Needham Schroeder Protocol

Software Security

Steffen Helke
Chair of Software Engineering

9th January 2019

Brandenburgische
Technische Universität
Cottbus - Senftenberg

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 Protocol

– Introduction –

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

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

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*

Needham-Schroeder Protocols (NSP)

Why is it so important to use secure protocols in addition to secure encryption algorithms?

- ➜ 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*

Variants: a form or version of something that differs in some respect from other forms of the same thing or from a standard

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

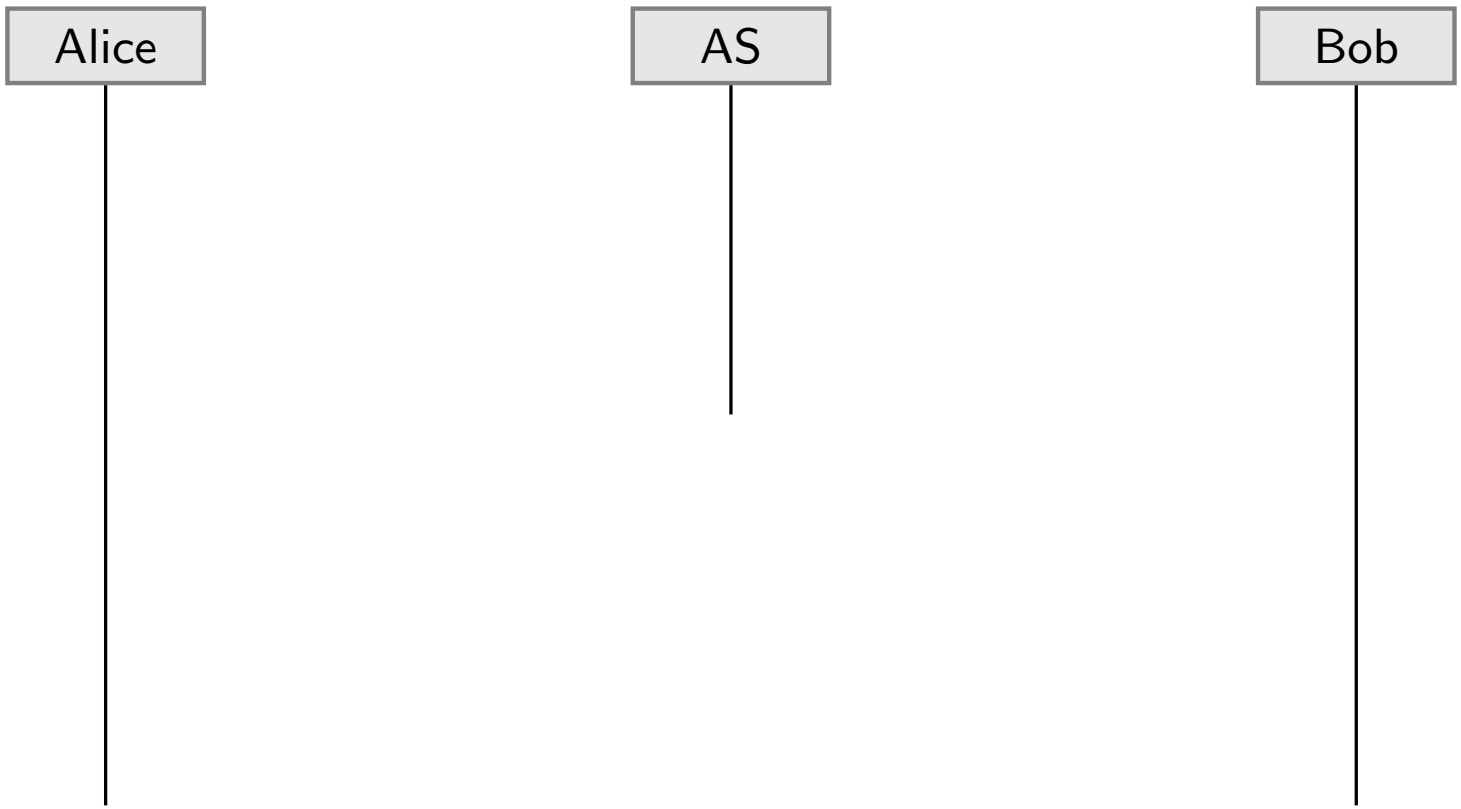
– Symmetric Encryption Variant –

Preliminary Specifications

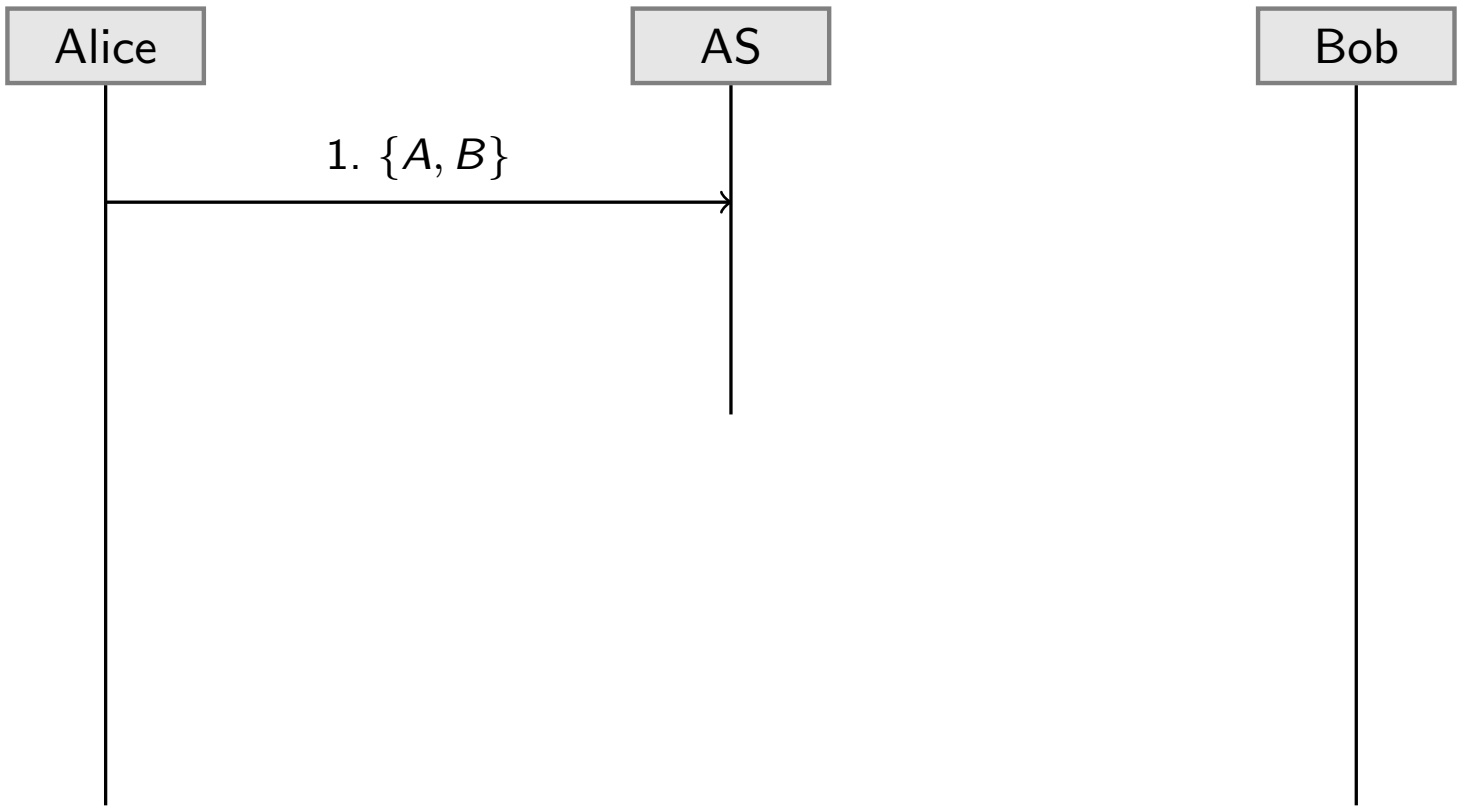
Which notation is usually used to specify a security protocol?

