CS 161: Computer Security

Lecture 9

October 6, 2015

Goal: establish secure channels

- We want to establish secure channels
- Problem is key management

Symmetric vs asymmetric encryption

- Asymmetric crypto has desirable properties
- But it is slow compared to symmetric crypto
- Can be up to 3-4 orders of magnitude slower

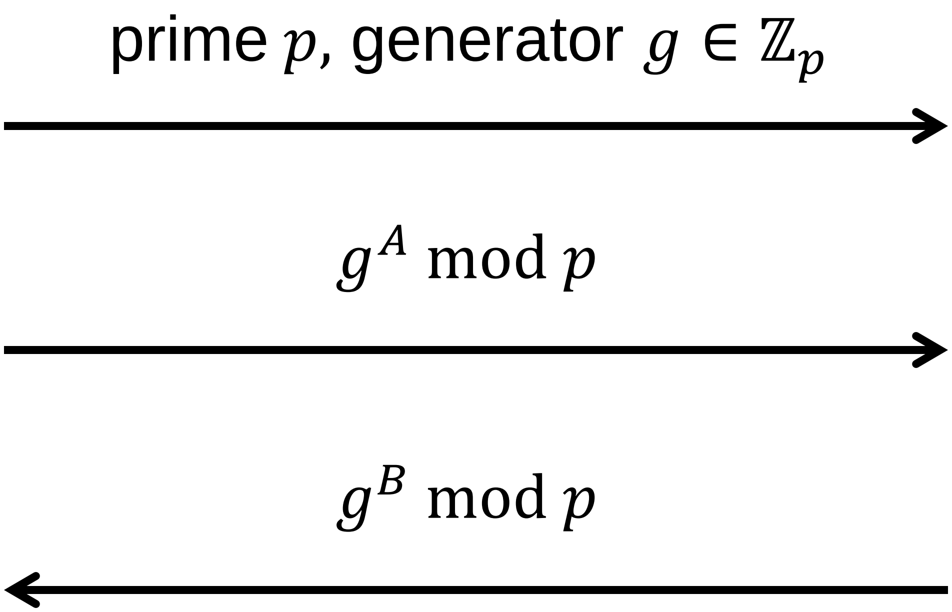
Pairwise key-exchange

- We can simply have every pair of communicating parties establish a private key in advance
- This requires $\binom{n}{2} = \frac{n(n-1)}{2}$ key pairs
- Too many!

