Needham Schroeder Protocol

# Software Security

**Steffen Helke**
Chair of Software Engineering

9th January 2019

Brandenburgische
Technische Universität
Cottbus - Senftenberg

**Needham-Schroeder Protocol**
– Introduction –

## Objectives of today's lecture

- Getting to know different variants of the famous *Needham-Schroeder protocol*
- Understanding attack types like *Man-in-the-Middle* and *Replay* attack and possible countermeasures

## Needham-Schroeder Protocols (NSP)

- Developed by Rodger Needham and Michael Schroeder at the Xerox Palo Alto Research Center (MIT) in 1978
- Protocol family to support secure data exchange
- Providing *key exchange* and *authentication* mechanism
- Development of different variants for *symmetric and asymmetric encryption systems*

### Remarks

- The NSP family is not only interesting for historical reasons, but also forms the basis for modern security protocols
- Note that the asymmetric encryption variant had a design flaw that was found 17 years later

# Attack Types

**Man-in-the-Middle Attack**

- The attacker places himself between the communication partners Alice and Bob
- He has full control over the data traffic between Alice and Bob
- He can see/modify any information
- Attack is not detectable

**Replay Attack**

- Assumption: The attacker has found old keys and/or old tickets
- Attacker reuses old tickets from a previous session to manipulate the current communication

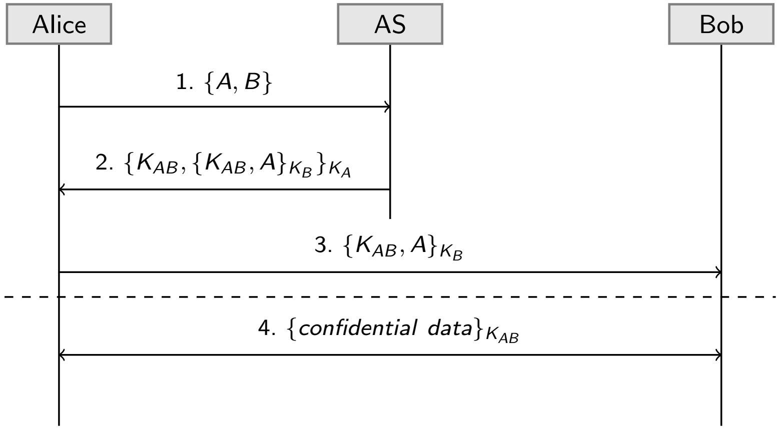
**Needham-Schroeder Protocol**

– Symmetric Encryption Variant –

# Preliminary Specifications

- $A$: Identity of Alice
- $B$: Identity of Bob
- $K_{AB}$: Symmetric session key of Alice and Bob
- $AS$: Authentication server, is trustworthy, generates and distributes the session key $K_{AB}$
- $K_A$: Symmetric key between $AS$ and $A$
- $K_B$: Symmetric key between $AS$ and $B$
- $N_A$ and $N_B$: Nonces (*number used one* or *number once*), random numbers used for only one protocol session

# Naive Variant of the Symmetric NSP



Alice

AS

Bob

1. $\{A, B\}$

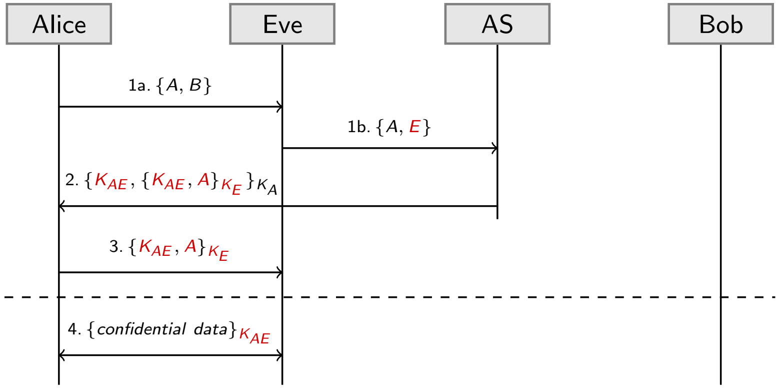
2. $\{K_{AB}, \{K_{AB}, A\}_{K_B}\}_{K_A}$

3. $\{K_{AB}, A\}_{K_B}$

4. $\{\textit{confidential data}\}_{K_{AB}}$

The naive variant of the NSP is not secure! Why?