- 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

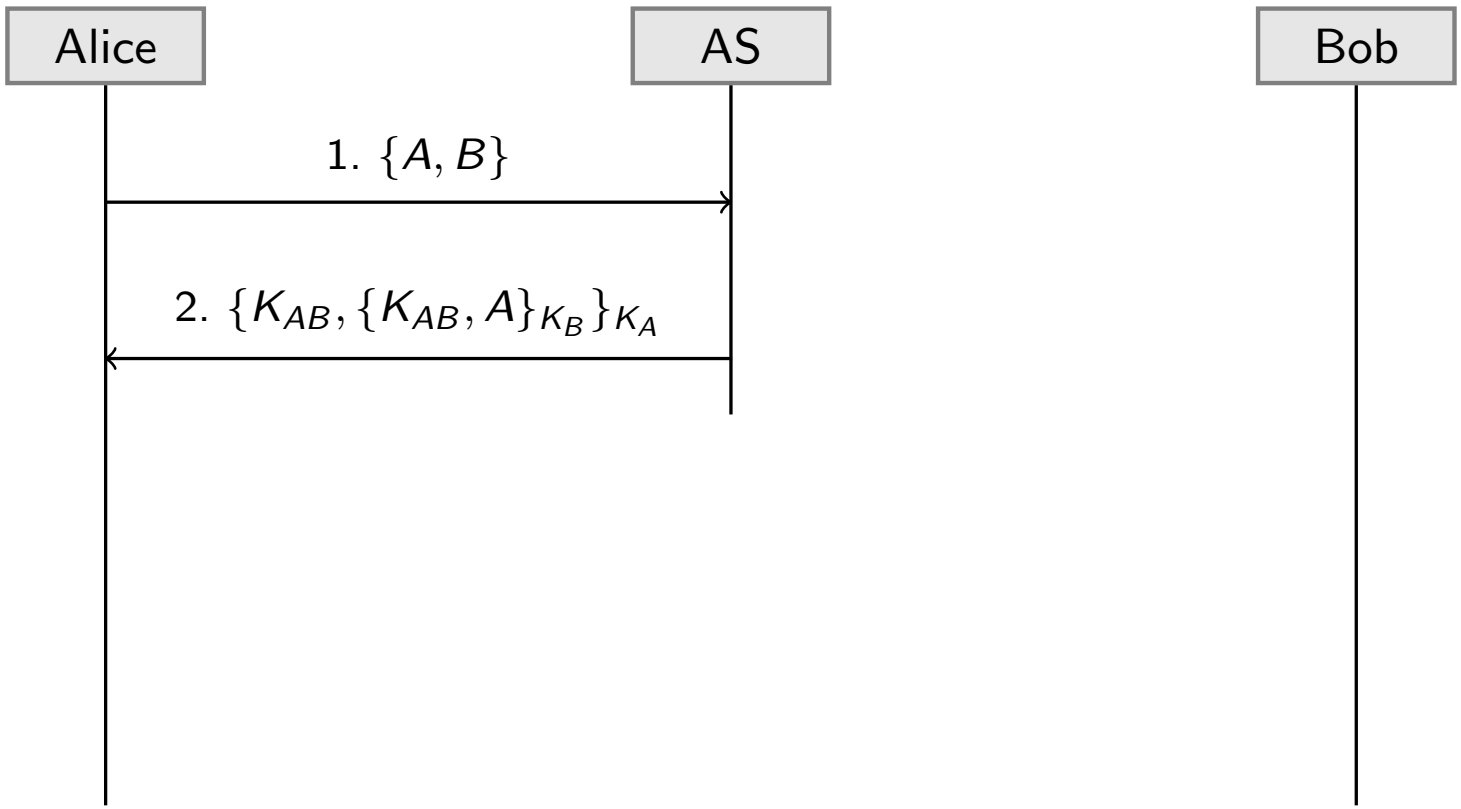
Naive Variant of the Symmetric NSP



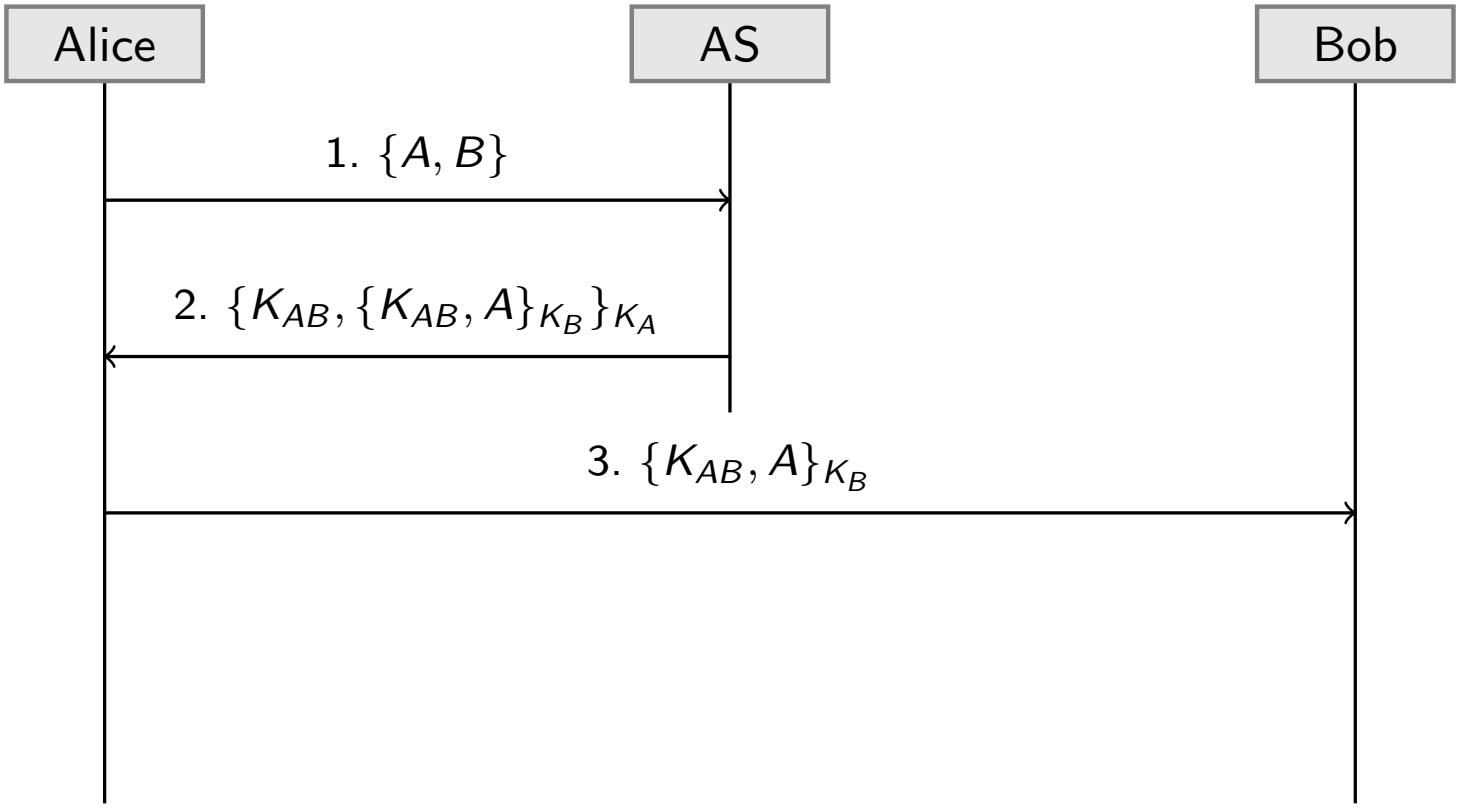
Naive Variant of the Symmetric NSP



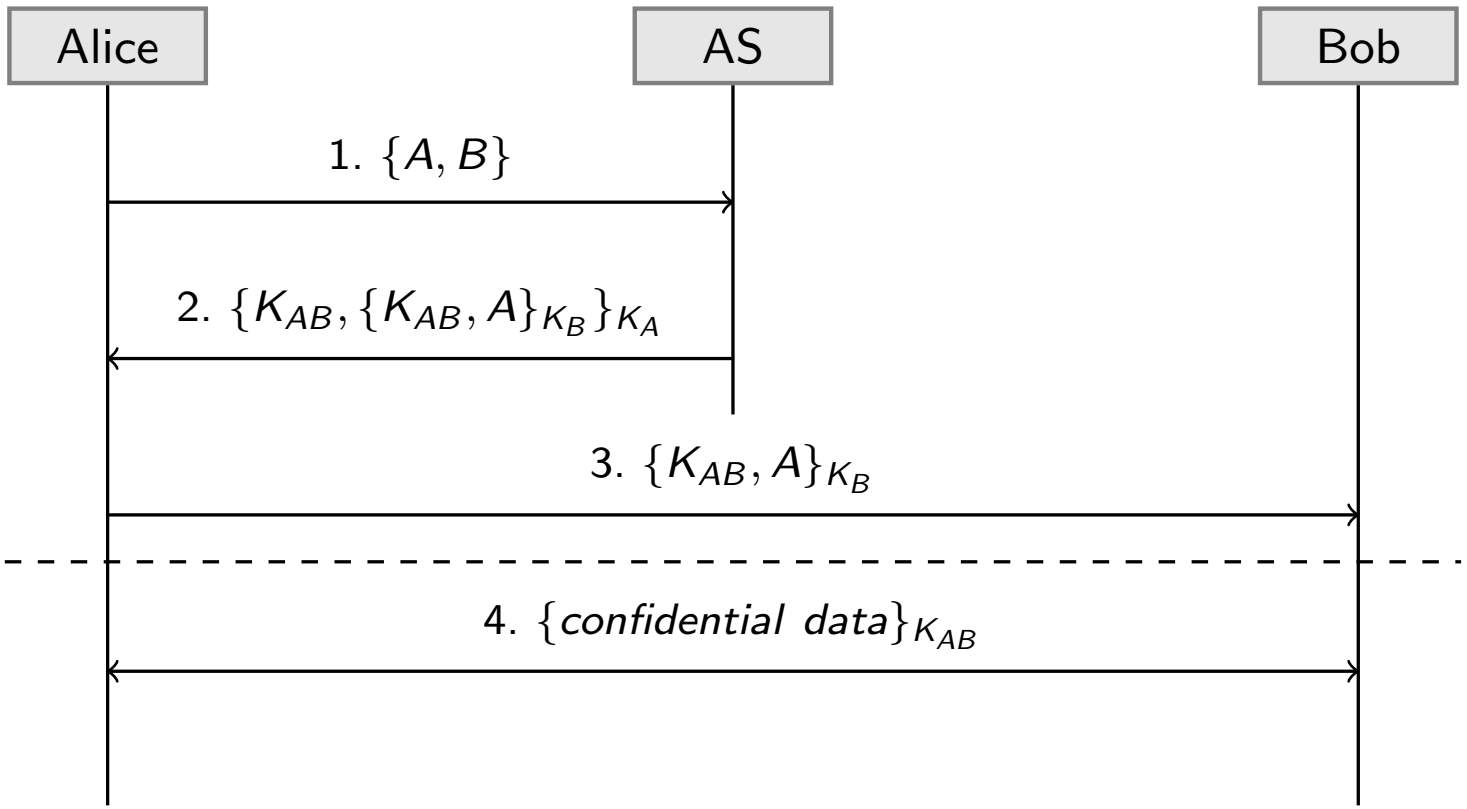
Naive Variant of the Symmetric NSP



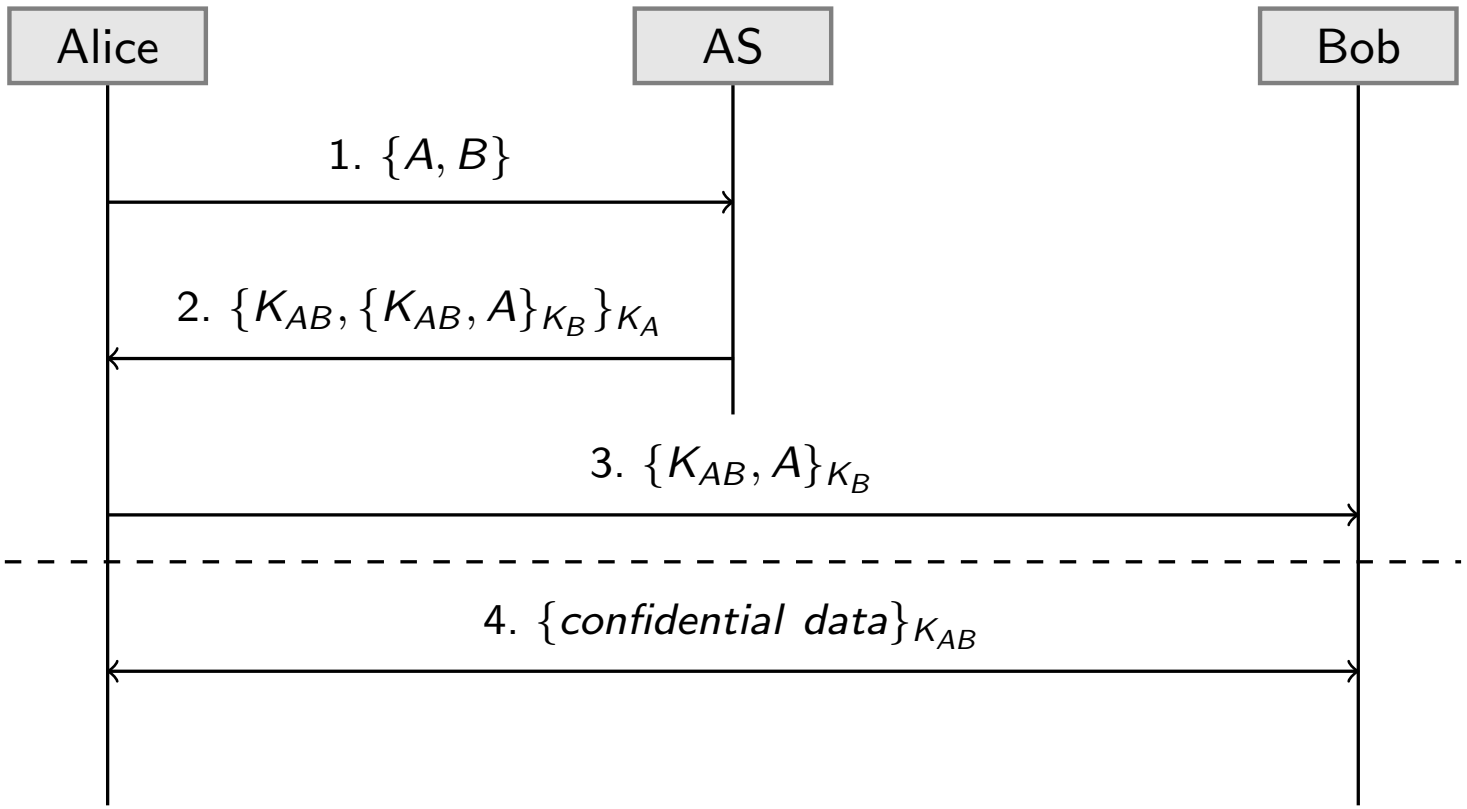
Naive Variant of the Symmetric NSP



Naive Variant of the Symmetric NSP



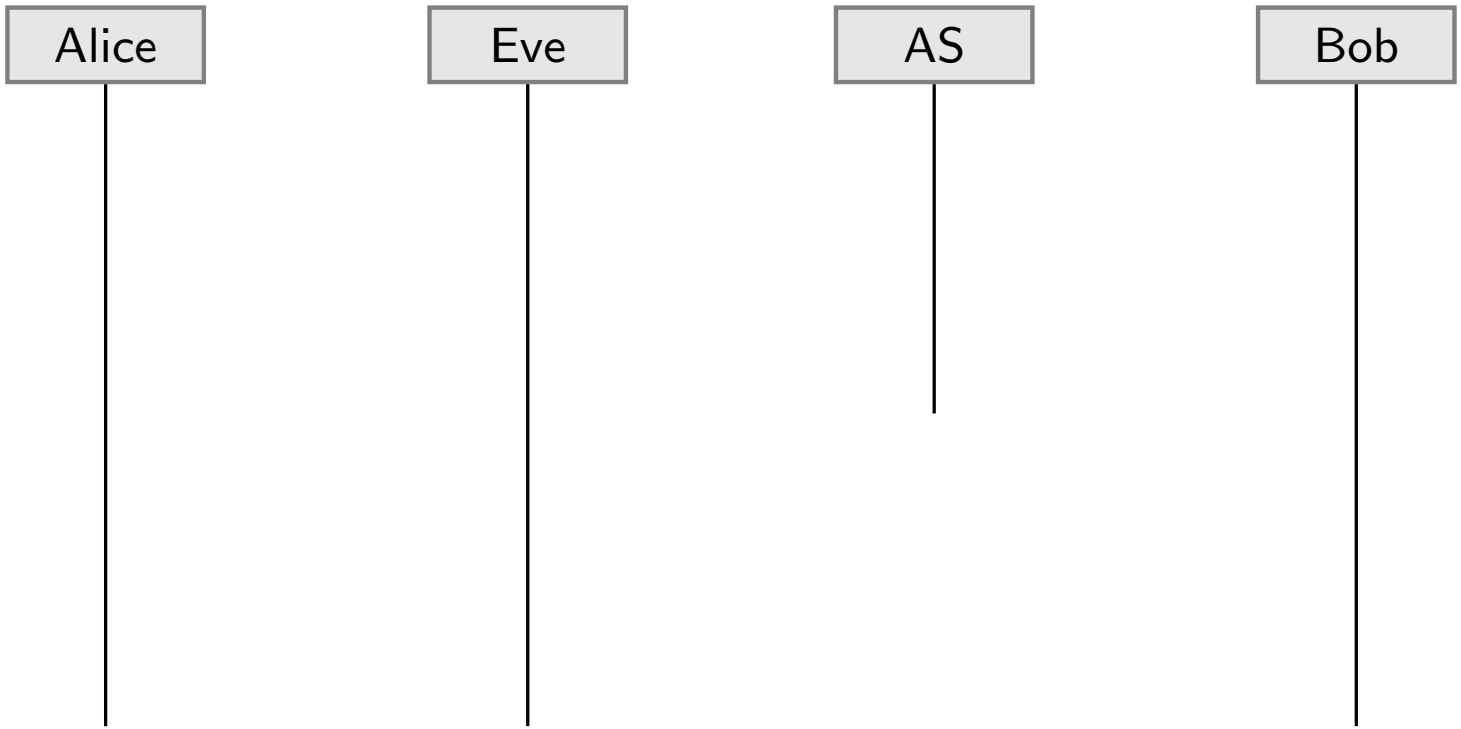
Naive Variant of the Symmetric NSP



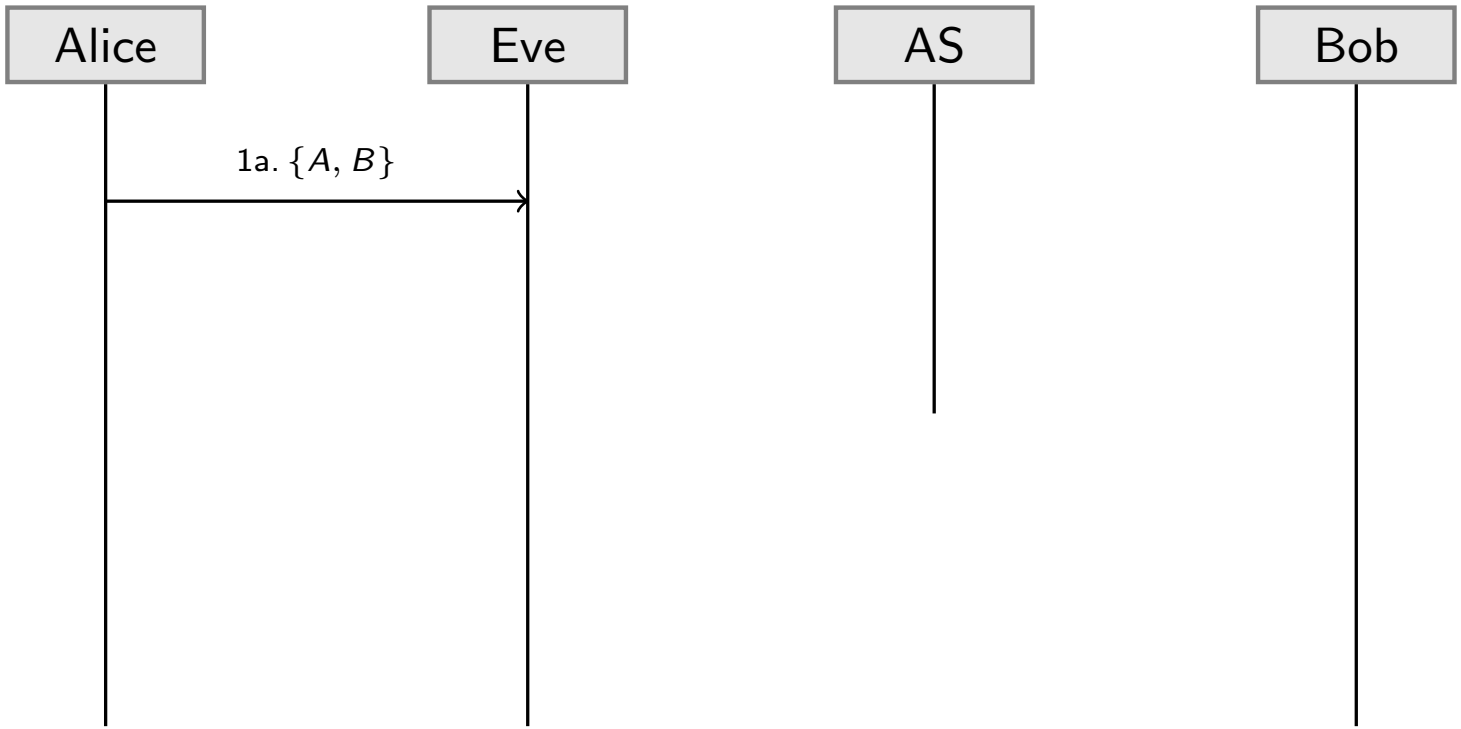
The naive variant of the NSP is not secure! Why?

Specify the steps of the traditional Needham-Schroeder protocol (symmetric variant)? Why is this protocol vulnerable?

