



Can this be done with an exponential gap?

The Diffie-Hellman protocol (informally)

Fix a large prime p (e.g. 600 digits i.e 2K bits)

Fix an integer g in {1, …, p}

Alice **Bob**

choose random **a** in {1,…,p-1} choose random **b** in {1,…,p-1}

"Alice", $A \leftarrow g^a \pmod p$ $\longrightarrow$

$\longleftarrow$ "Bob", $B \leftarrow g^b \pmod p$

$$\mathbf{B^a} \pmod p \;=\; (g^b)^a \;=\; \mathbf{k_{AB} = g^{ab}} \pmod p \;=\; (g^a)^b \;=\; \mathbf{A^b} \pmod p$$

Security

Eavesdropper sees:

$p, g, \quad A=g^a \pmod{p}, \quad$ and $\quad B=g^b \pmod{p}$

Can she compute $\quad g^{ab} \pmod{p} \quad$??

More generally: define $\quad DH_g(g^a, g^b) = g^{ab} \quad \pmod{p}$

How hard is the DH function mod p?

How hard is the DH function mod p?

Suppose prime p is n bits long.

Best known algorithm (GNFS): run time exp($\tilde{O}(\sqrt[3]{n})$)

cipher key size	modulus size	Elliptic Curve size
80 bits	1024 bits	160 bits
128 bits	3072 bits	256 bits
256 bits (AES)	**15360** bits	512 bits

As a result: slow transition away from (mod p) to elliptic curves

Elliptic curve Diffie-Hellman

Insecure against man-in-the-middle

As described, the protocol is insecure against **active** attacks

Alice | **MiTM** | **Bob**

$A \leftarrow g^{a}$ $\longrightarrow$ a' $\quad$ $A' \leftarrow g^{a'}$ $\longrightarrow$

$\longleftarrow$ $B' \leftarrow g^{b'}$ $\quad$ b' $\longleftarrow$ $B \leftarrow g^{b}$

$g^{ab'}$ $\quad$ $g^{ab'}, g^{a'b}$ $\quad$ $g^{a'b}$

attacker relays traffic from Alice to Bob and reads it in the clear

Later we will see that it is not that difficult to enhance the protocol against MiTM attack

Further readings

- Merkle Puzzles are Optimal,
  B. Barak, M. Mahmoody-Ghidary, Crypto ’09

- On formal models of key exchange (sections 7-9)
  V. Shoup, 1999

Acknowledgements

Material in this lecture are taken from the slides prepared by:

- Prof. Dan Boneh (Standford)