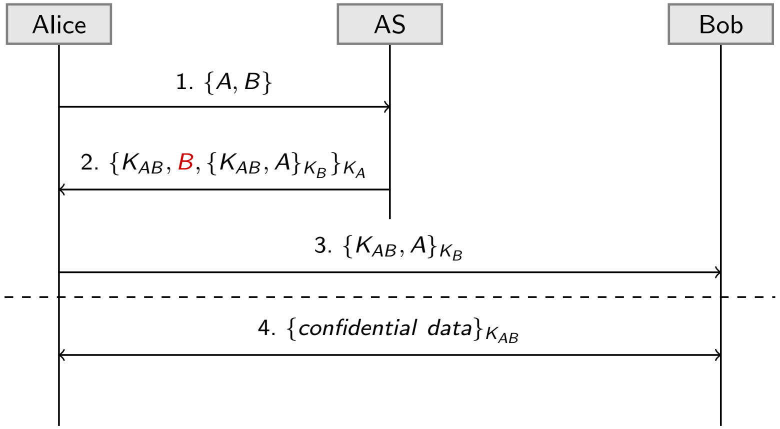
## Attack for the naive Symmetric NSP


Alice
Eve
AS
Bob
1a. $\{A, B\}$
1b. $\{A, E\}$
2. $\{K_{AE}, \{K_{AE}, A\}_{K_E}\}_{K_A}$
3. $\{K_{AE}, A\}_{K_E}$
4. $\{confidential\ data\}_{K_{AE}}$

Eve is pretending to Alice to be Bob! Countermeasures?

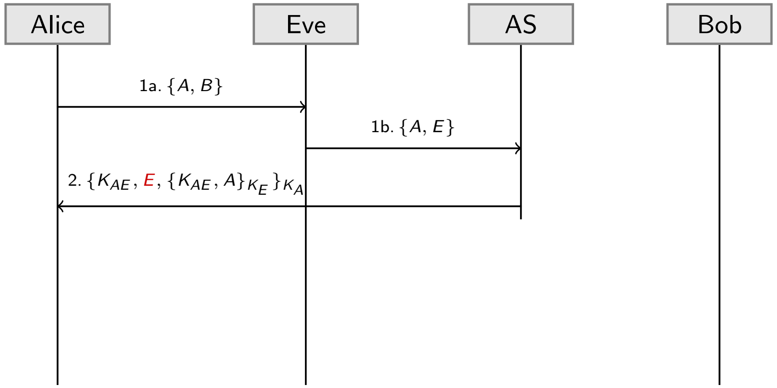
➜ Man-in-the-middle attacks can be prevented by sending identities inside the tickets!

## Variant 2 for the Symmetric NSP


Alice
AS
Bob
1. $\{A, B\}$
2. $\{K_{AB}, B, \{K_{AB}, A\}_{K_B}\}_{K_A}$
3. $\{K_{AB}, A\}_{K_B}$
4. $\{confidential\ data\}_{K_{AB}}$

By specifying Bob's identity in step 2, Alice is able to detect the Man-in-the-middle attack!

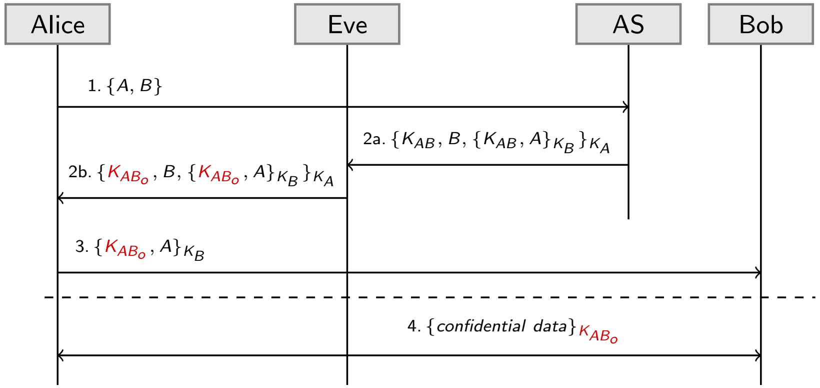
## Detecting a Man-in-the-middle Attack



Alice detects in Step 2 that the ticket of the authentication server has been manipulated and cancels the session!