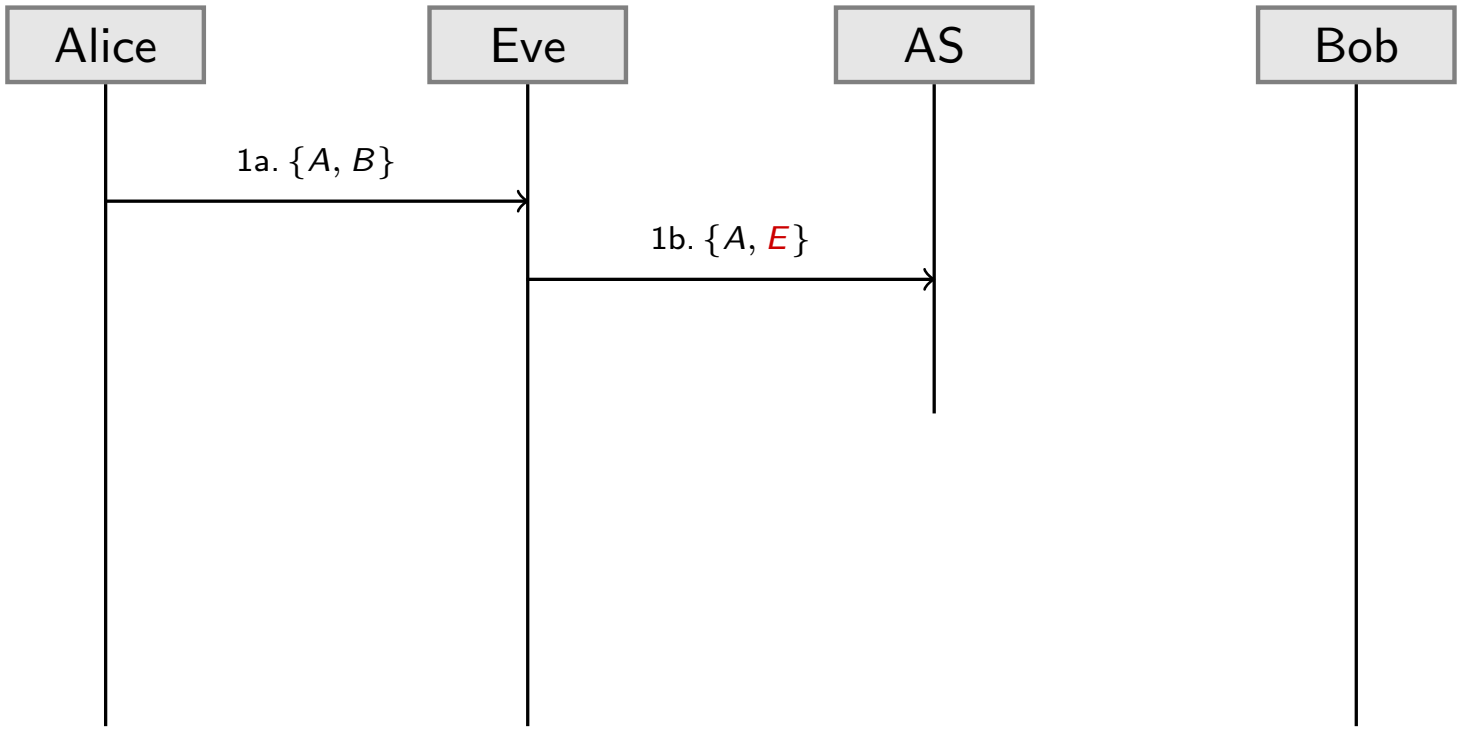
Attack for the naive Symmetric NSP



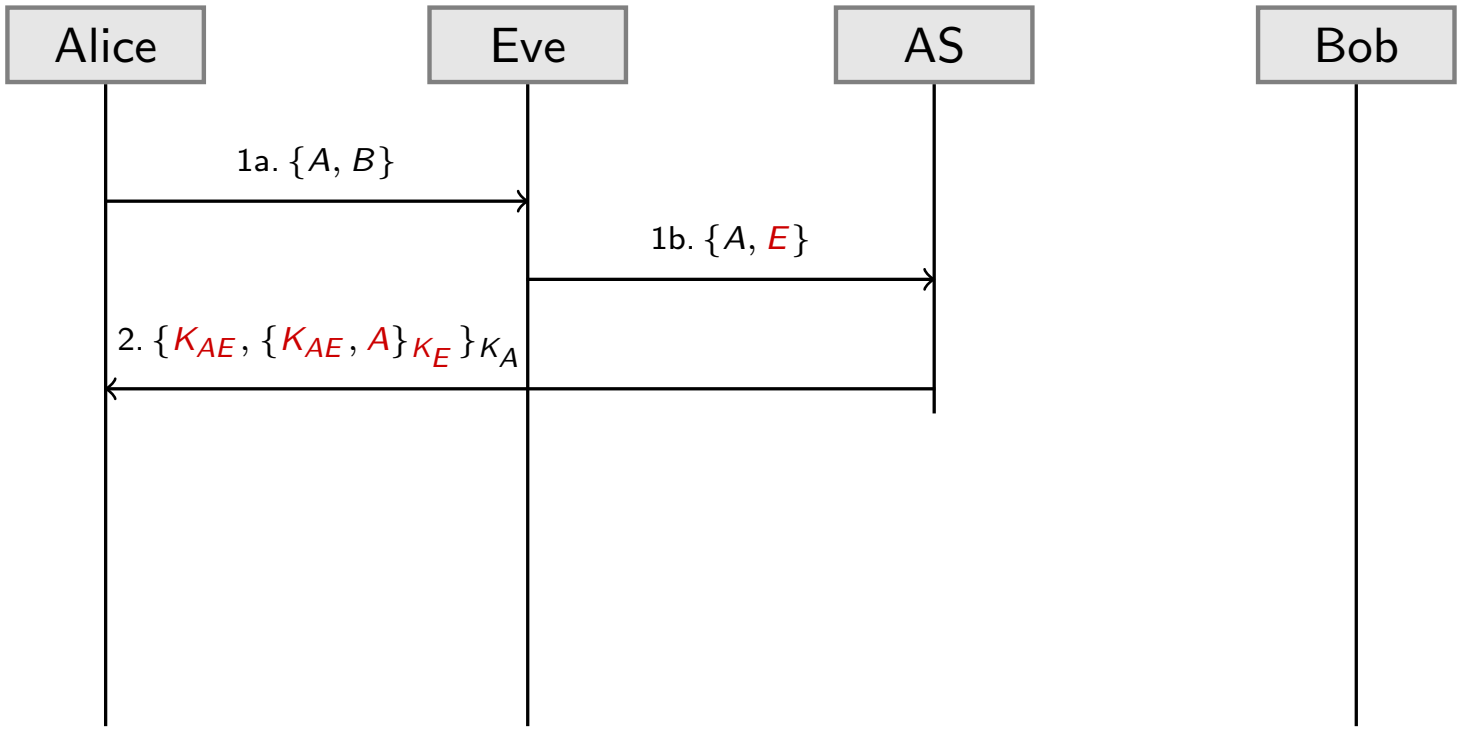
Attack for the naive Symmetric NSP



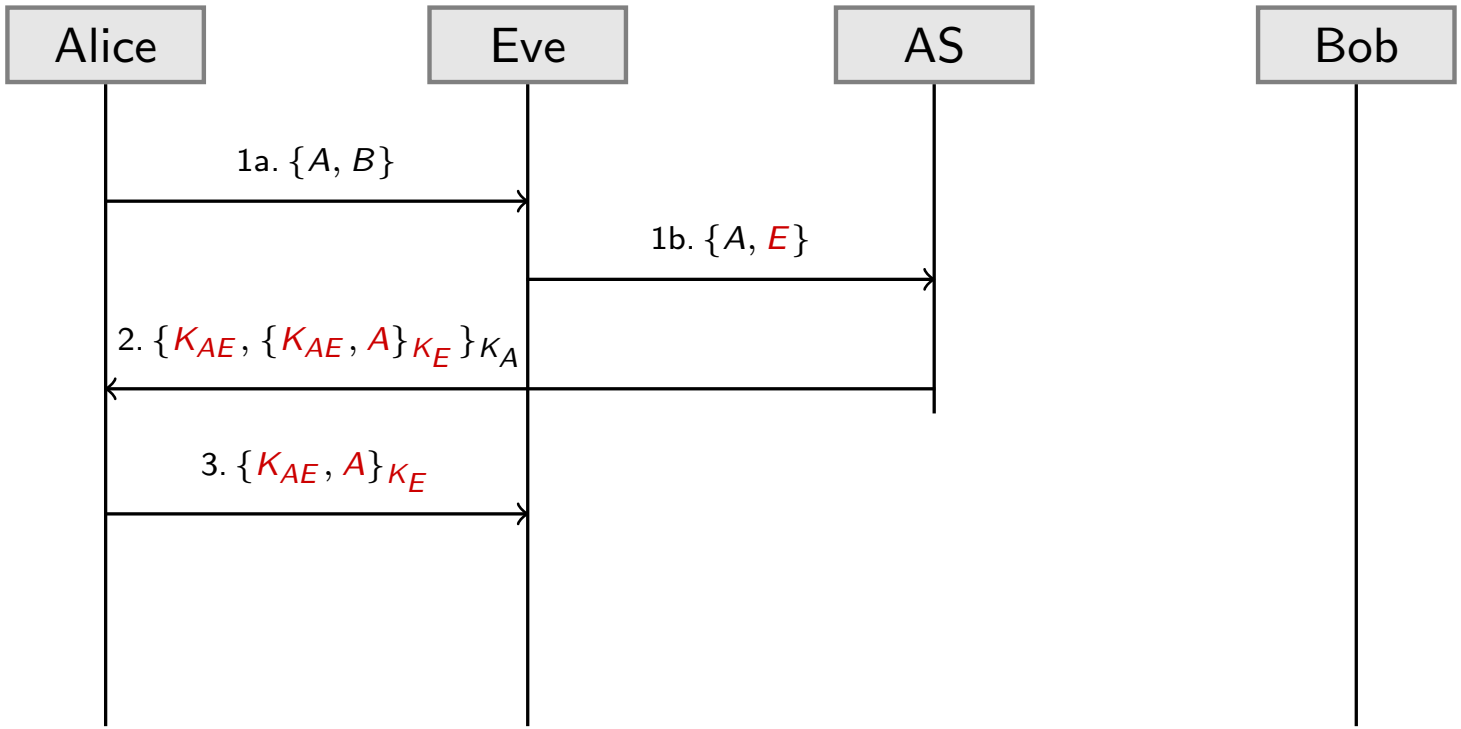
Attack for the naive Symmetric NSP



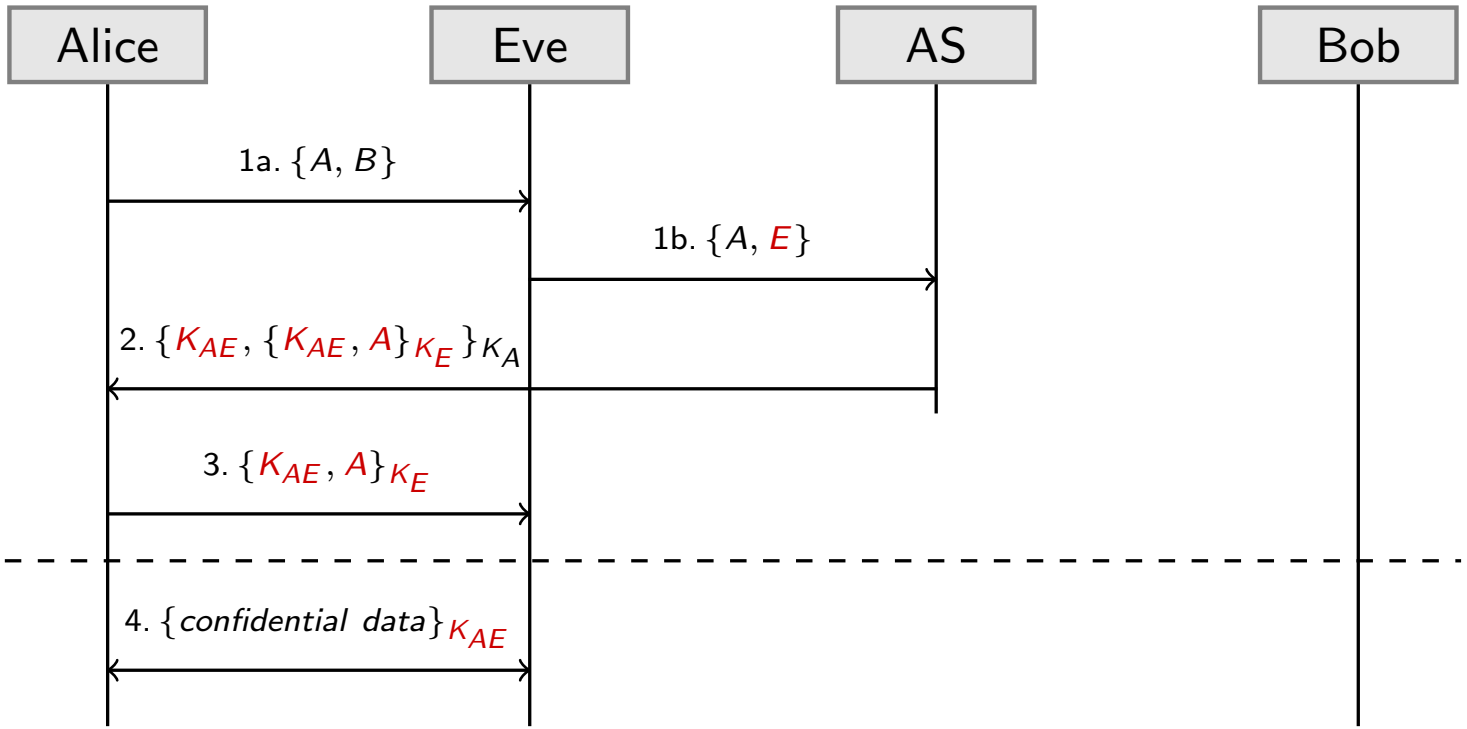
Attack for the naive Symmetric NSP



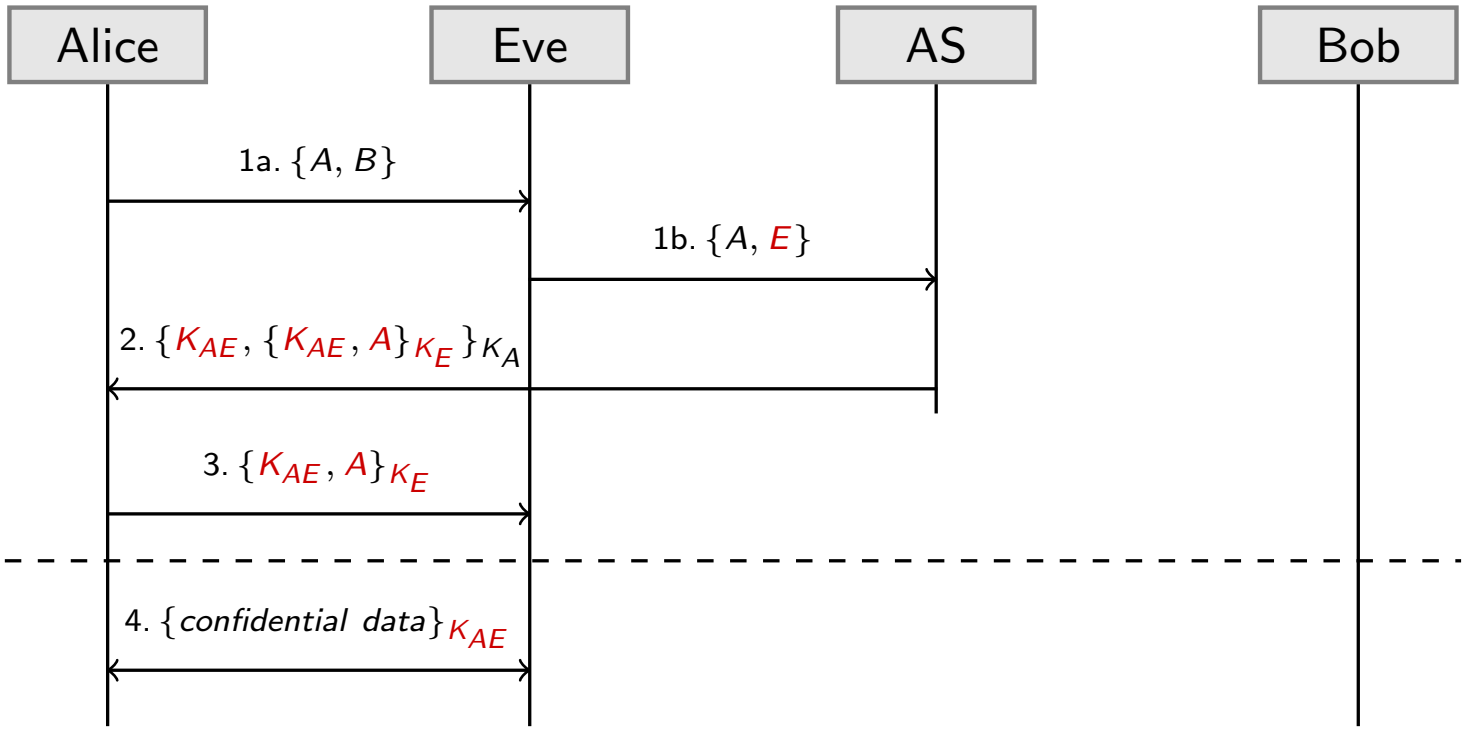
Attack for the naive Symmetric NSP



Attack for the naive Symmetric NSP

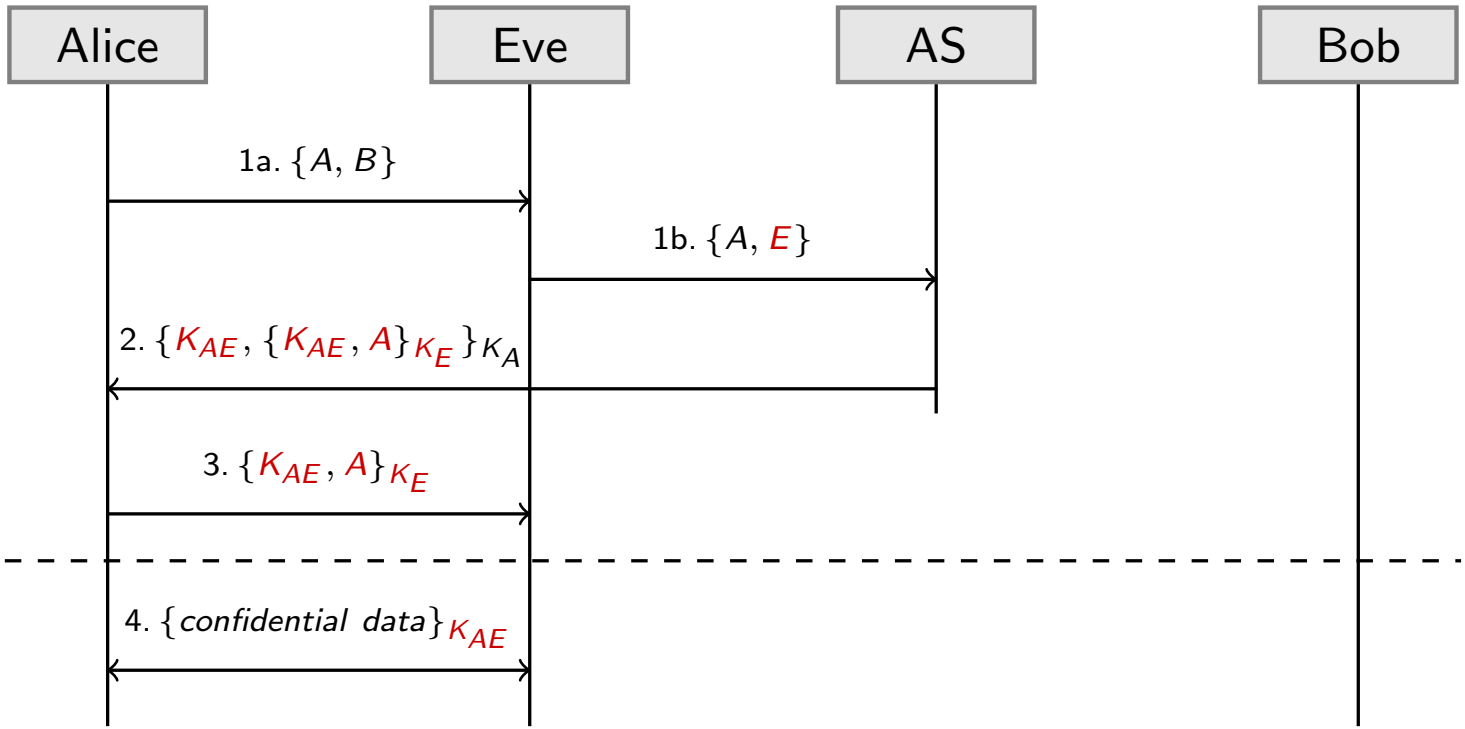


Attack for the naive Symmetric NSP



Eve is pretending to Alice to be Bob! Countermeasures?

Attack for the naive Symmetric NSP

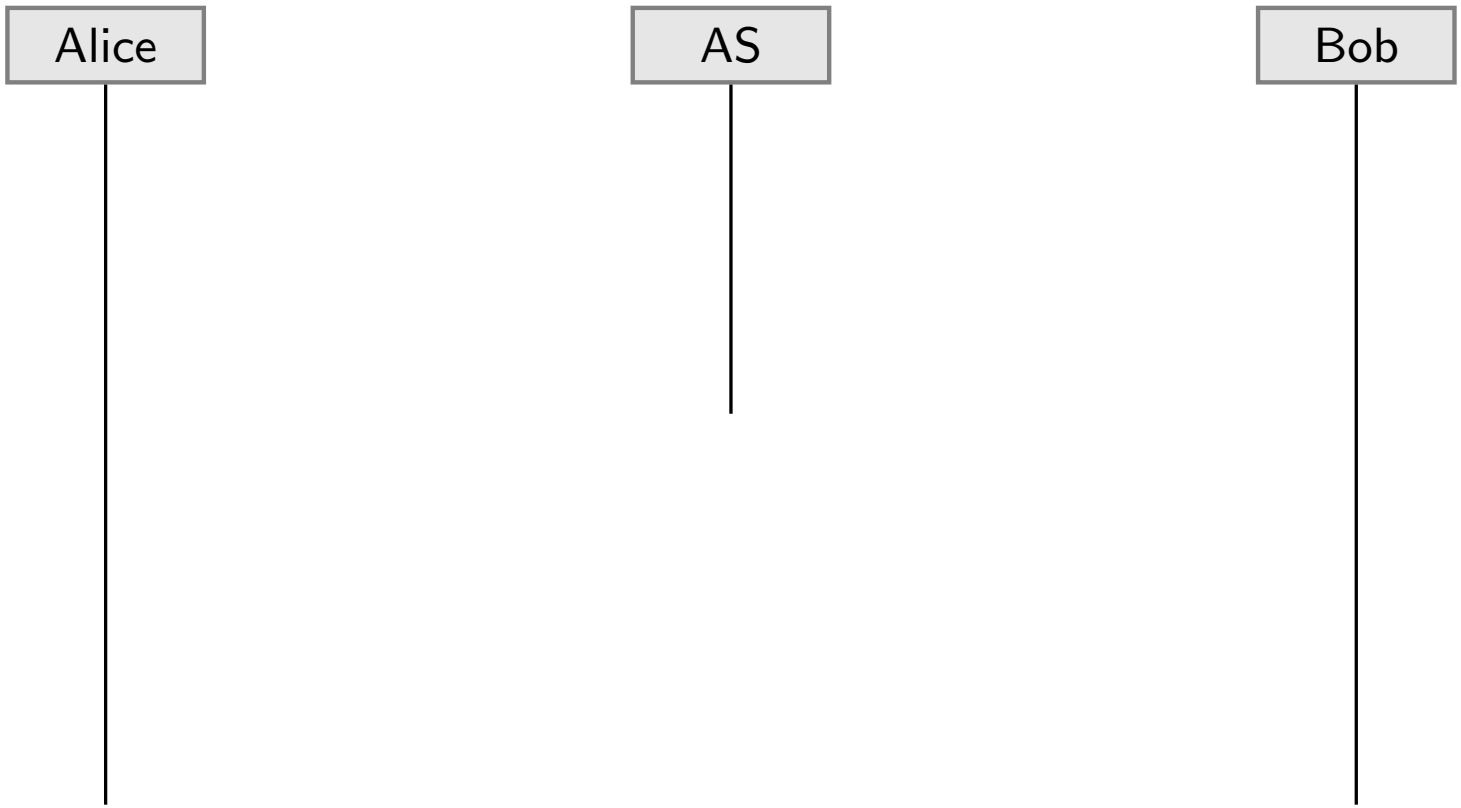


Eve is pretending to Alice to be Bob! Countermeasures?

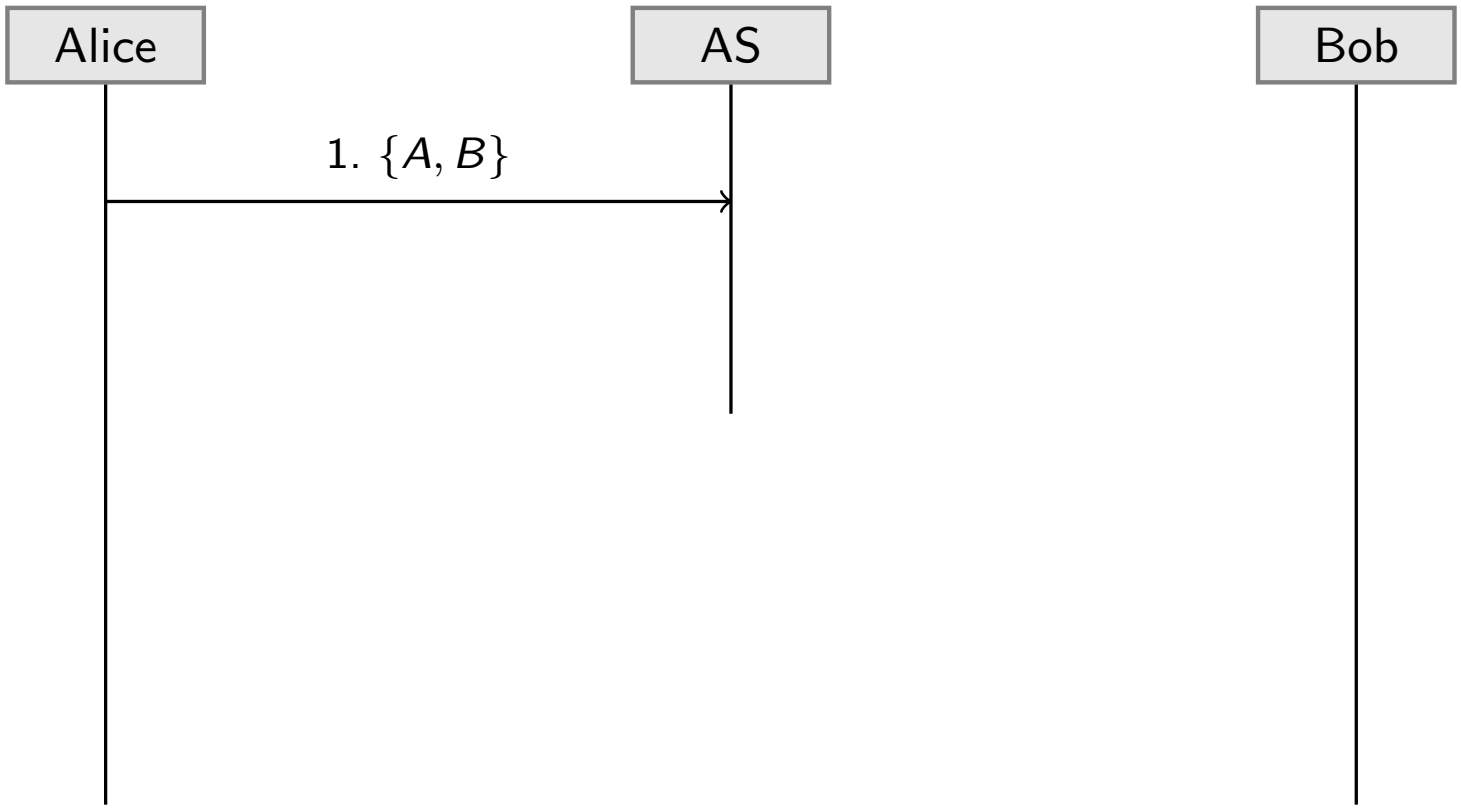
➜ Man-in-the-middle attacks can be prevented by sending identities inside the tickets!