Review: Diffie-Hellman key exchange

Alice

Bob



prime p, generator $g \in \mathbb{Z}_p$ →

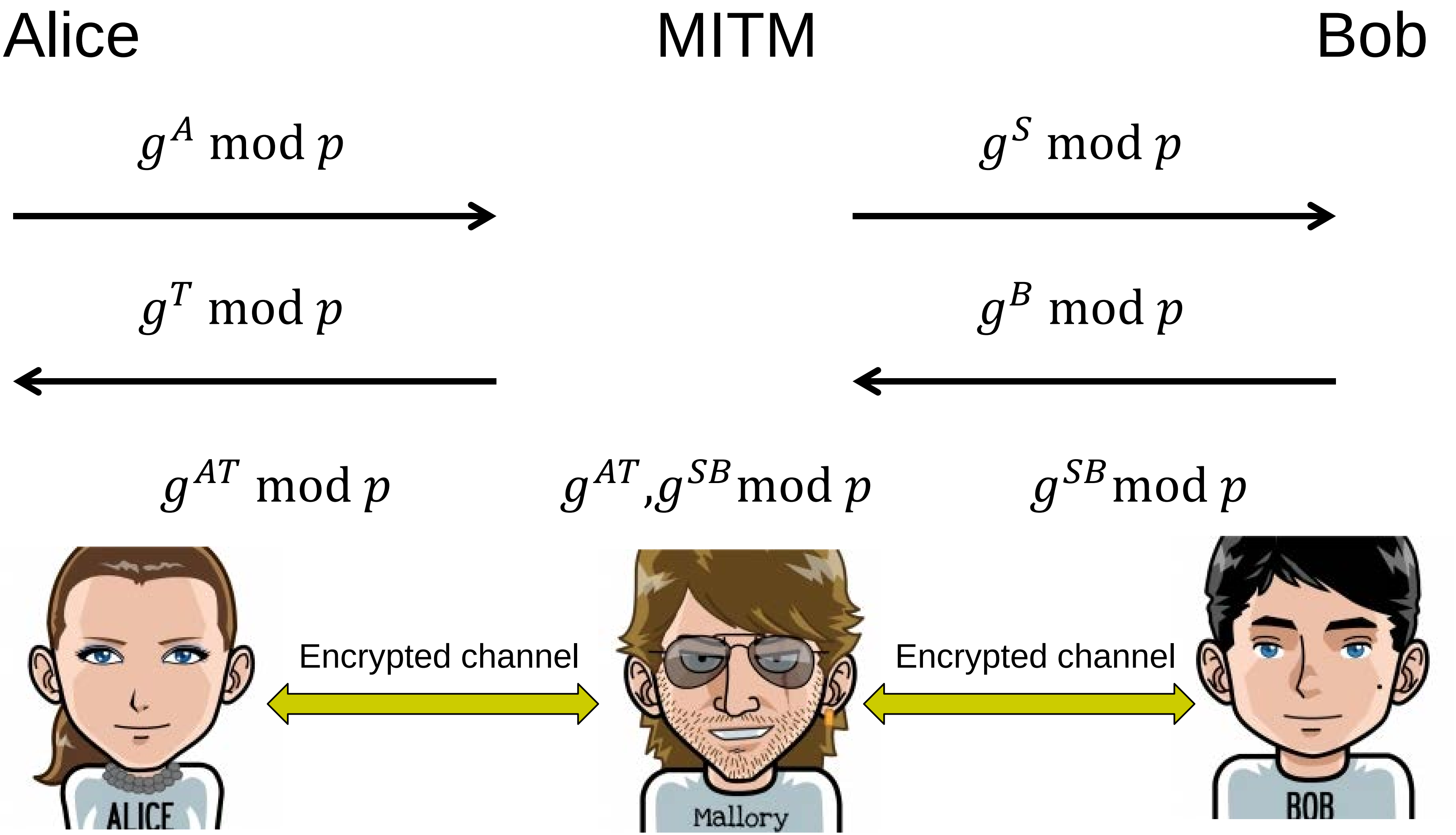
$g^A \bmod p$ →

← $g^B \bmod p$

$(g^B)^A \mod p$

$(g^A)^B \mod p$

Review: Man in the middle attack



Alice MITM Bob

$g^A \bmod p$ $\xrightarrow{\qquad}$ $g^S \bmod p$ $\xrightarrow{\qquad}$

$g^T \bmod p$ $\xleftarrow{\qquad}$ $g^B \bmod p$ $\xleftarrow{\qquad}$

$g^{AT} \bmod p$ $g^{AT}, g^{SB} \bmod p$ $g^{SB} \bmod p$

Encrypted channel

Encrypted channel

Authentication & key exchange

- Diffie-Hellman shows that key exchange is not sufficient
- We need to authenticate parties as well
- Coming up with good authentication and key exchange protocols is notoriously difficult