➜ Is the protocol secure now? No, because replay attacks are possible

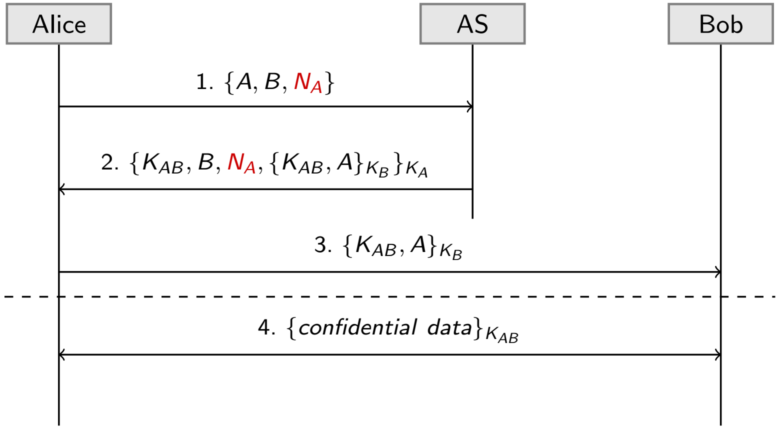
## Attack for Variant 2 of the Symmetric NSP


Alice
Eve
AS
Bob
1. $\{A, B\}$
2a. $\{K_{AB}, B, \{K_{AB}, A\}_{K_B}\}_{K_A}$
2b. $\{K_{AB_o}, B, \{K_{AB_o}, A\}_{K_B}\}_{K_A}$
3. $\{K_{AB_o}, A\}_{K_B}$
4. $\{confidential\ data\}_{K_{AB_o}}$

Assumption: Eve knows the old session key $K_{AB_o}$ of Alice & Bob and also the corresponding ticket of the authentication server

➜ How to prevent such a replay attack?

# Variant 3 for the Symmetric NSP



Alice

AS

Bob

1. $\{A, B, N_A\}$

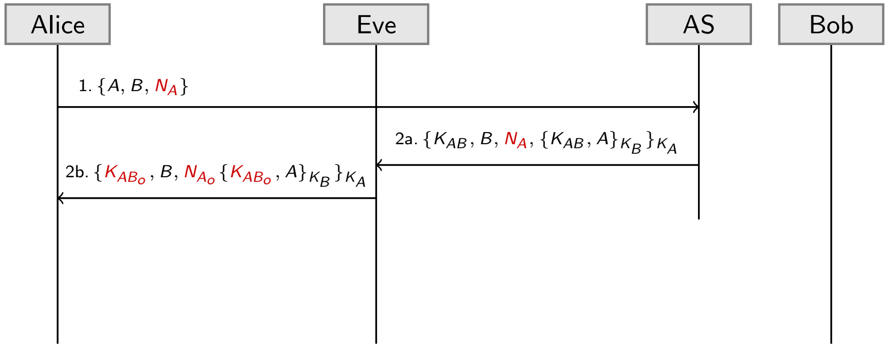
2. $\{K_{AB}, B, N_A, \{K_{AB}, A\}_{K_B}\}_{K_A}$

3. $\{K_{AB}, A\}_{K_B}$

4. $\{confidential\ data\}_{K_{AB}}$

By using Nonce $N_A$ (number used once), a correlation between Step 1 and Step 2 is implemented, such that Alice is able to check the freshness of the received ticket