Specify a man-in-the-middle and a replay attack for the NSP example

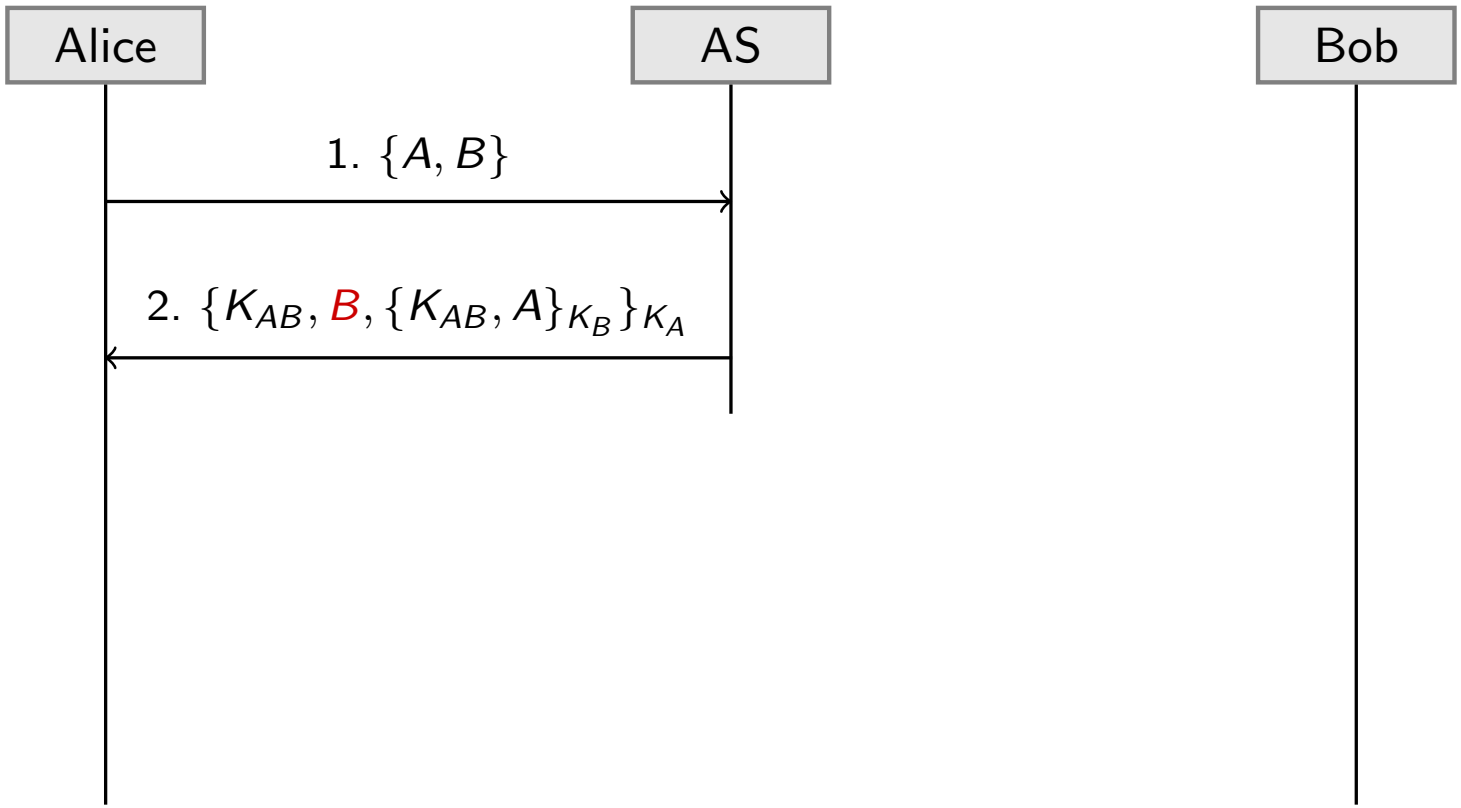
Variant 2 for the Symmetric NSP



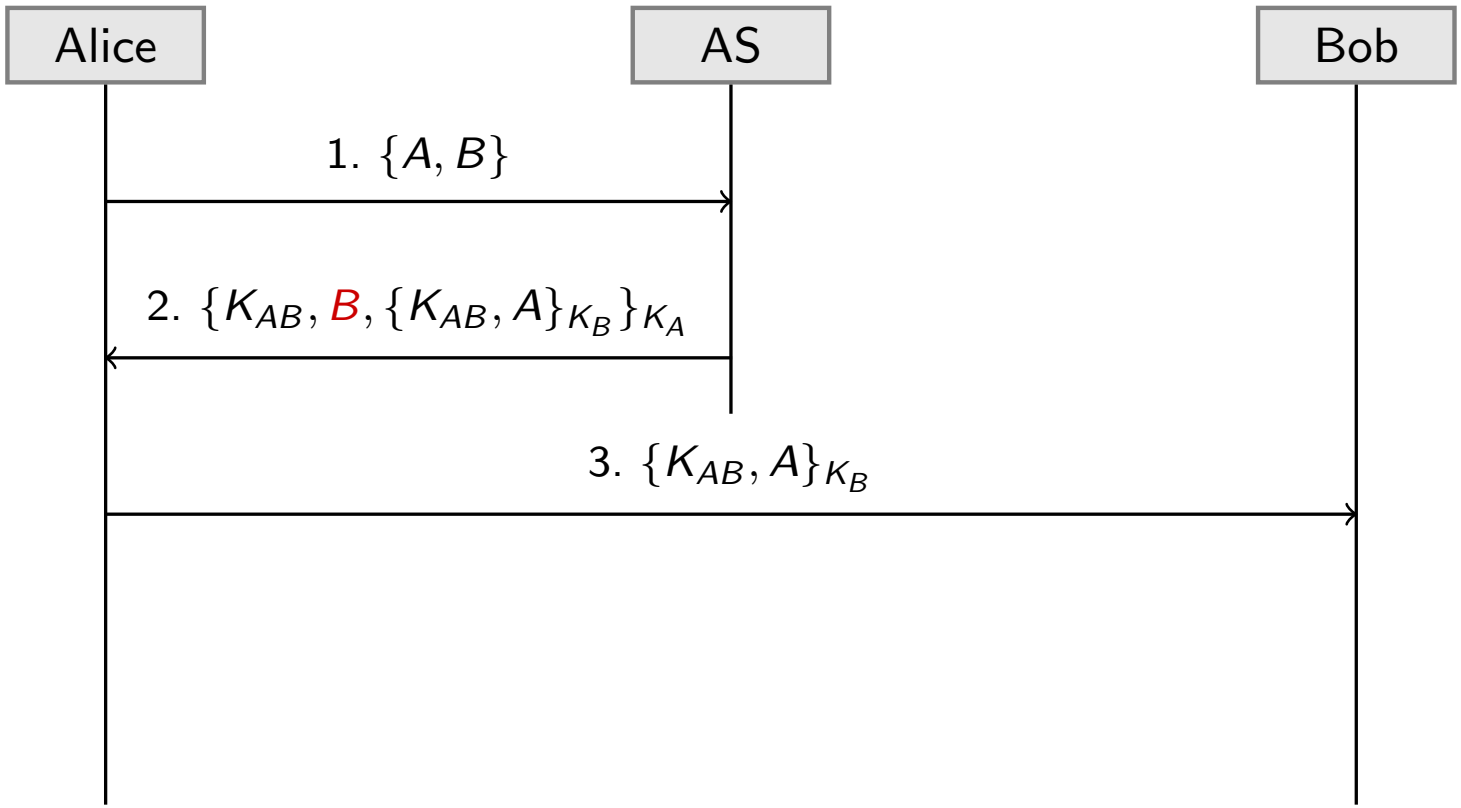
Variant 2 for the Symmetric NSP



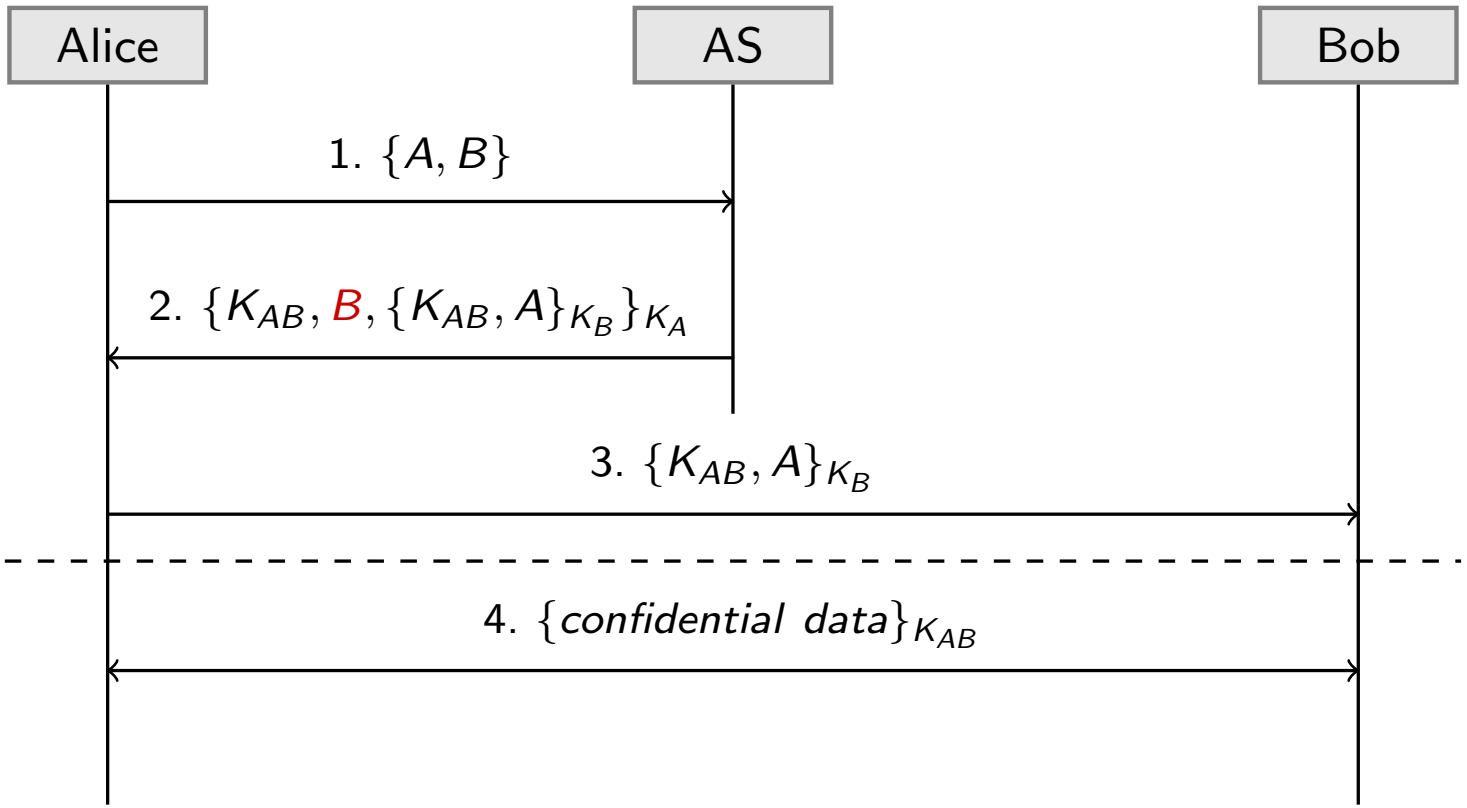
Variant 2 for the Symmetric NSP



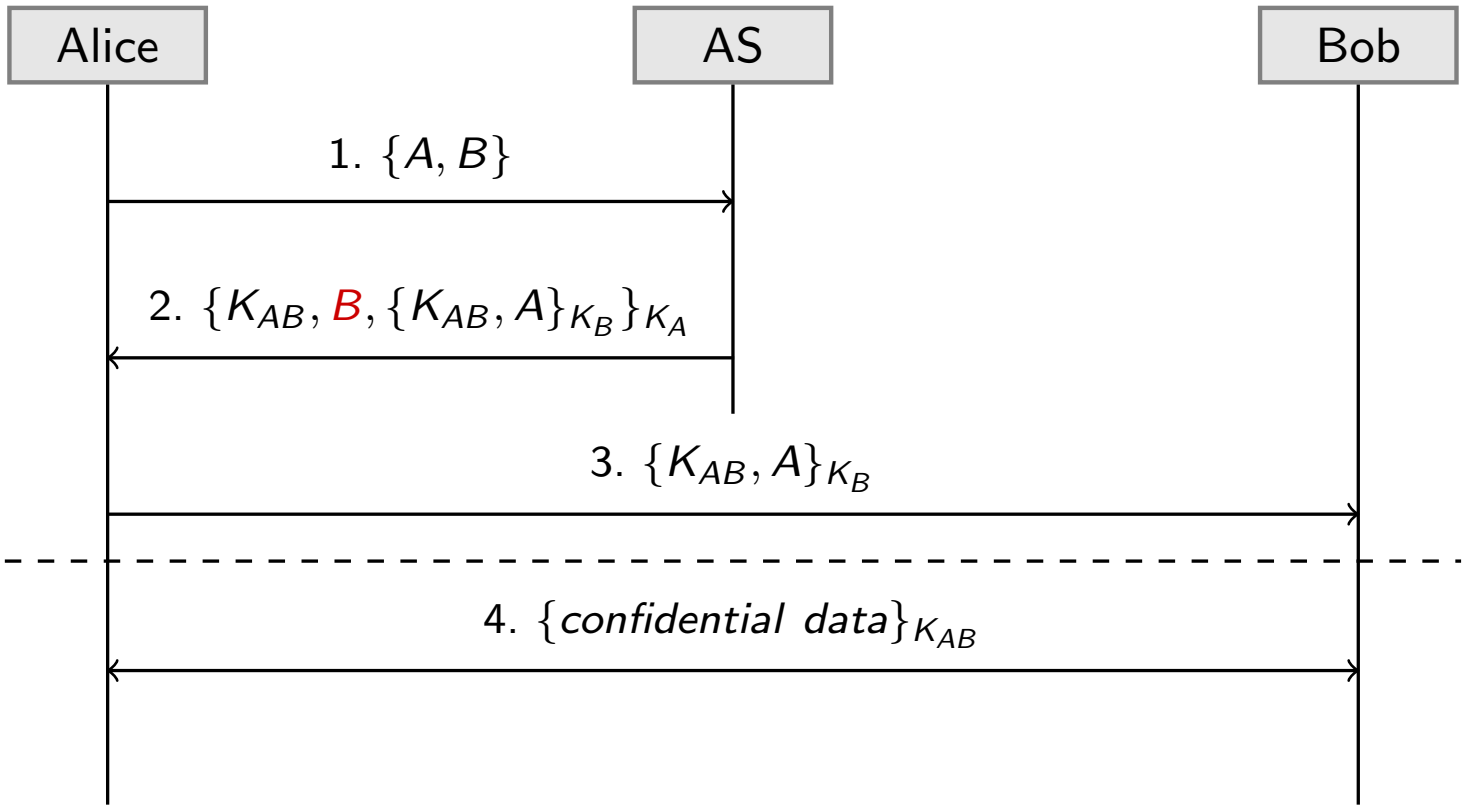
Variant 2 for the Symmetric NSP



Variant 2 for the Symmetric NSP

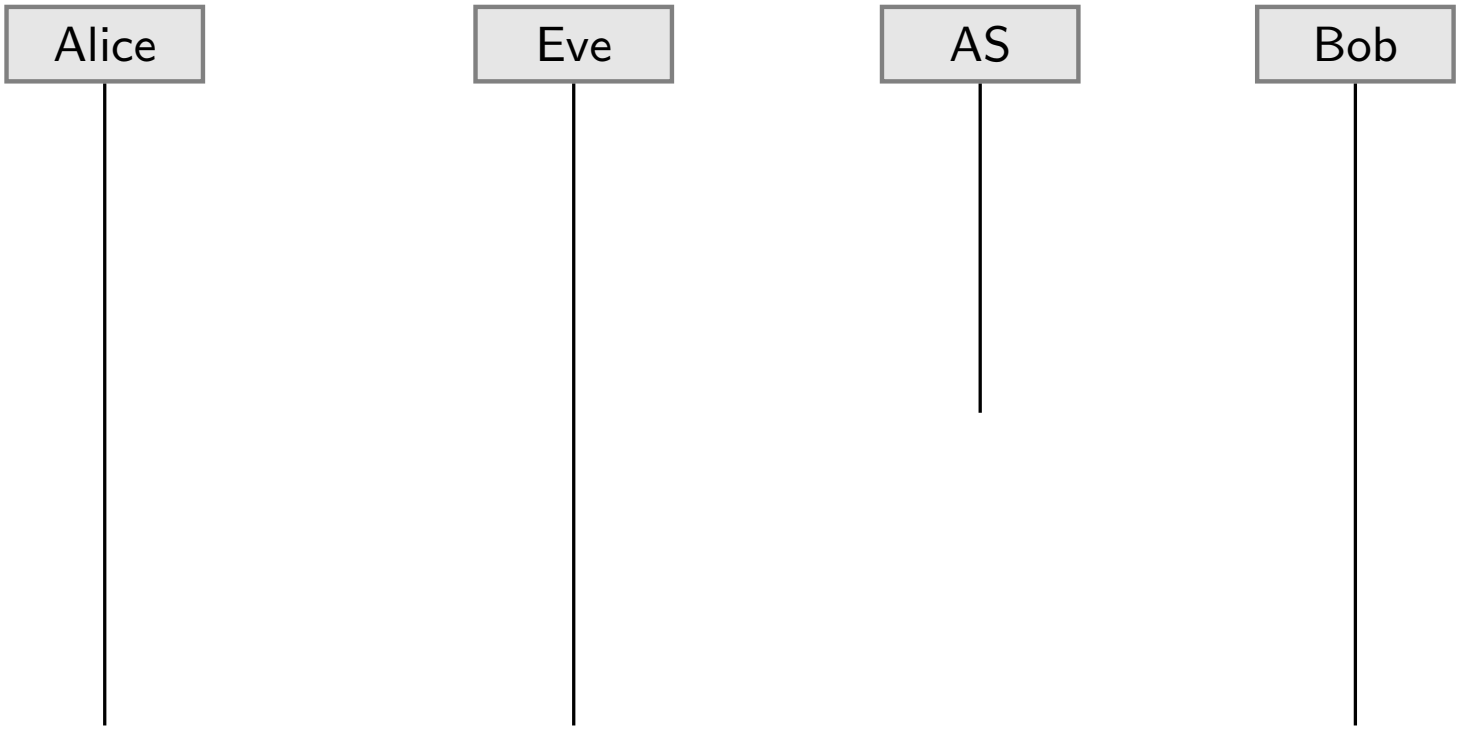


Variant 2 for the Symmetric NSP

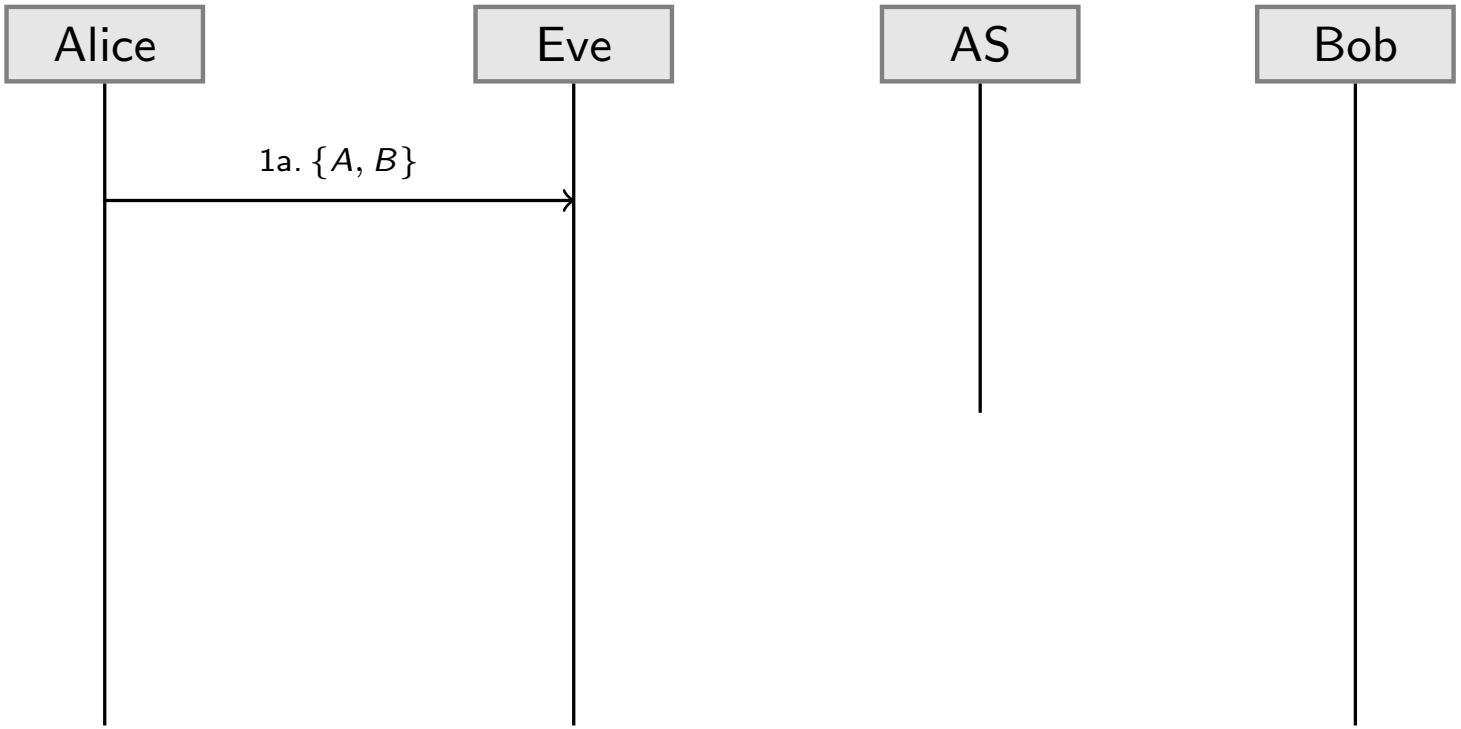


By specifying Bob's identity in step 2, Alice is able to detect the Man-in-the-middle attack!