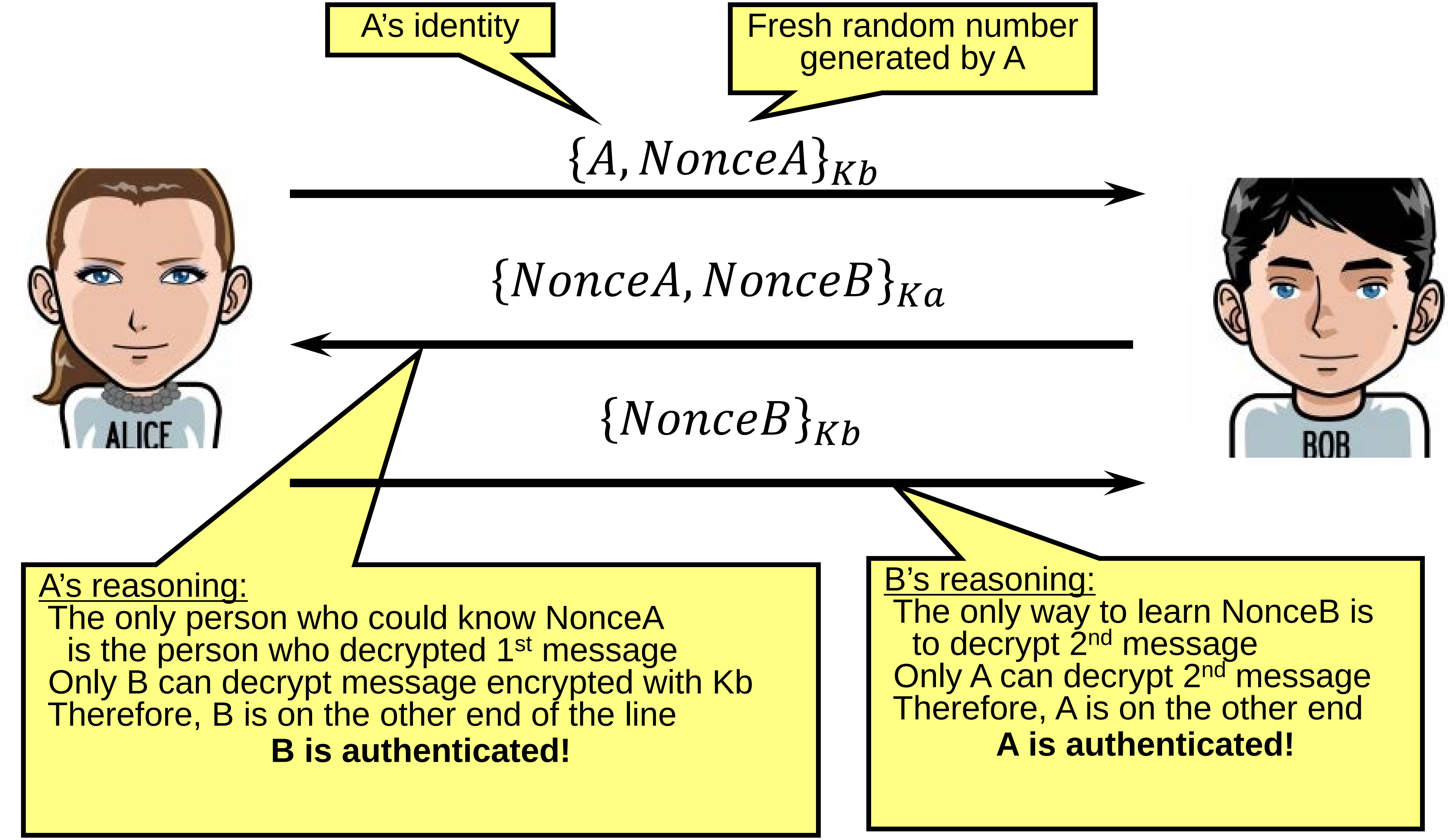
Needham-Schroeder

- Needham and Schroeder tried to develop authentication protocols in a 1979 paper
  - Asymmetric (public key)version
  - Symmetric (shared key) version
- We are still finding bugs in their protocols today!

Notation

- A: Alice's identity
- B: Bob's identity
- Ka: Alice's public key
- Kb: Bob's public key
- $\{m\}_{Ka}$: message m signed/encrypted in Ka
- $\{m\}_{Kb}$: message m signed/encrypted in Kb

Needham-Schroeder Asymmetric Protocol



A’s identity

Fresh random number generated by A

$\{A, NonceA\}_{Kb}$

$\{NonceA, NonceB\}_{Ka}$

$\{NonceB\}_{Kb}$

A’s reasoning:

The only person who could know NonceA is the person who decrypted 1st message

Only B can decrypt message encrypted with Kb

Therefore, B is on the other end of the line

B is authenticated!

B’s reasoning:

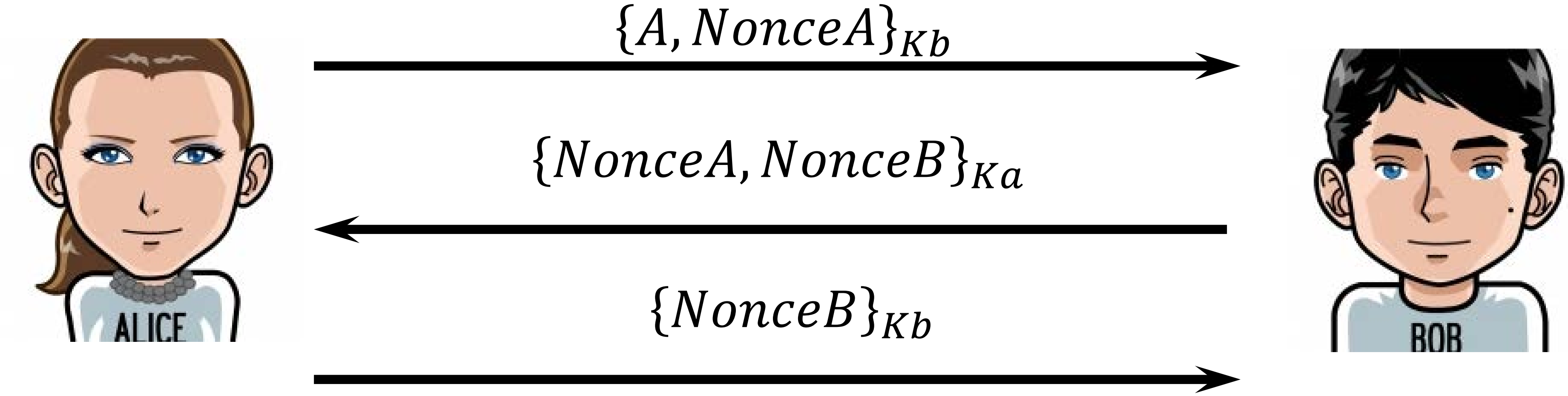
The only way to learn NonceB is to decrypt 2nd message

Only A can decrypt 2nd message

Therefore, A is on the other end

A is authenticated!