# Detecting a Replay Attack



Alice

Eve

AS

Bob

1. $\{A, B, N_A\}$

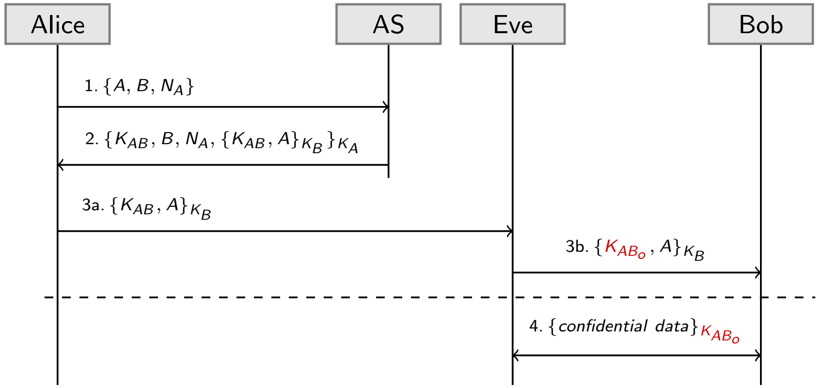
2a. $\{K_{AB}, B, N_A, \{K_{AB}, A\}_{K_B}\}_{K_A}$

2b. $\{K_{AB_o}, B, N_{A_o} \{K_{AB_o}, A\}_{K_B}\}_{K_A}$

Alice detects Eve's manipulation by finding out that the Nonce $N_A$ has been changed

➜ Is the protocol now finally secure? No, because Bob is attackable!

# Attack for Variant 3 of the symmetric NSP



Alice

AS

Eve

Bob

1. $\{A, B, N_A\}$

2. $\{K_{AB}, B, N_A, \{K_{AB}, A\}_{K_B}\}_{K_A}$

3a. $\{K_{AB}, A\}_{K_B}$

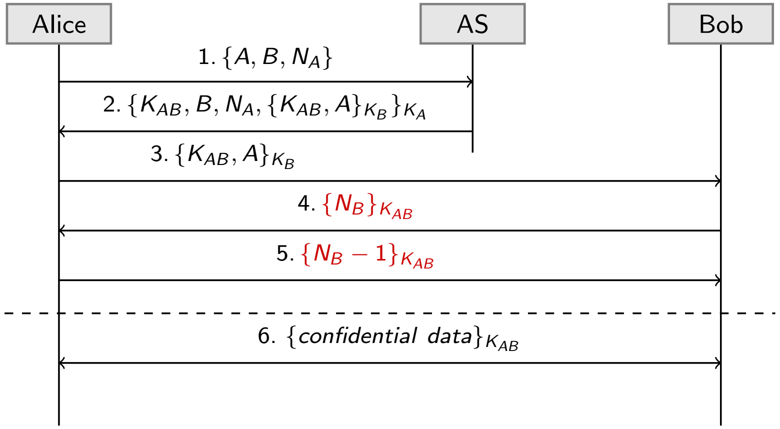
3b. $\{K_{AB_o}, A\}_{K_B}$

4. $\{confidential\ data\}_{K_{AB_o}}$

Assumption: Eve knows the old session key $K_{AB_o}$ of Alice & Bob and also the corresponding ticket of Step 3b

➜ How can this replay attack against Bob be prevented?