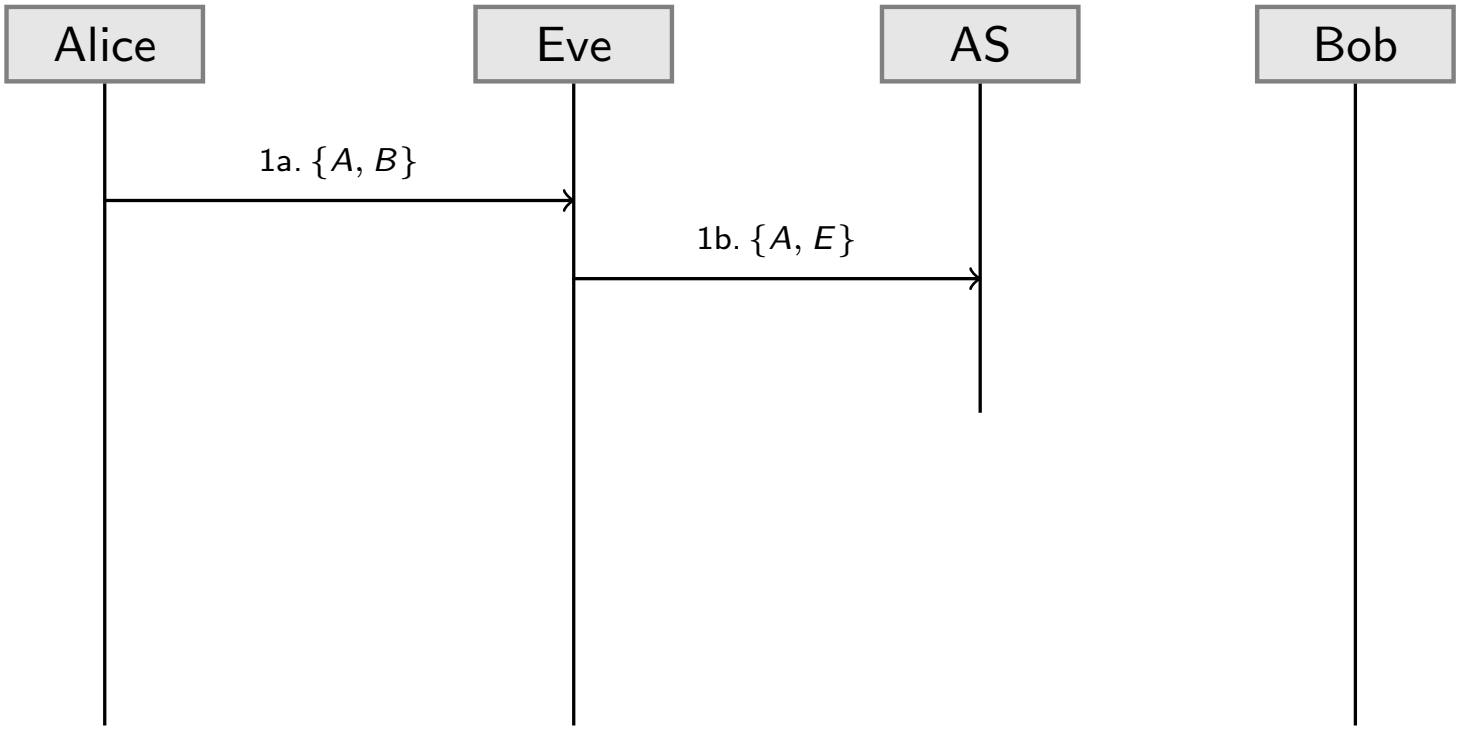
Detecting a Man-in-the-middle Attack



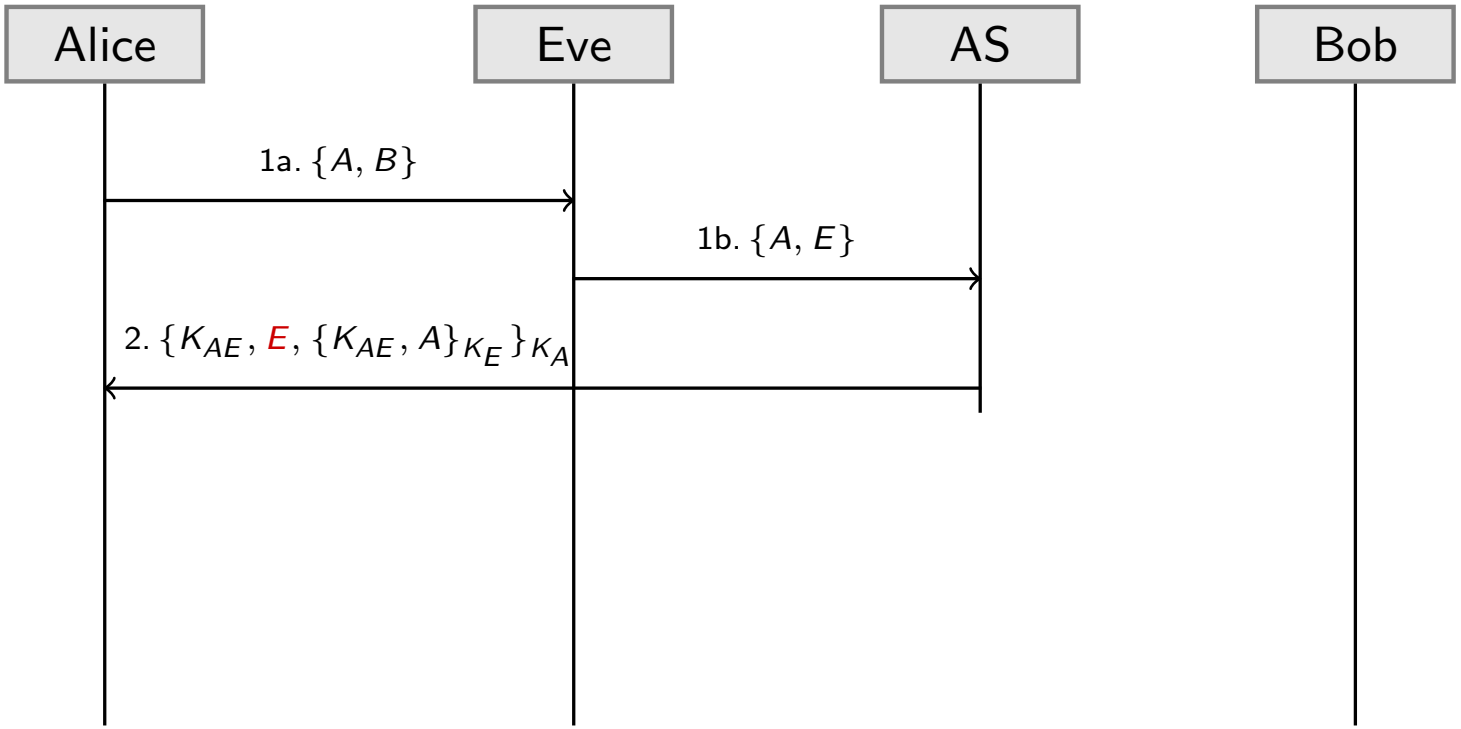
Detecting a Man-in-the-middle Attack



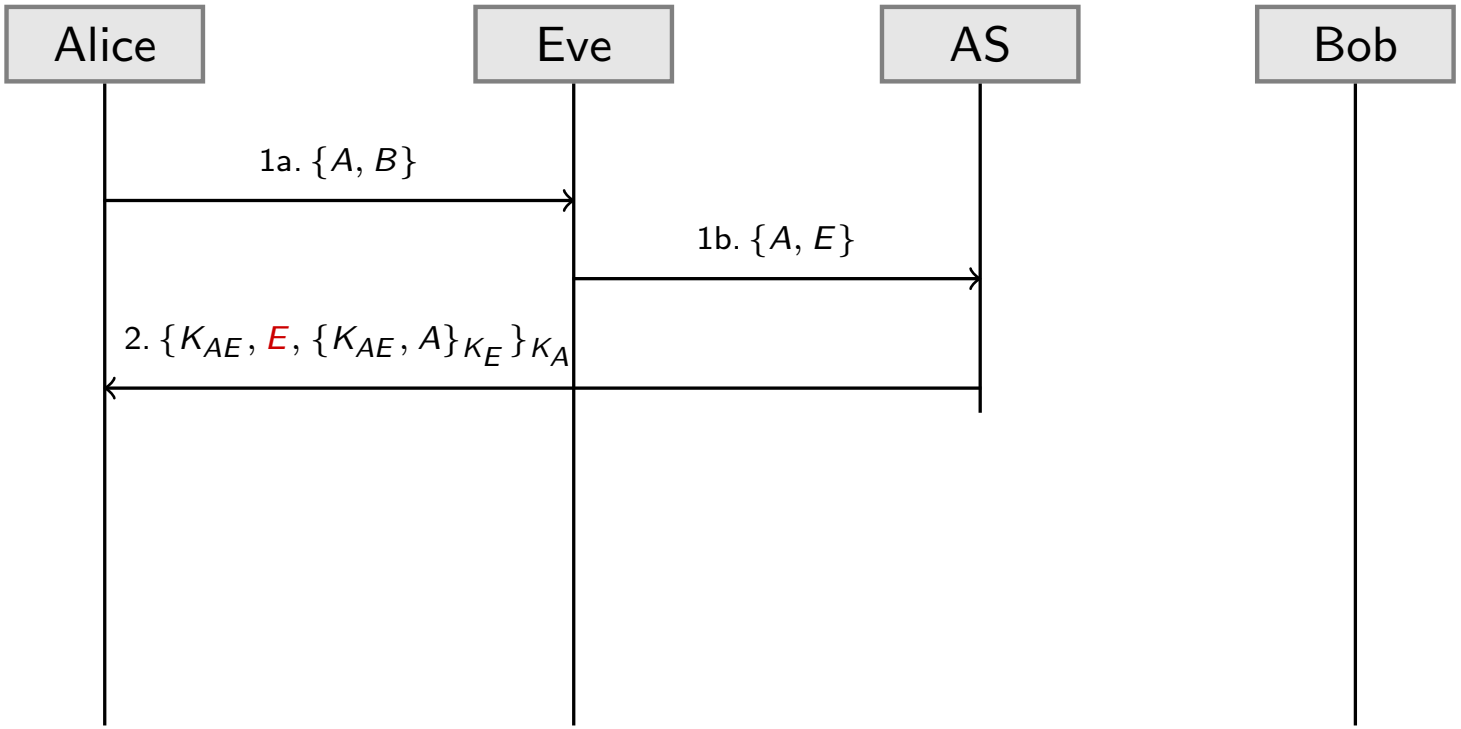
Detecting a Man-in-the-middle Attack



Detecting a Man-in-the-middle Attack

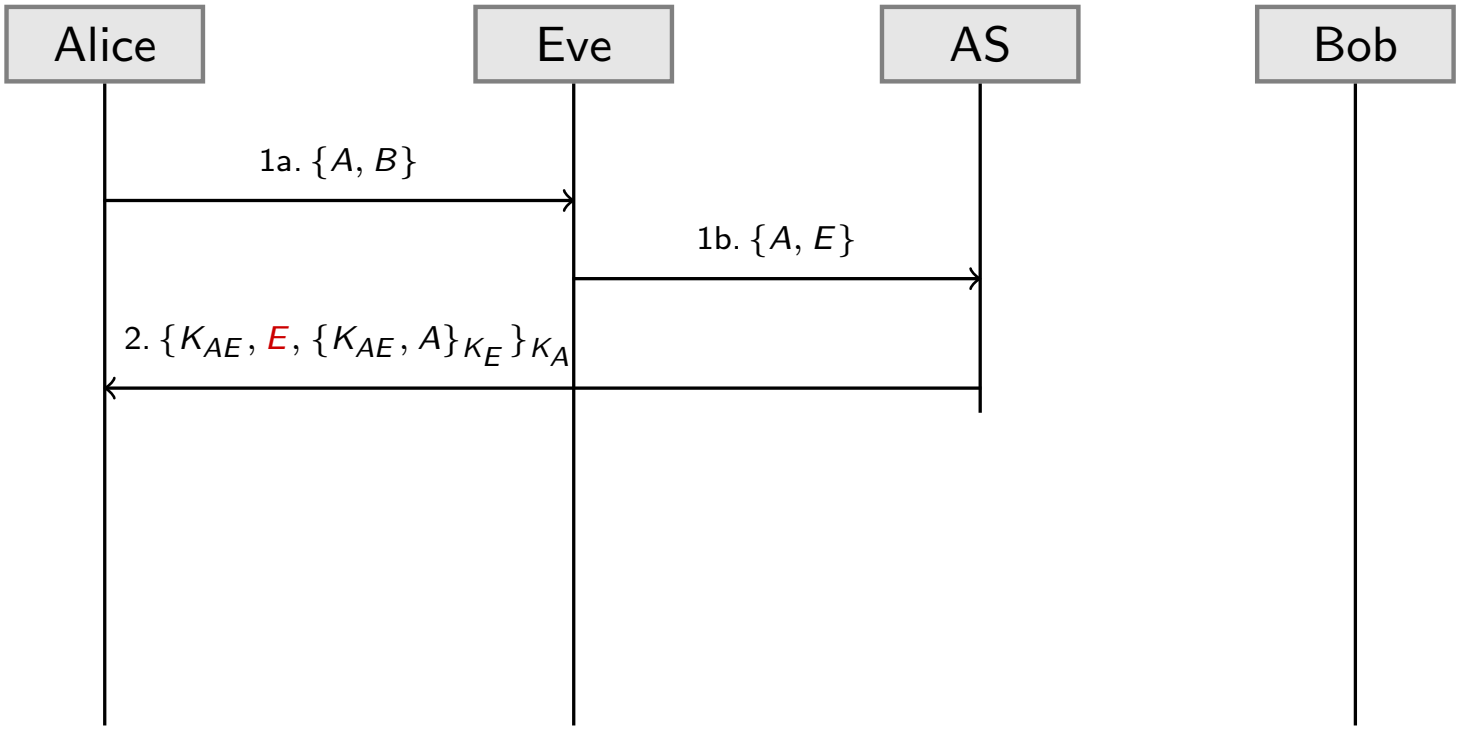


Detecting a Man-in-the-middle Attack



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

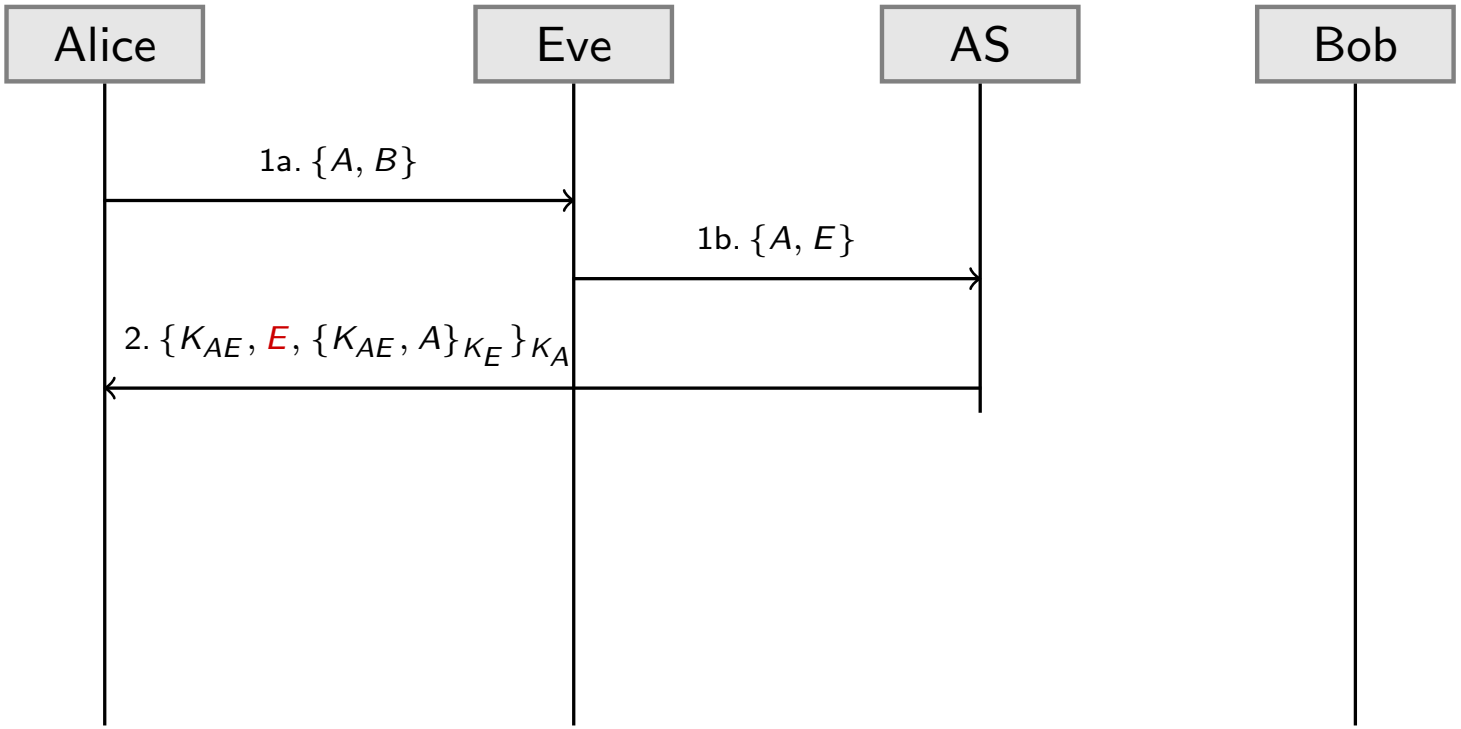
Detecting a Man-in-the-middle Attack



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

➜ Is the protocol secure now?

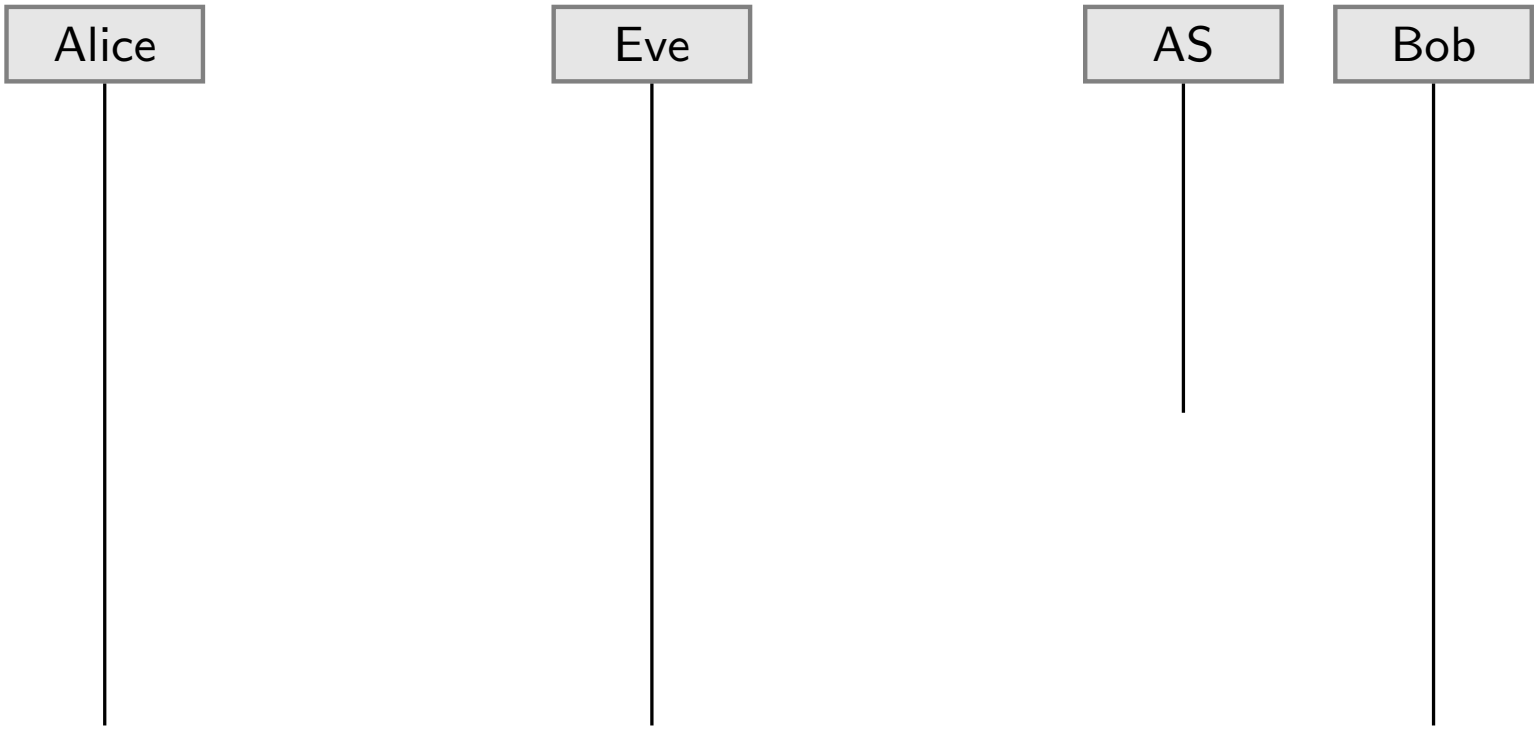
Detecting a Man-in-the-middle Attack



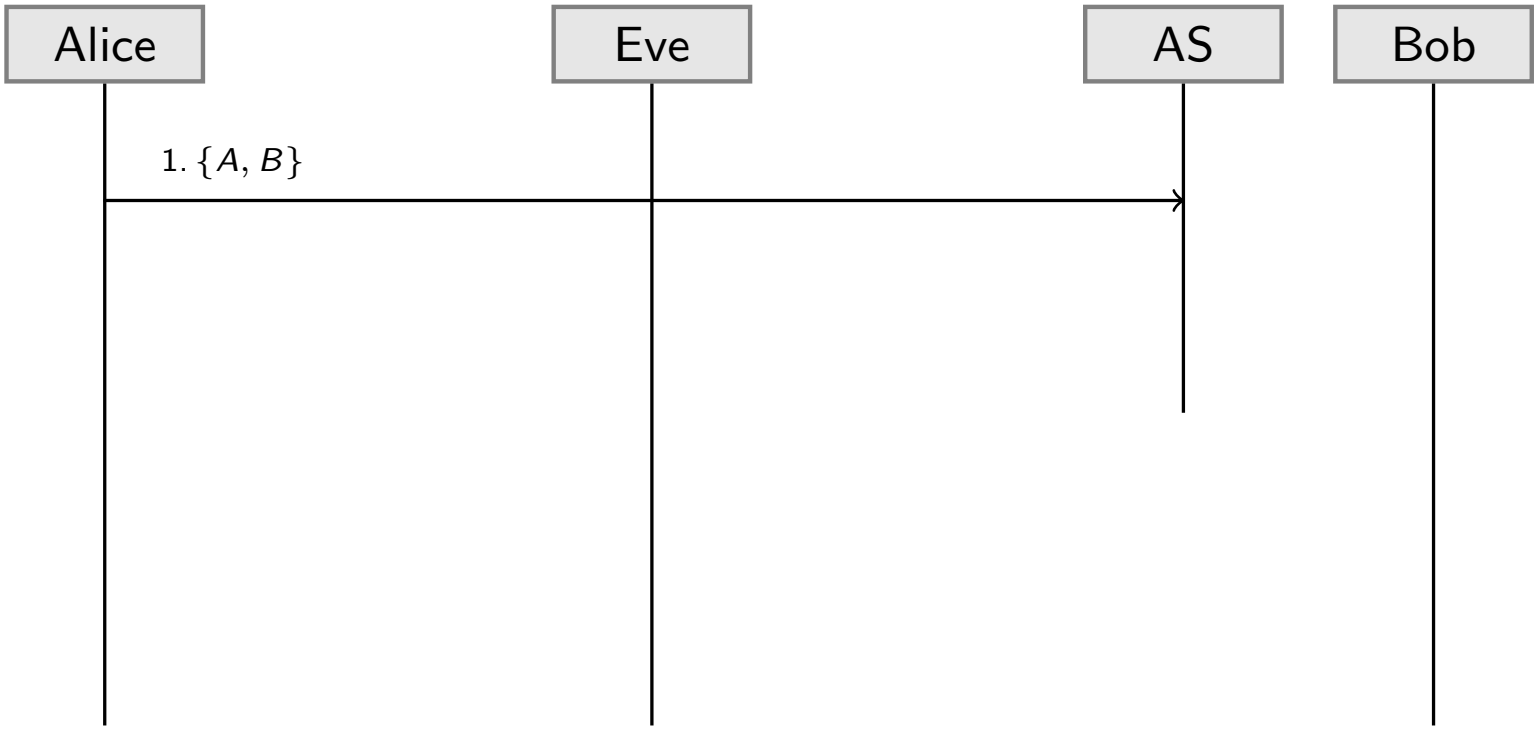
Alice detects in Step 2 that the ticket of the authentication server has been manipulated and cancels the session!

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

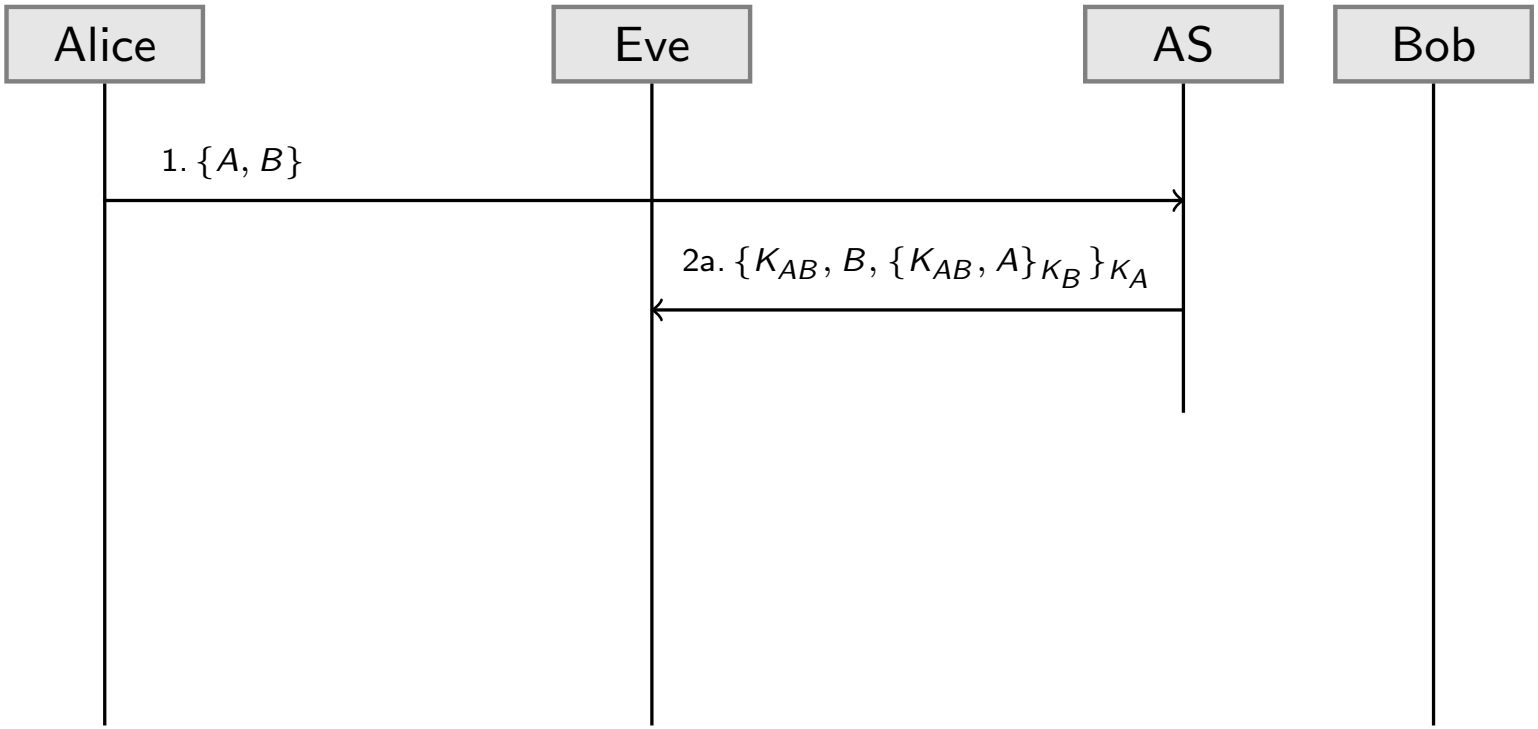
Attack for Variant 2 of the Symmetric NSP



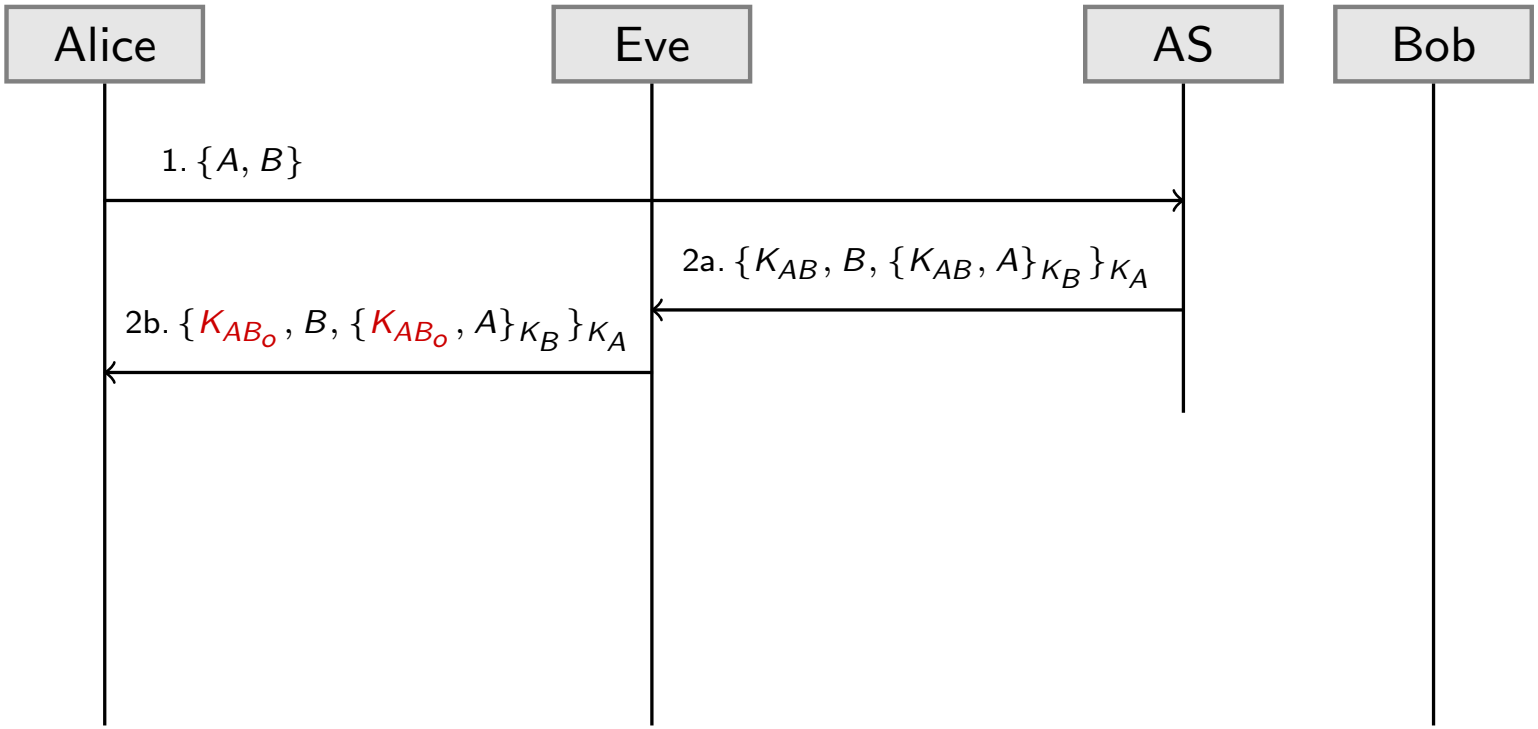
Attack for Variant 2 of the Symmetric NSP