What Does This Protocol Achieve?



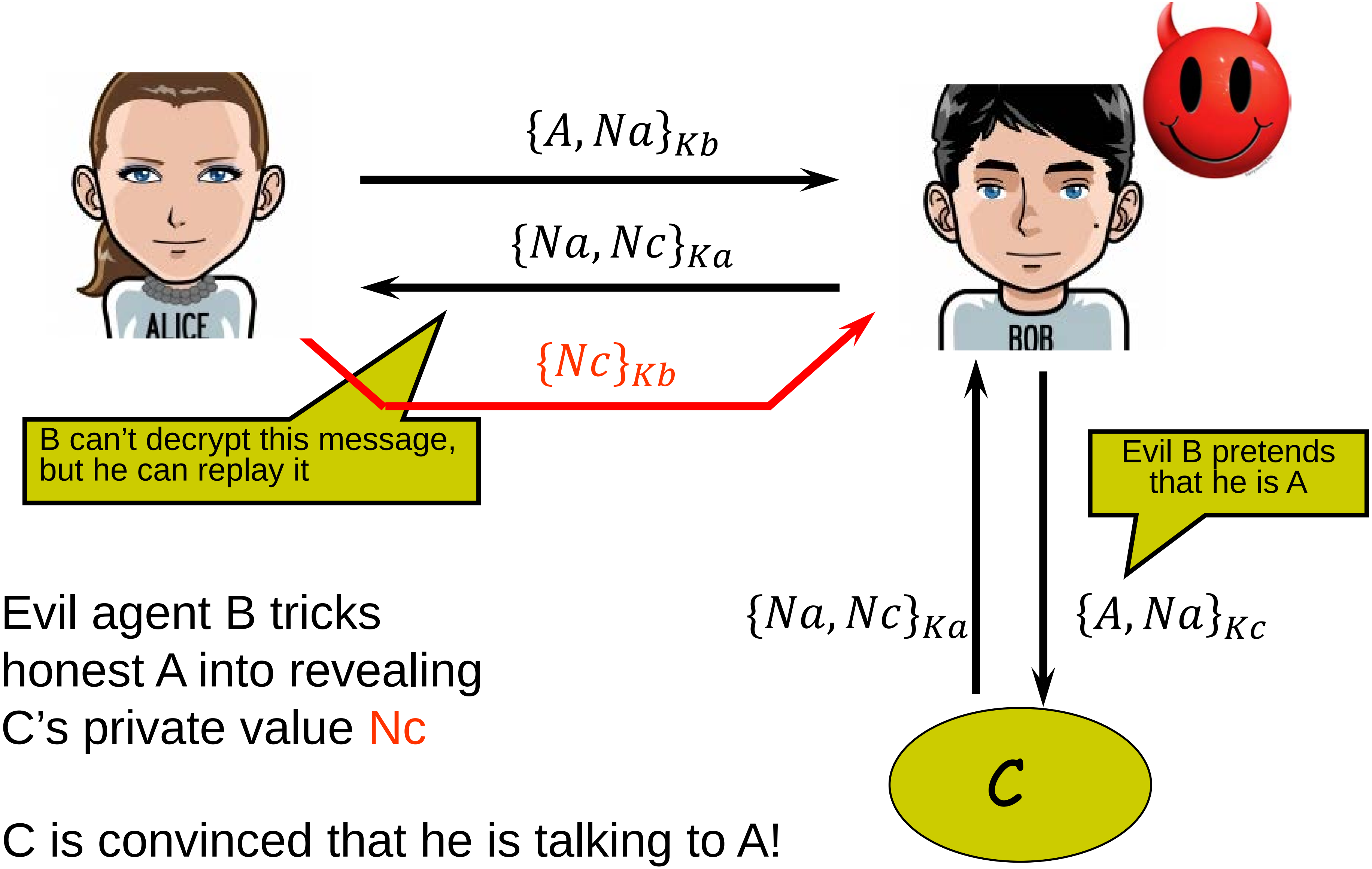
$\{A, NonceA\}_{Kb}$

$\{NonceA, NonceB\}_{Ka}$

$\{NonceB\}_{Kb}$

- Protocol aims to provide both authentication and secrecy
- After this the exchange, only A and B know Na and Nb
- Na and Nb can be used to derive a shared key