# Variant 4: Symmetric NSP with Handshake



Alice

AS

Bob

1. $\{A, B, N_A\}$

2. $\{K_{AB}, B, N_A, \{K_{AB}, A\}_{K_B}\}_{K_A}$

3. $\{K_{AB}, A\}_{K_B}$

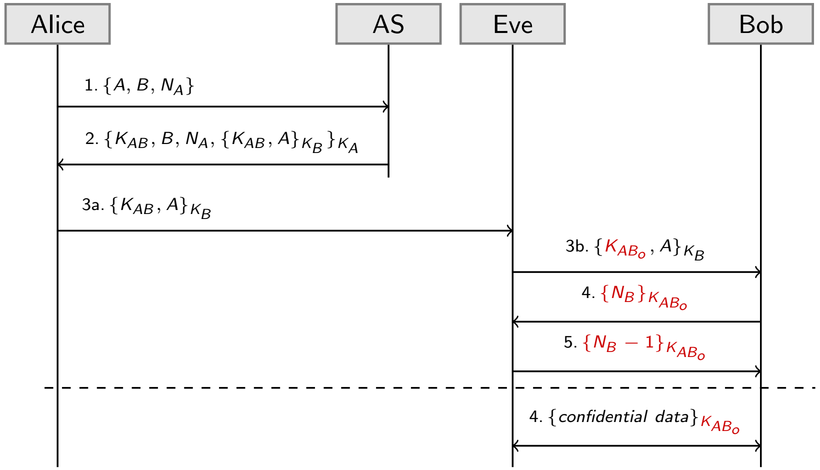
4. $\{N_B\}_{K_{AB}}$

5. $\{N_B - 1\}_{K_{AB}}$

6. $\{confidential\ data\}_{K_{AB}}$

The handshake implemented in the original NSP does *not offer* Bob additional protection against replay attacks! Why?

# Attack of Variant 4 of the Symmetric NSP



Alice AS Eve Bob

1. $\{A, B, N_A\}$

2. $\{K_{AB}, B, N_A, \{K_{AB}, A\}_{K_B}\}_{K_A}$

3a. $\{K_{AB}, A\}_{K_B}$

3b. $\{K_{AB_o}, A\}_{K_B}$

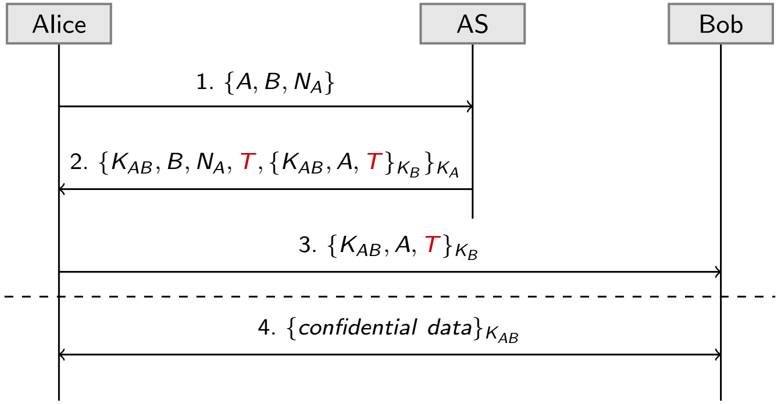
4. $\{N_B\}_{K_{AB_o}}$

5. $\{N_B - 1\}_{K_{AB_o}}$

4. $\{confidential\ data\}_{K_{AB_o}}$

How to uncover the replay attack against Bob?

# Variant 5: Symmetric NSP & Time Stamps



Alice AS Bob

1. $\{A, B, N_A\}$

2. $\{K_{AB}, B, N_A, T, \{K_{AB}, A, T\}_{K_B}\}_{K_A}$

3. $\{K_{AB}, A, T\}_{K_B}$

4. $\{confidential\ data\}_{K_{AB}}$

A time stamp $T$ gives information about the freshness of tickets and enables Bob to detect replay attacks

➜ Is Bob protected now? No! You could also manipulate time!

# Attacks on Protocols with Time Stamps

**We assume that ...**

- the local clock of the target system can be manipulated or
- a time service (e.g. of a time server) can be manipulated

**Procedure**

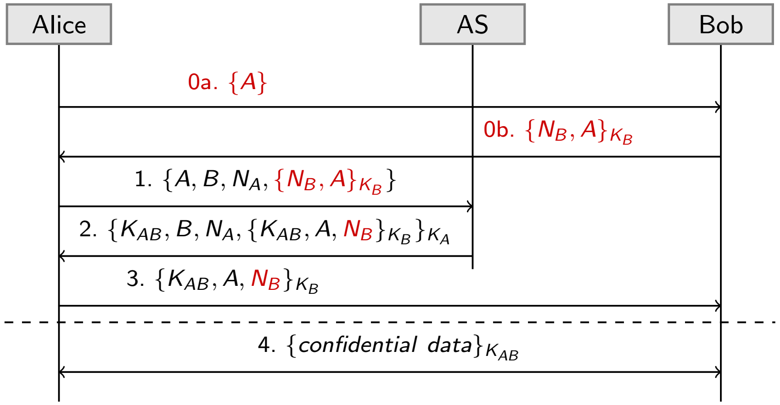
1. Modify the time of your target system
2. Perform a replay attack