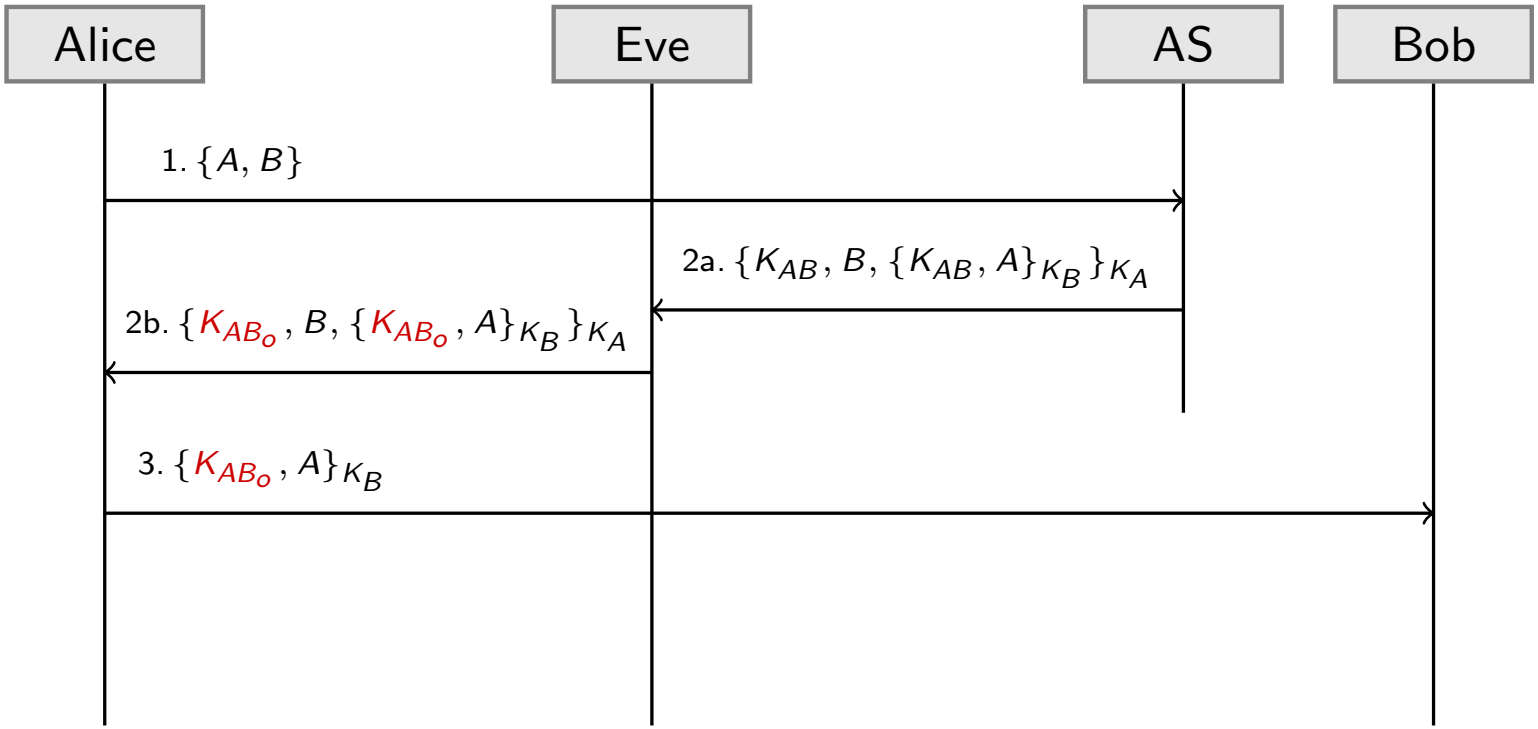
Attack for Variant 2 of the Symmetric NSP



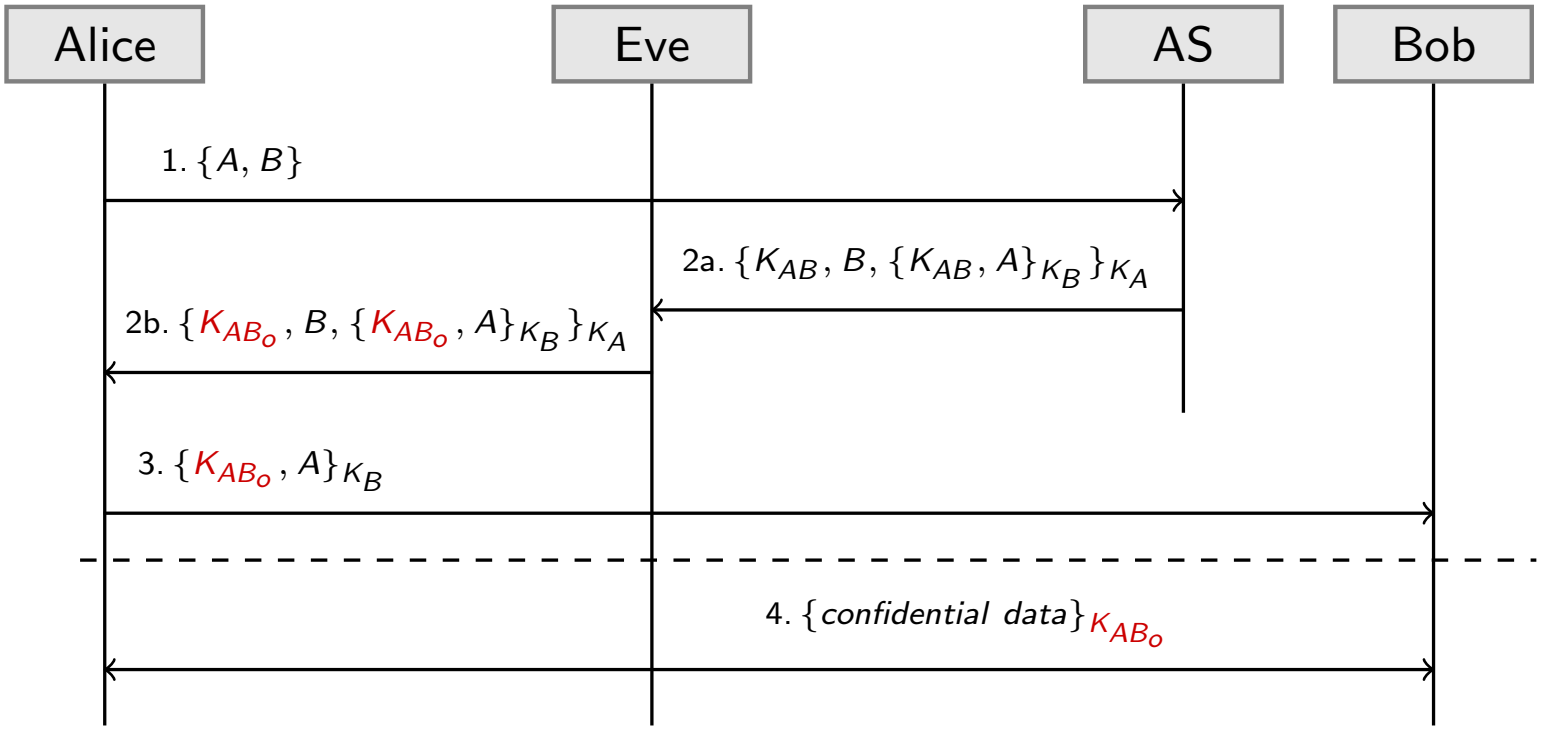
Attack for Variant 2 of the Symmetric NSP



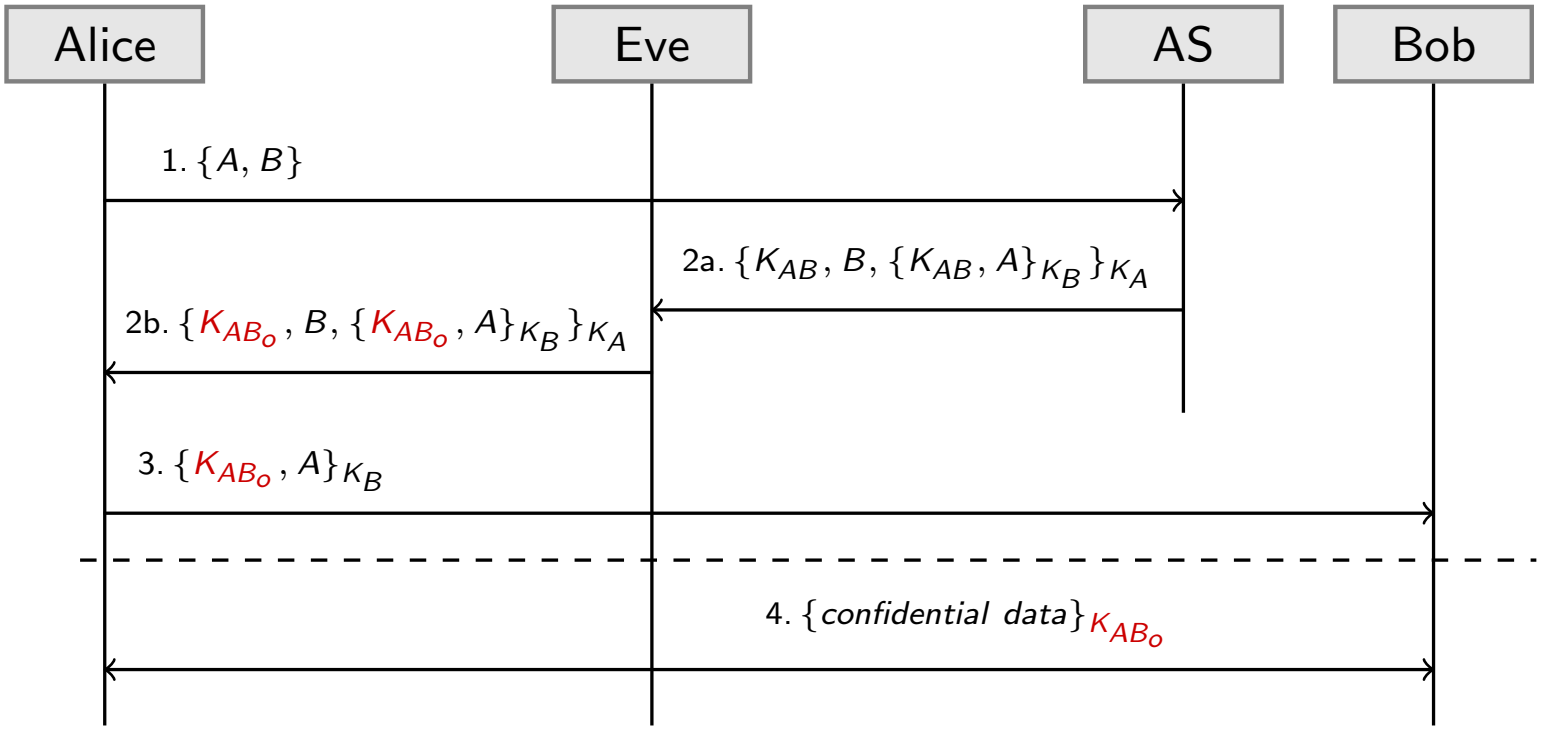
Attack for Variant 2 of the Symmetric NSP



Attack for Variant 2 of the Symmetric NSP

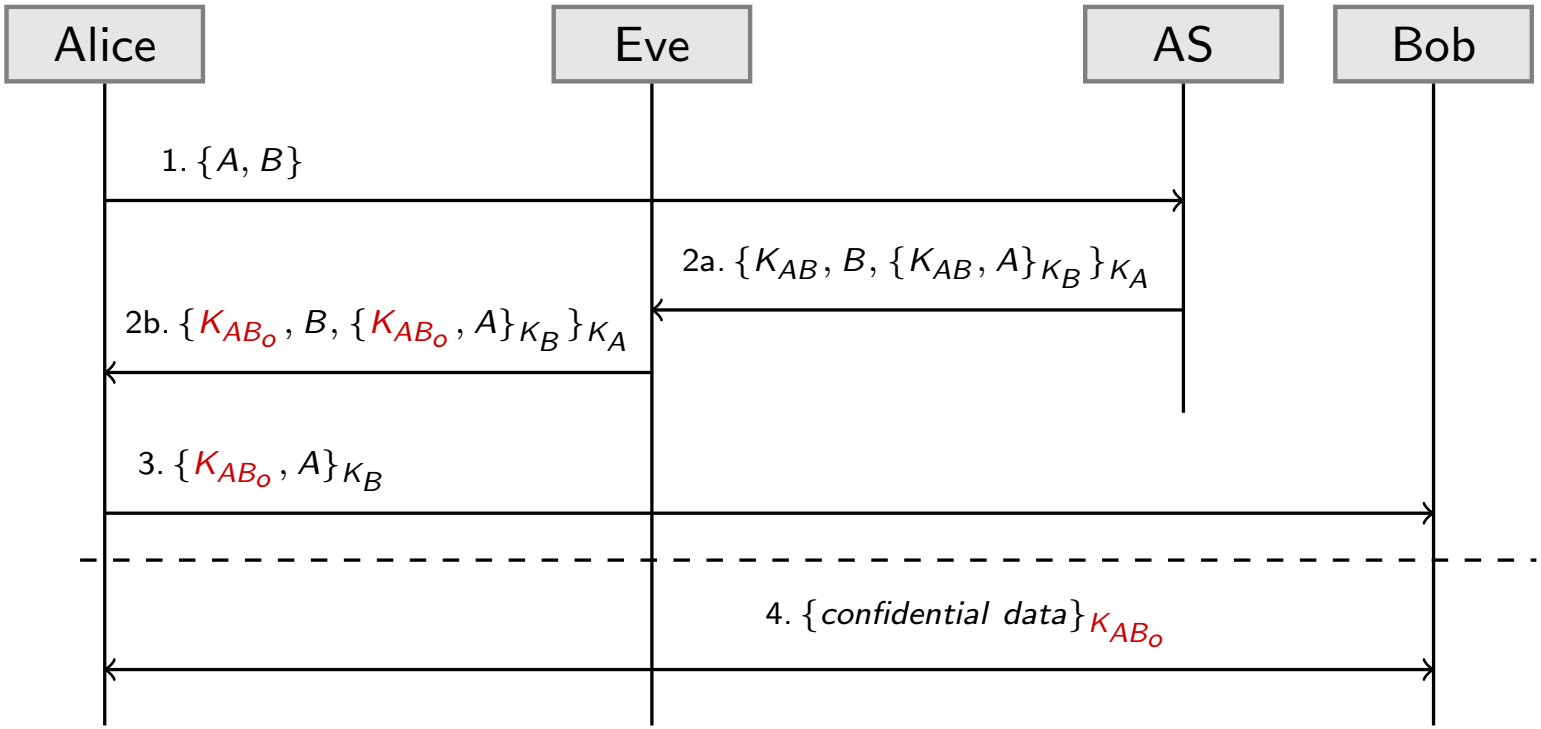


Attack for Variant 2 of the Symmetric NSP



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

Attack for Variant 2 of the Symmetric NSP

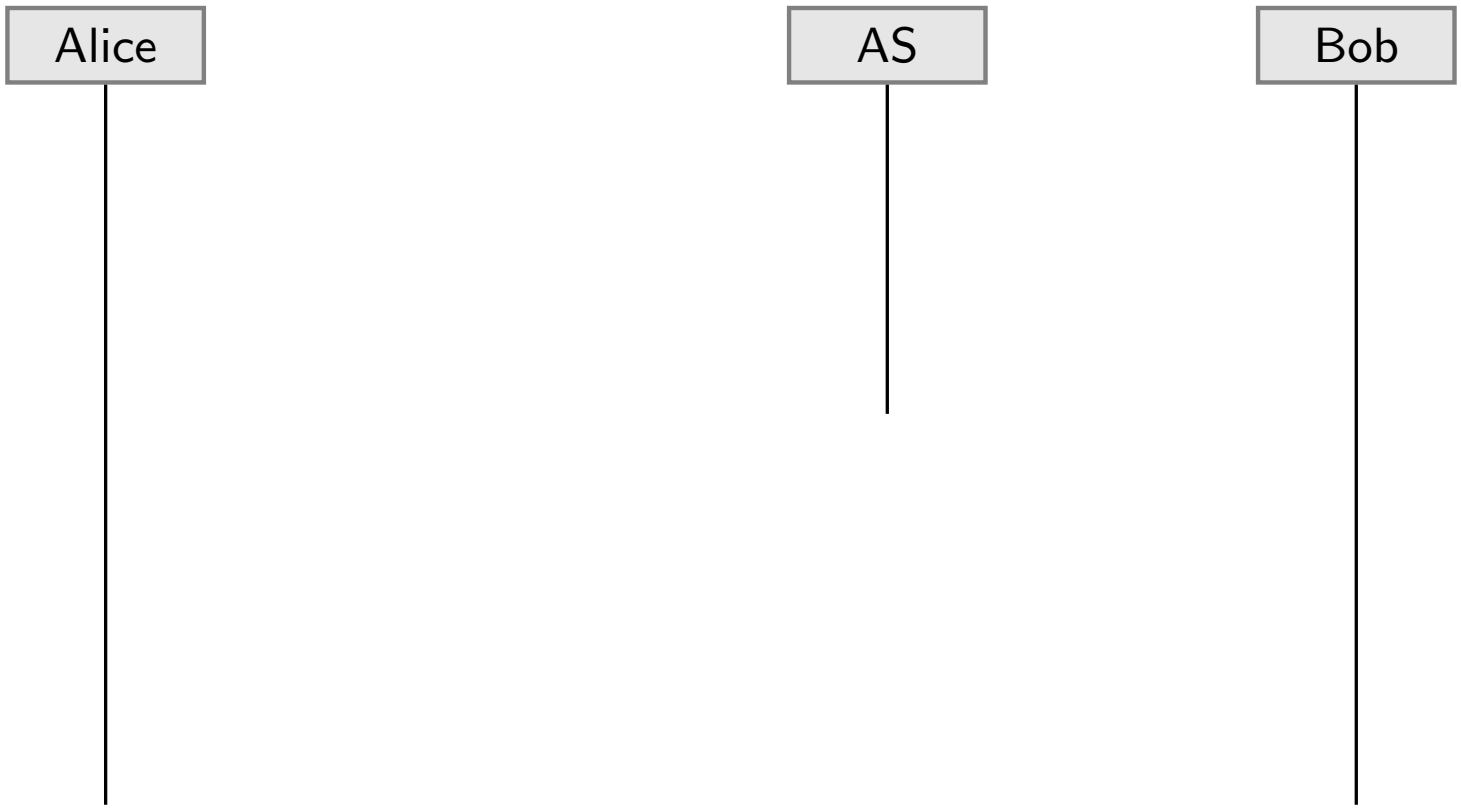


Assumption: Eve knows the old session key K_{AB_o} of Alice & Bob and also the corresponding ticket of the authentication server

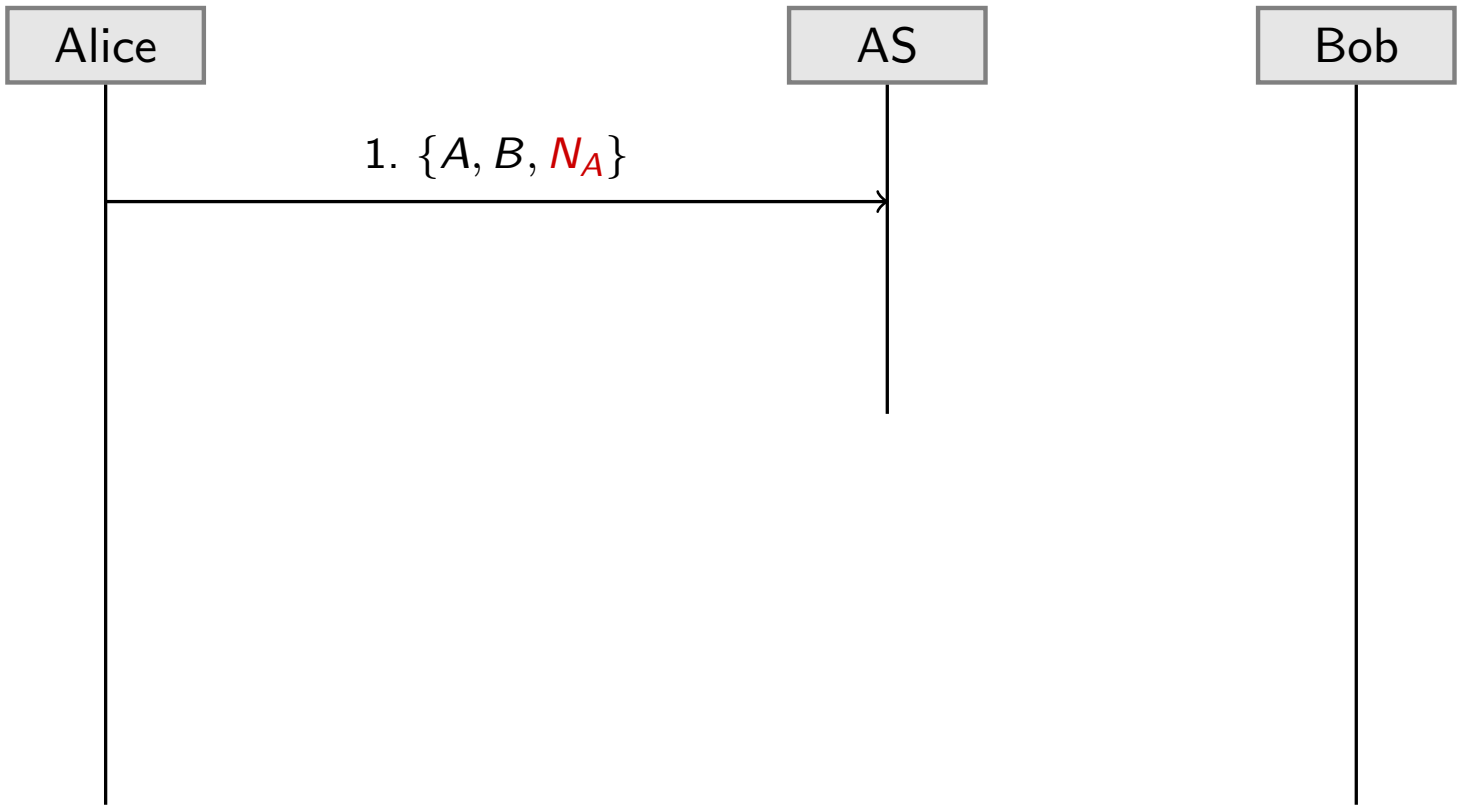
➜ How to prevent such a replay attack?