Anomaly in Needham-Schroeder



$\{A, Na\}_{Kb}$

$\{Na, Nc\}_{Ka}$

$\{Nc\}_{Kb}$

B can’t decrypt this message, but he can replay it

Evil B pretends that he is A

$\{Na, Nc\}_{Ka}$

$\{A, Na\}_{Kc}$

Evil agent B tricks honest A into revealing C’s private value Nc

C is convinced that he is talking to A!

Nonces

- We often use nonces often in security protocols
- Two approaches
    - Random values – guarantee freshness
    - Timestamps – require synchronized clocks

Needham-Schroeder Symmetric Protocol

- To use a symmetric protocol requires a trusted third party server to handle keys
  - TTP (for trusted third party)
  - S (for server)
- S is assumed to be trusted (honest)
- S already shares keys with parties:
  - Both Alice and S know Ka
  - Both Bob and S know Kb
  - $\{m\}_{Ka}$: message m signed/encrypted in Ka

Needham-Schroeder Symmetric Protocol

$A \rightarrow S$: A, B, Na

A requests S to supply key (Kab) for communication with B

$S \rightarrow A$: $\{Na, B, Kab, \{A, Kab\}_{Kb}\}_{Ka}$

S returns message encrypted in A's secret key Ka, containing session key Kab, and a ticket encrypted in B's secret key Kb

$A \rightarrow B$: $\{A, Kab\}_{Kb}$

A sends the ticket to B

$B \rightarrow A$: $\{Nb\}_{Kab}$

B decrypts the ticket and uses the new key Kab to encrypt another nonce Nb

$A \rightarrow B$: $\{Nb - 1\}_{Kab}$

A demonstrates to B that she was the sender of the previous message by returning an agreed transformation of Nb

Anomaly in Needham-Schroeder Symmetric Protocol

$A \rightarrow S\text{:}\ \ A, B, Na$

$S \rightarrow A\text{:}\ \ \{Na, B, Kab, \{A, Kab\}_{Kb}\}_{Ka}$

$A \rightarrow B\text{:}\ \ \{A, Kab\}_{Kb}$

$B \rightarrow A\text{:}\ \ \{Nb\}_{Kab}$

$A \rightarrow B\text{:}\ \ \{Nb - 1\}_{Kab}$

Suppose C cracks Kab from last week's run of protocol and has saved message 3 (ticket) from that session: $\{A, Kab\}_{Kb}$

$C \rightarrow B\text{:}\ \ \{A, Kab\}_{Kb}$

$B \rightarrow C\text{:}\ \ \{Nb\}_{Kab}$

$C \rightarrow B\text{:}\ \ \{Nb - 1\}_{Kab}$

B will believe he is talking to A

Anomaly in Needham-Schroeder Symmetric Protocol

- $A \rightarrow B$: $\{A, Kab\}_{Kb}$ not protected by nonces.
  - No way for B to know if Kab it receives is current

- Example attack: employee runs the first few steps of the protocols multiple times
  - Gathers tickets $\{A, Kab\}_{Kb}$ for servers B.
  - If she is fired, she can still login to all the servers

Ottway-Rees

M is a unique message identifier

$$A \rightarrow B\text{:}\ \ M, A, B, \{Na, M, A, B\}_{Ka}$$

$$B \rightarrow S\text{:}\ \ M, A, B, \{Na, M, A, B\}_{Ka}, \{Nb, M, A, B\}_{Kb}$$

$$S \rightarrow B\text{:}\ \ M, \{Na, Kab\}_{Ka}, \{Nb, Kab\}_{Kb}$$

$$B \rightarrow A\text{:}\ \ M, \{Na, Kab\}_{Ka}$$

Anomaly in Ottway-Rees

$A \to B\colon\ \ M, A, B, \{Na, M, A, B\}_{Ka}$

$B \to S\colon\ \ M, A, B, \{Na, M, A, B\}_{Ka},\ \{Nb, M, A, B\}_{Kb}$

$S \to B\colon\ \ M, \{Na, Kab\}_{Ka}, \{Nb, Kab\}_{Kb}$

$B \to A\colon\ \ M, \{Na, Kab\}_{Ka}$

Intruder blocks message 4;

Intruder replays message 2, captures message 3, and sends different key Kab' to A