**How to protect?**

➜ Use of previously negotiated nonces also for Bob

➜ Disadvantage: The protocol is getting more complicated

# Variant 6 of the Symmetric NSP with Nonces



Alice AS Bob

0a. $\{A\}$

0b. $\{N_B, A\}_{K_B}$

1. $\{A, B, N_A, \{N_B, A\}_{K_B}\}$

2. $\{K_{AB}, B, N_A, \{K_{AB}, A, N_B\}_{K_B}\}_{K_A}$

3. $\{K_{AB}, A, N_B\}_{K_B}$

4. $\{confidential\ data\}_{K_{AB}}$

This variant of the NSP prevents replay attacks against Alice & Bob and allows to detect man-in-the-middle attacks

## Needham-Schroeder Protocols

– Asymmetric Variants –

# Preliminary Specifications

### Given Keys

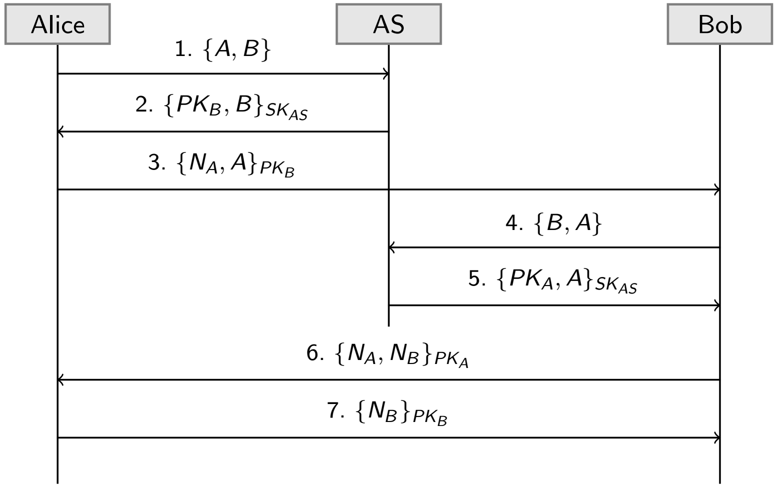
1. $PK_{AS}$: Public key of the authentication server $AS$
2. $SK_{AS}$: Secret key of the authentication server $AS$
3. $PK_A$ and $PK_B$: Public keys of Alice and Bob
4. $SK_A$ and $SK_B$: Secret keys of Alice and Bob

### Assumptions

- $AS$ knows the public keys of all participants
- All participants only know the public key $PK_{AS}$ before the protocol is started

➜ Participants must request all other required keys from $AS$

# Asymmetric Variant of the NSP



Alice AS Bob

1. $\{A, B\}$

2. $\{PK_B, B\}_{SK_{AS}}$

3. $\{N_A, A\}_{PK_B}$

4. $\{B, A\}$

5. $\{PK_A, A\}_{SK_{AS}}$

6. $\{N_A, N_B\}_{PK_A}$

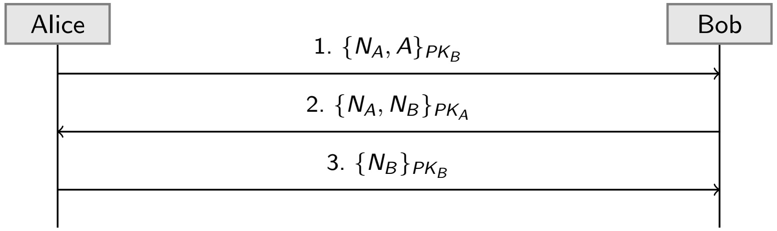
7. $\{N_B\}_{PK_B}$

The protocol is not secure against man-in-the-middle attacks! Why? Find the attack scenario!