Specify a man-in-the-middle and a replay attack for the NSP example.

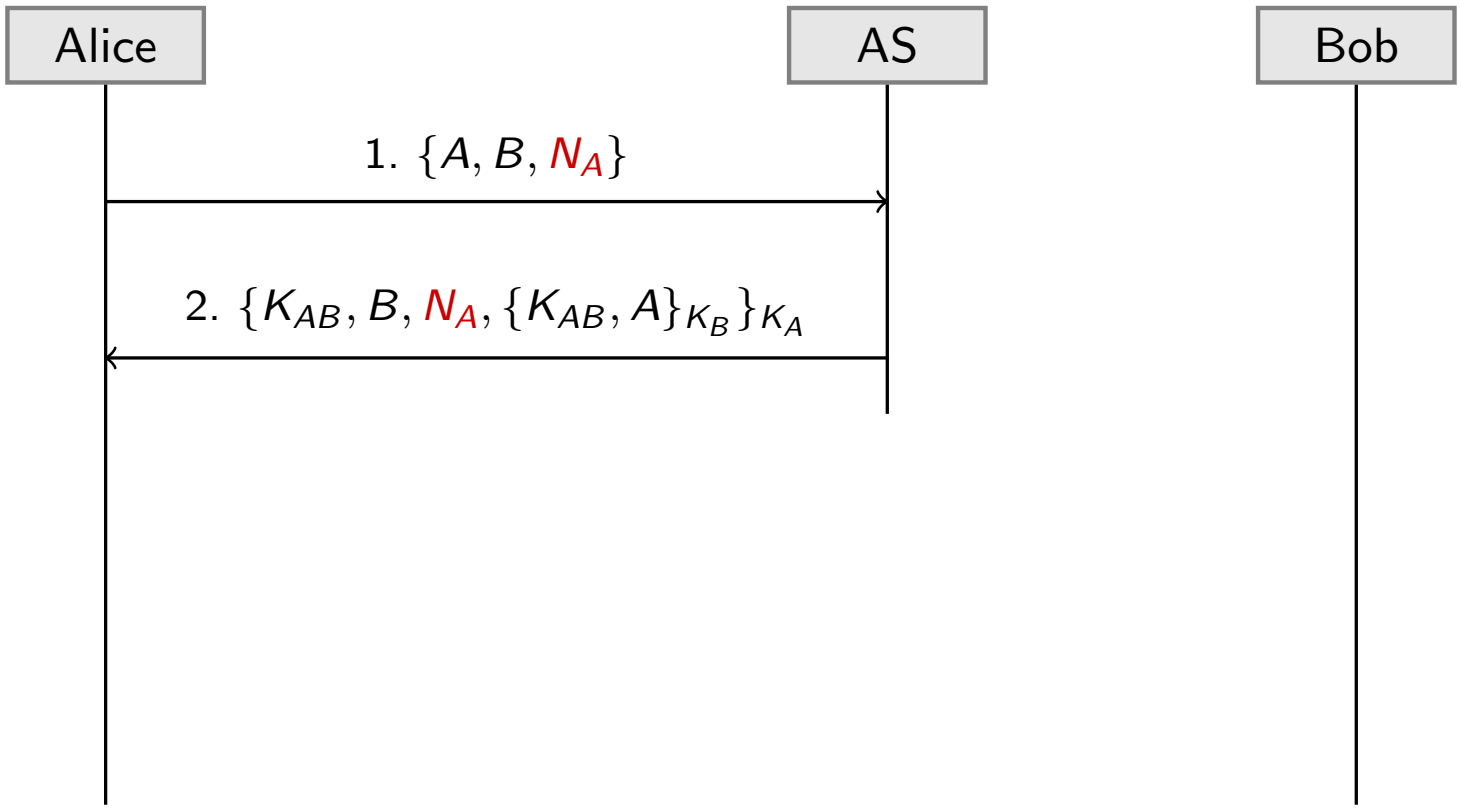
Variant 3 for the Symmetric NSP



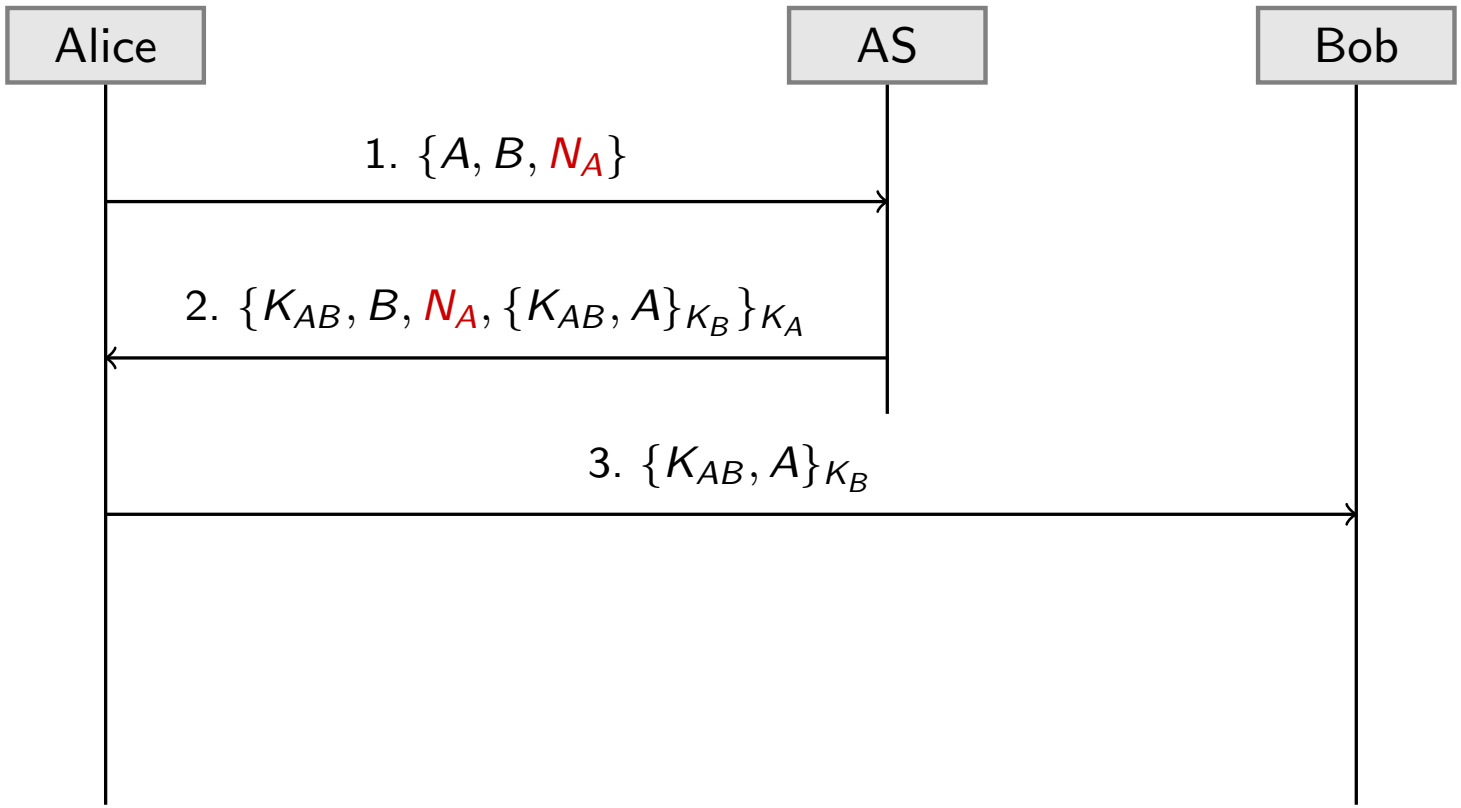
Variant 3 for the Symmetric NSP



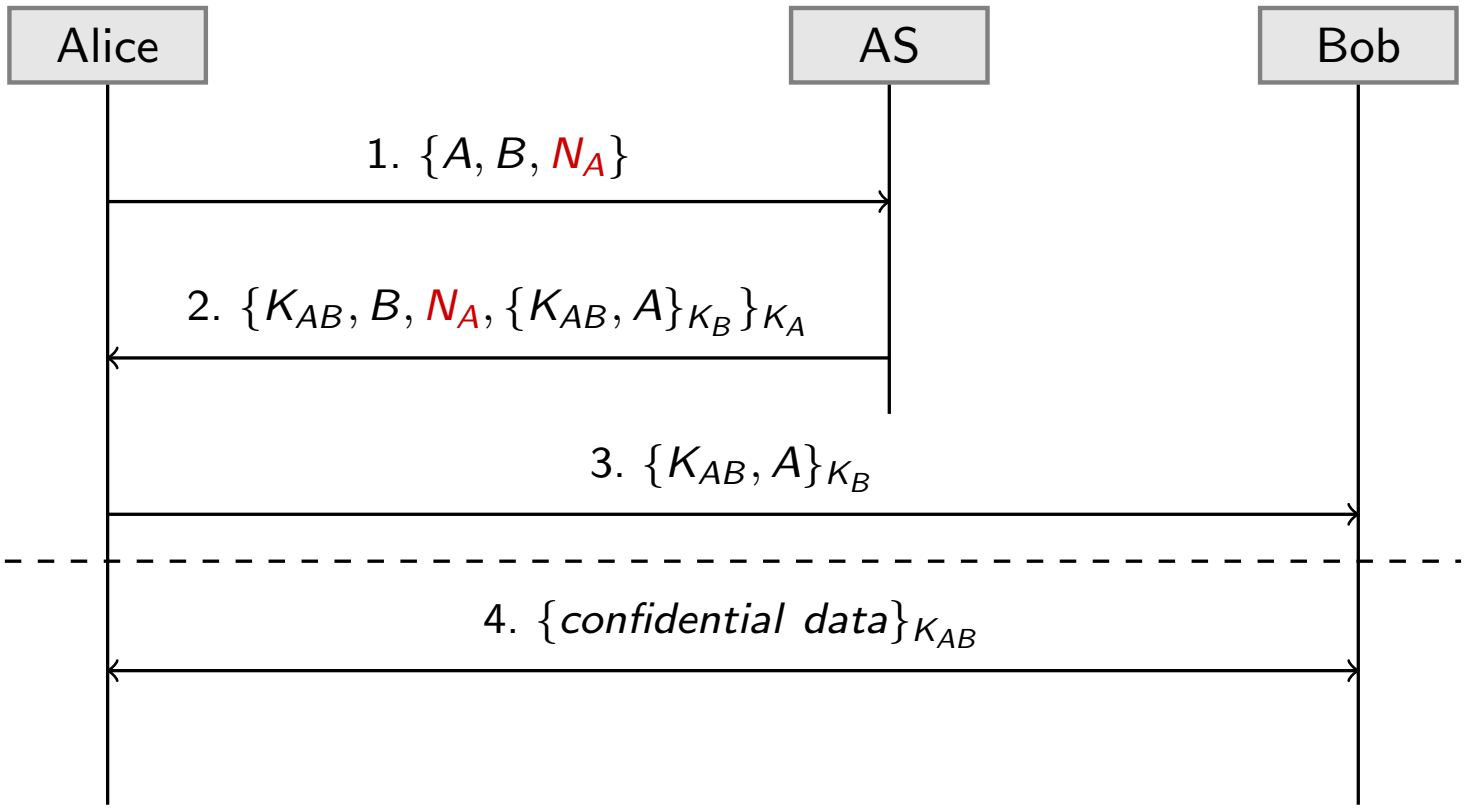
Variant 3 for the Symmetric NSP



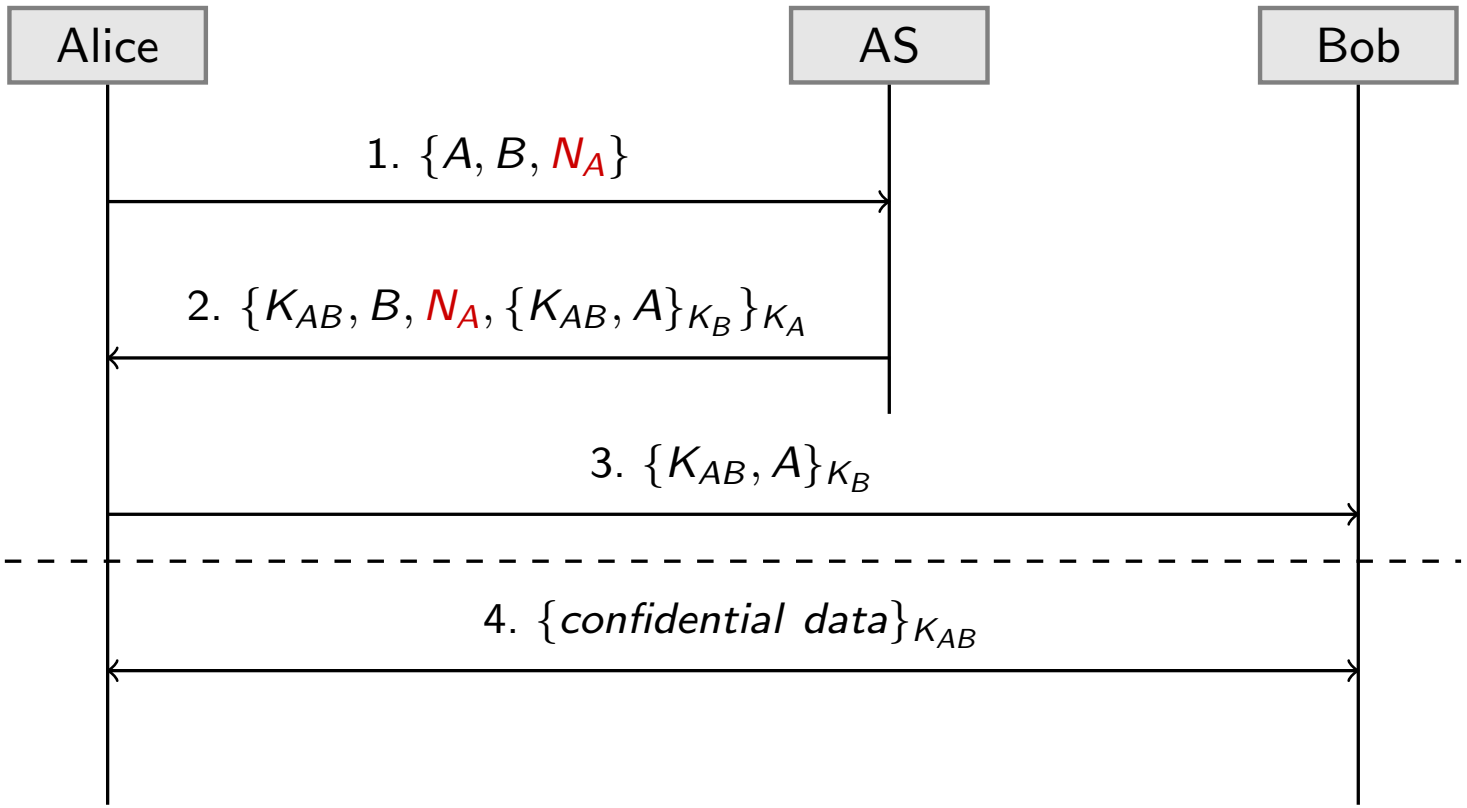
Variant 3 for the Symmetric NSP



Variant 3 for the Symmetric NSP

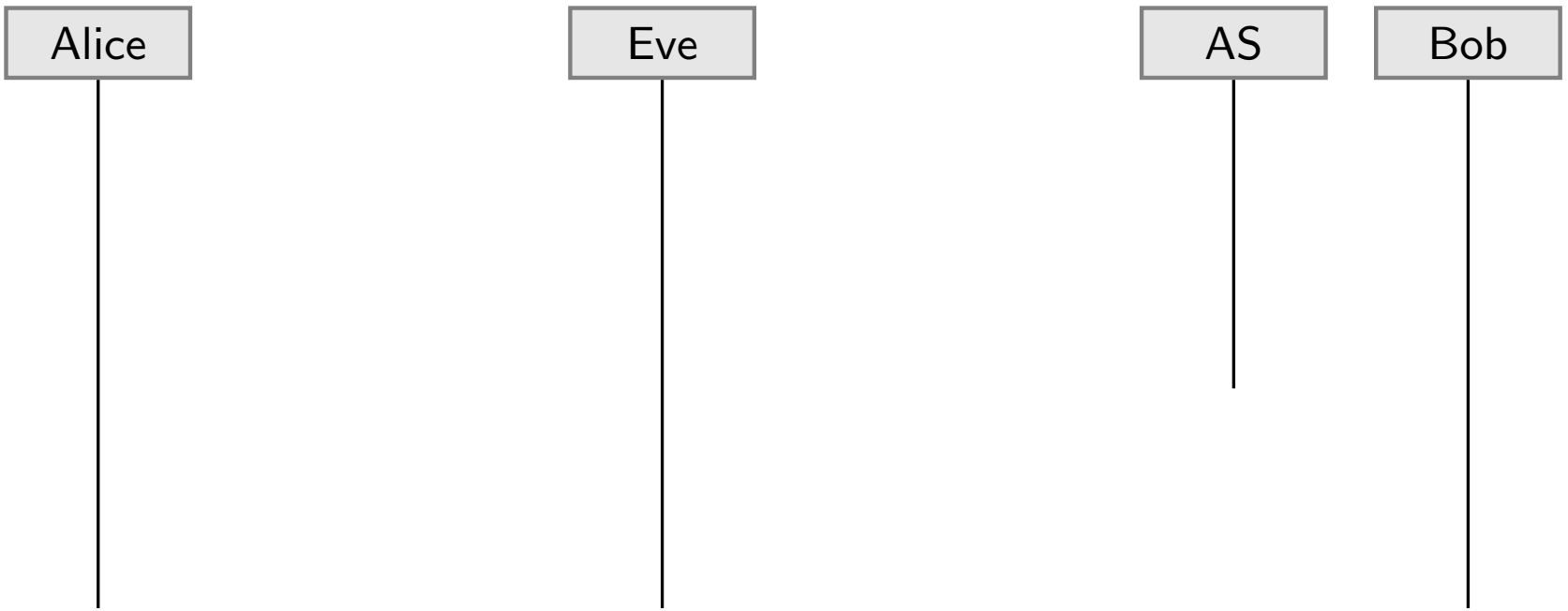


Variant 3 for the Symmetric NSP

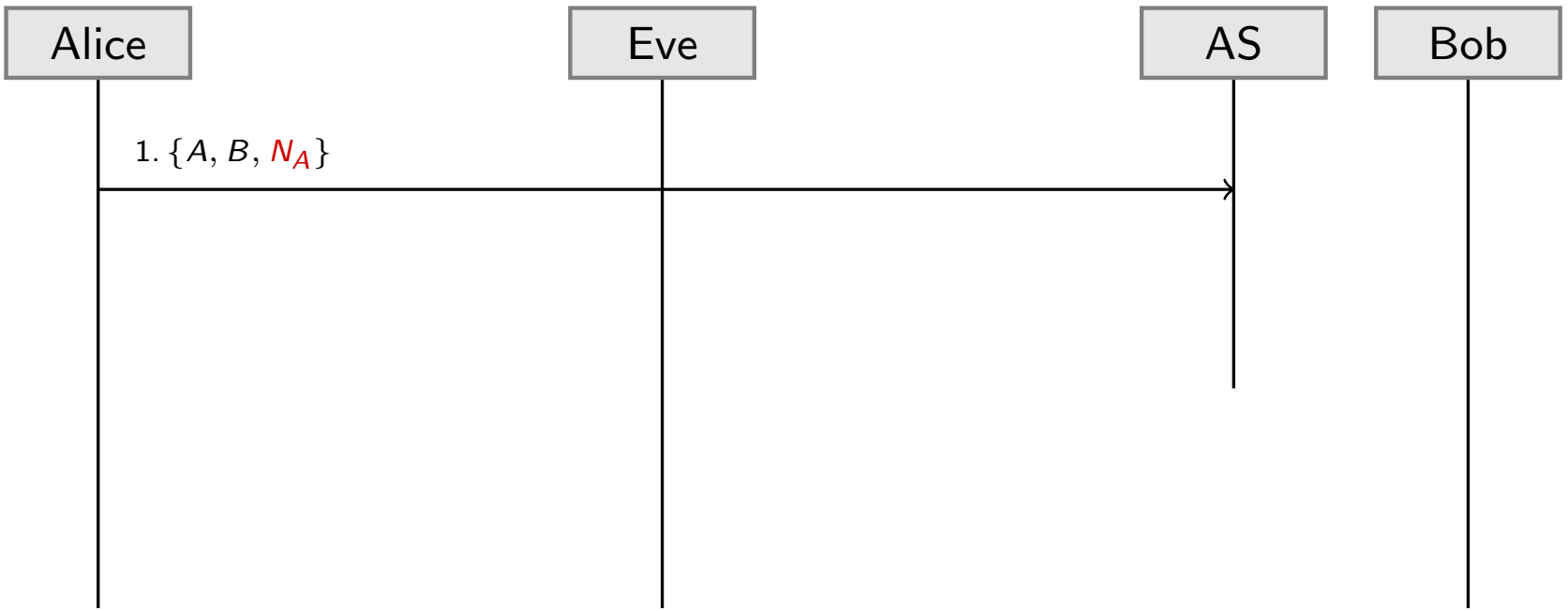


By using Nonce N_A (number used once), a correlation between Step 1 and Step 2 is implemented, such that Alice is able to check the freshness of the received ticket