# Simplified Version of the Asymmetric NSP

➜ Assumption: Participants have already received all required public keys from the AS

➜ Therefore, Steps 1,2,4 & 5 can be omitted



Alice Bob

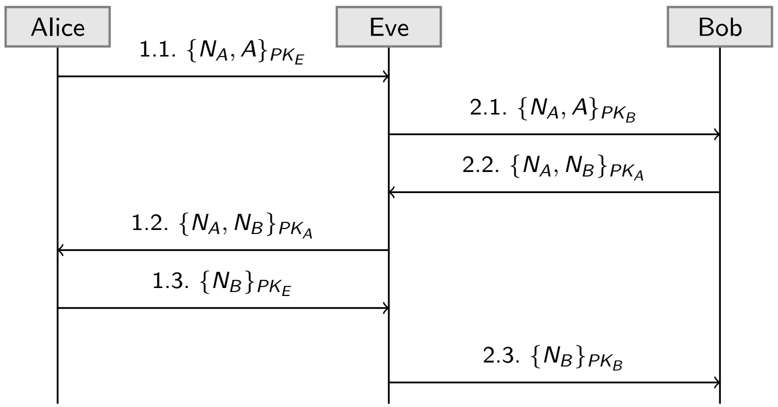
1. $\{N_A, A\}_{PK_B}$

2. $\{N_A, N_B\}_{PK_A}$

3. $\{N_B\}_{PK_B}$

Note: The attacker Eve executes two of these protocol sessions in parallel to perform the attack!

# Attack for the Asymmetric Variant of the NSP



Alice

Eve

Bob

1.1. $\{N_A, A\}_{PK_E}$

2.1. $\{N_A, A\}_{PK_B}$

2.2. $\{N_A, N_B\}_{PK_A}$

1.2. $\{N_A, N_B\}_{PK_A}$

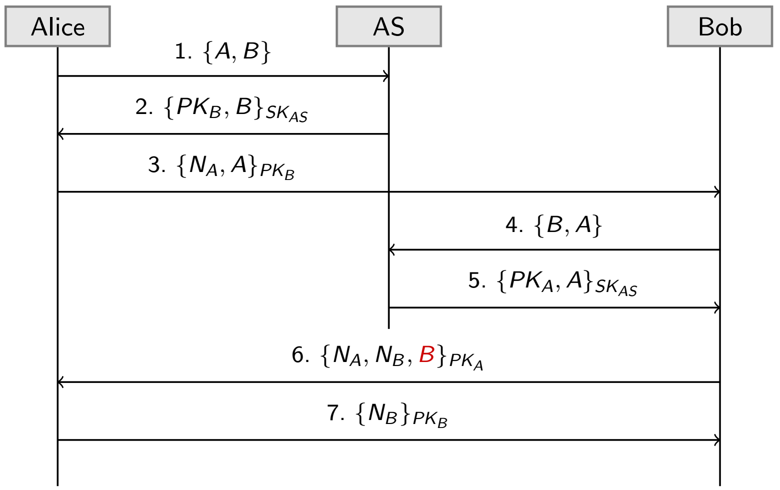
1.3. $\{N_B\}_{PK_E}$

2.3. $\{N_B\}_{PK_B}$

Eve cheats on Bob. She pretends to be Alice in reality.

➜ How to protect Bob? How to adapt the protocol?

# Corrected Variant of the Asymmetric NSP



Alice

AS

Bob

1. $\{A, B\}$

2. $\{PK_B, B\}_{SK_{AS}}$

3. $\{N_A, A\}_{PK_B}$

4. $\{B, A\}$

5. $\{PK_A, A\}_{SK_{AS}}$

6. $\{N_A, N_B, B\}_{PK_A}$

7. $\{N_B\}_{PK_B}$

Sending Bob's identity in Step 6 enables Alice to detect the man-in-the-middle attack