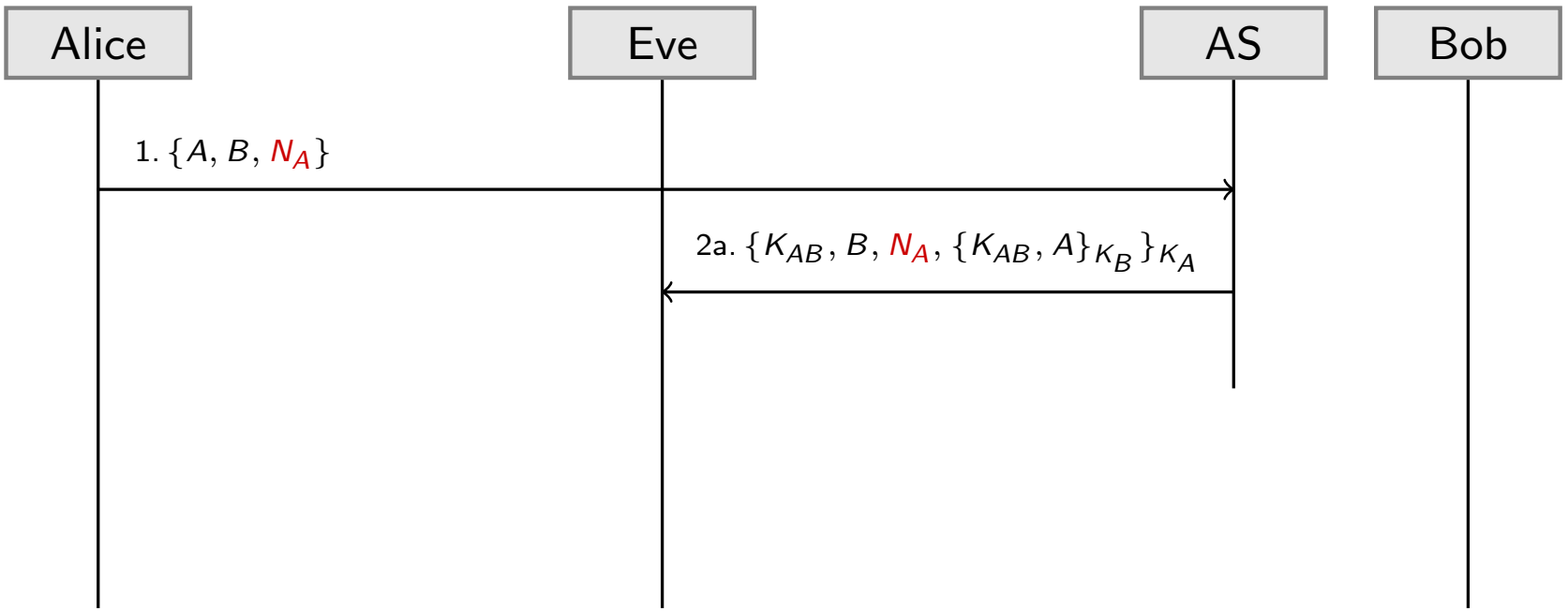
Detecting a Replay Attack



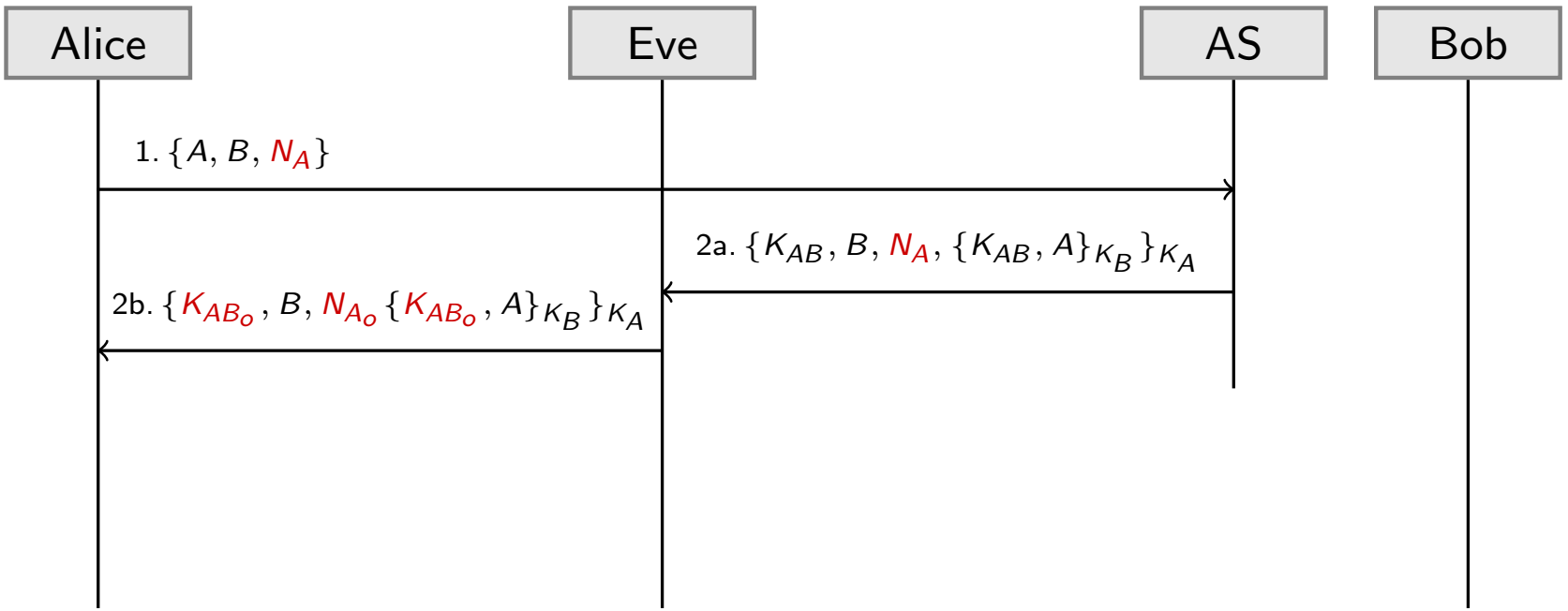
Detecting a Replay Attack



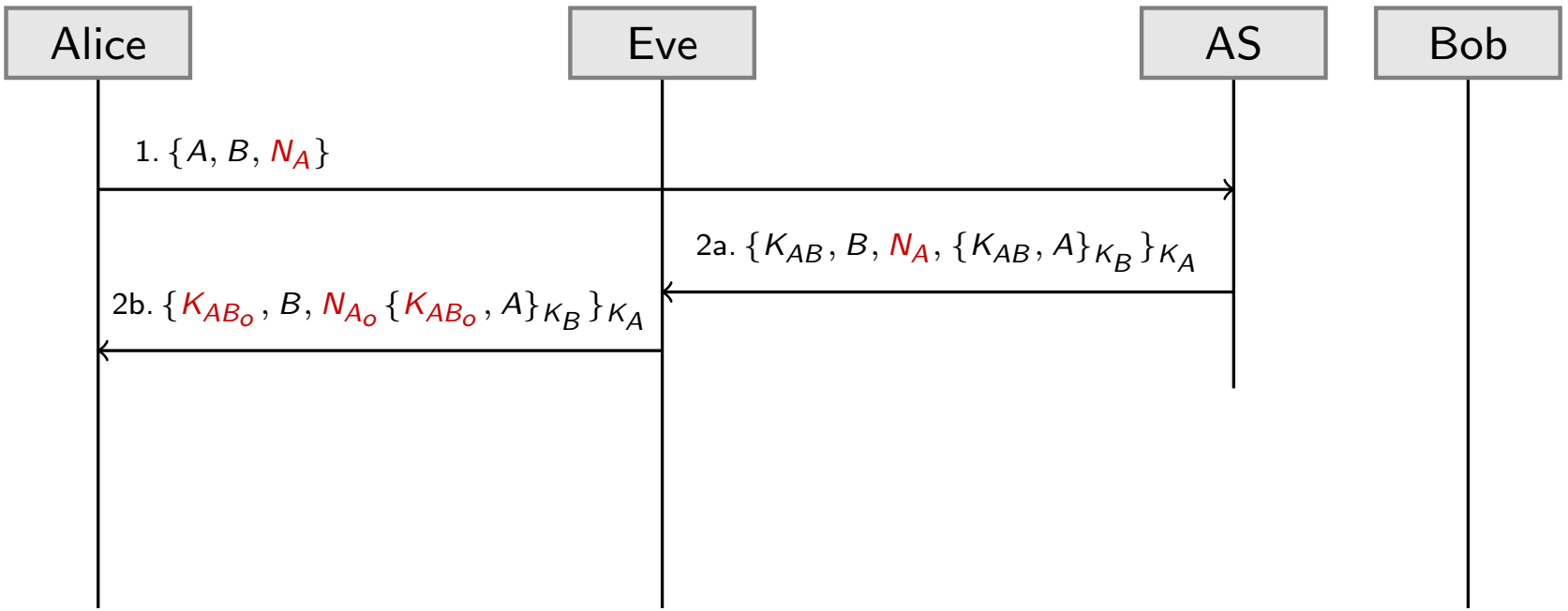
Detecting a Replay Attack



Detecting a Replay Attack

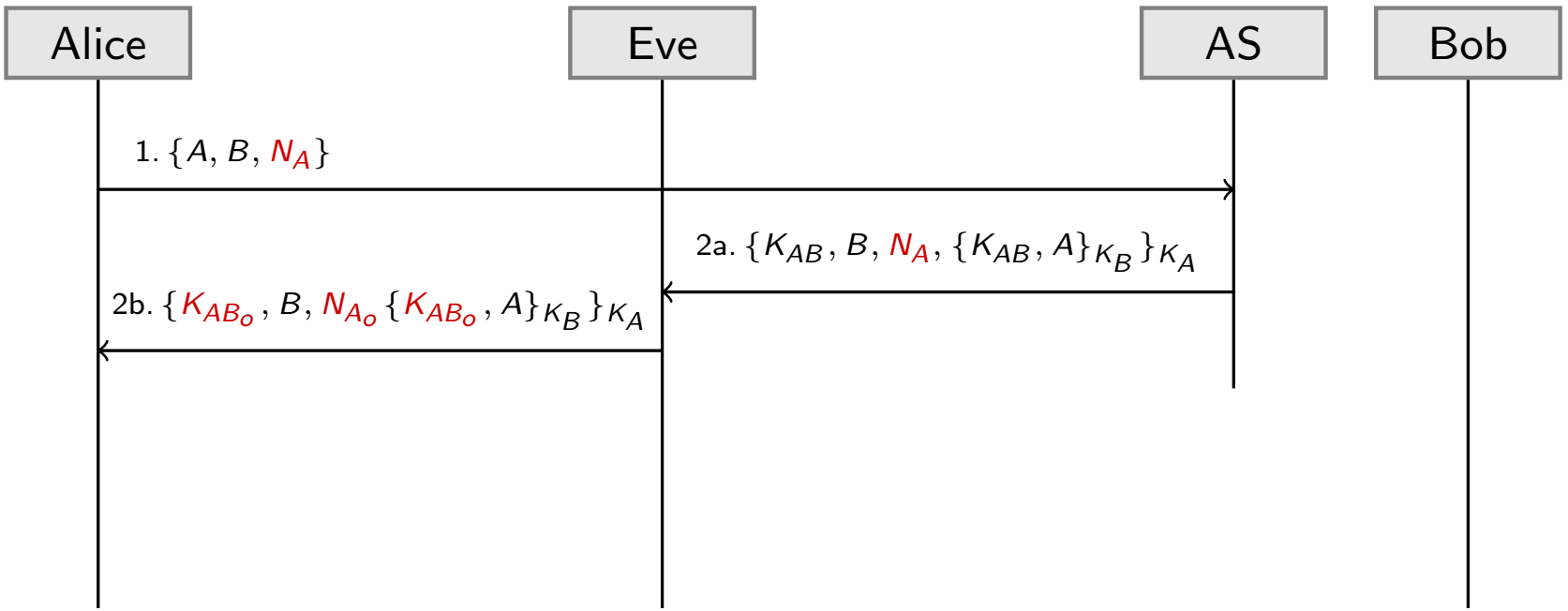


Detecting a Replay Attack



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

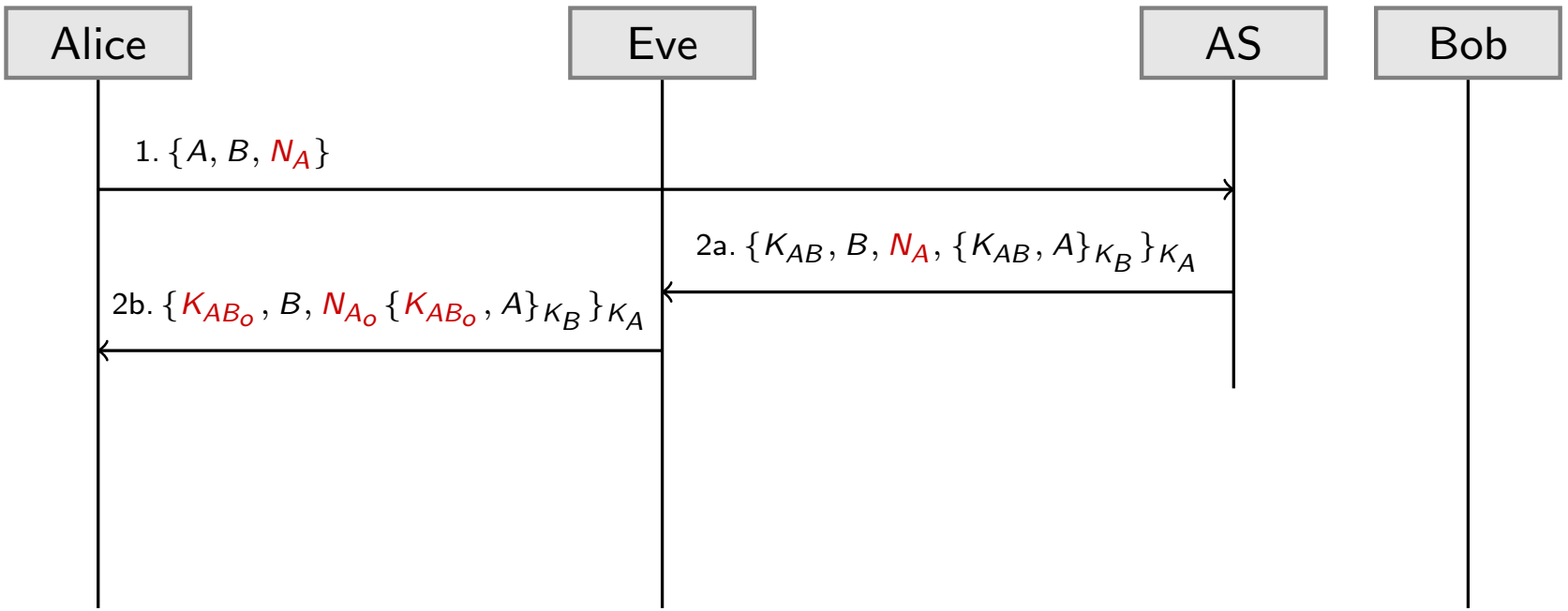
Detecting a Replay Attack



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

➜ Is the protocol now finally secure?

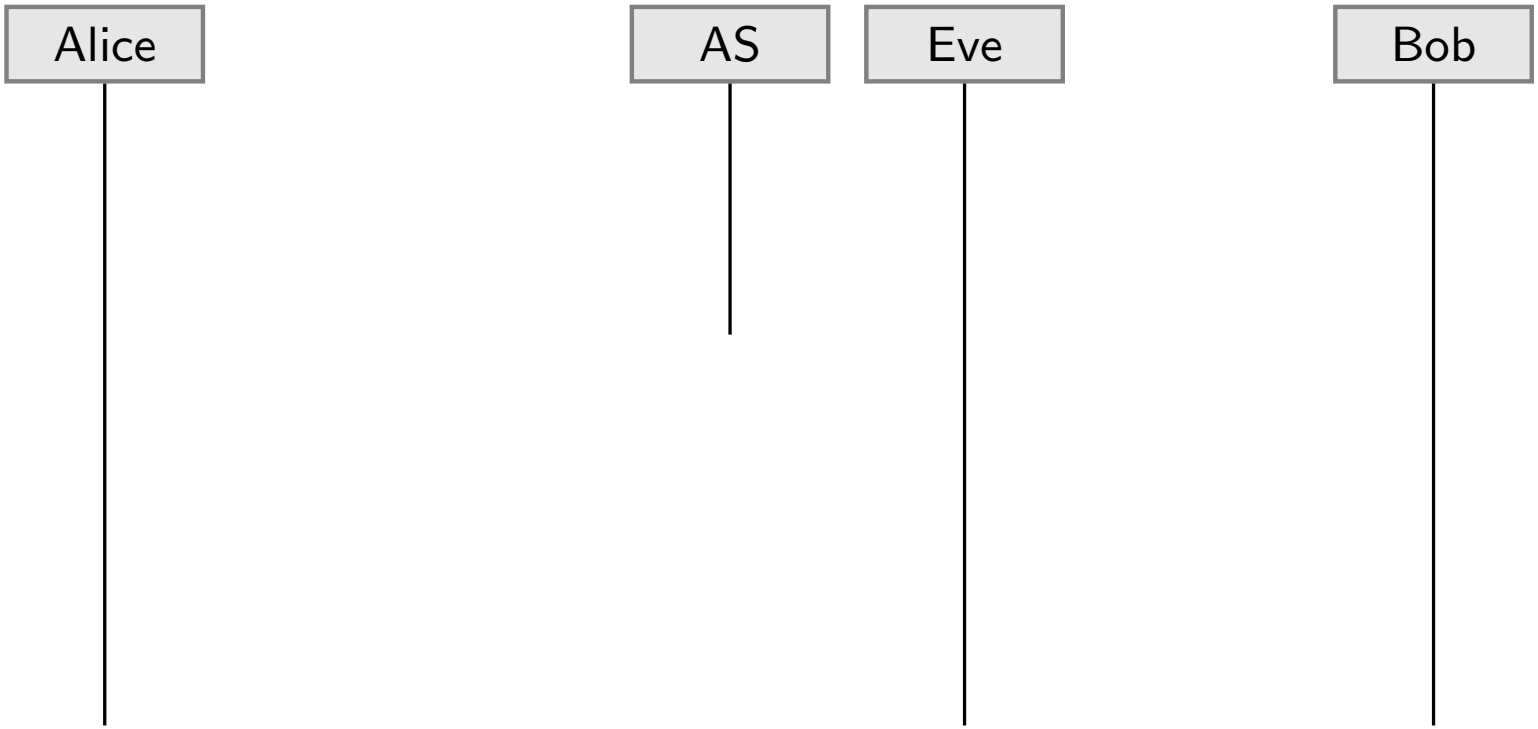
Detecting a Replay Attack



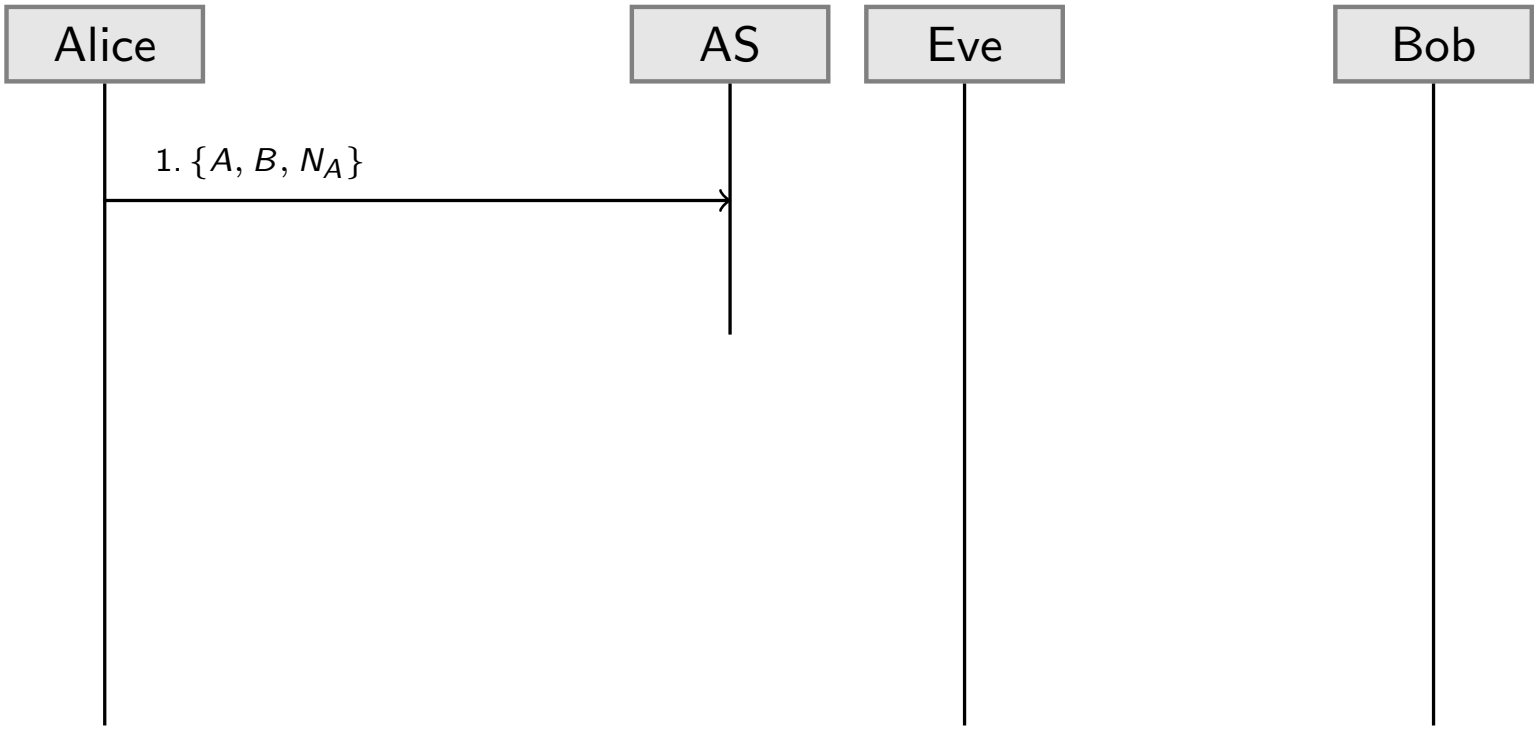
Alice detects Eve's manipulation by finding out that the Nonce N_A has been changed

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

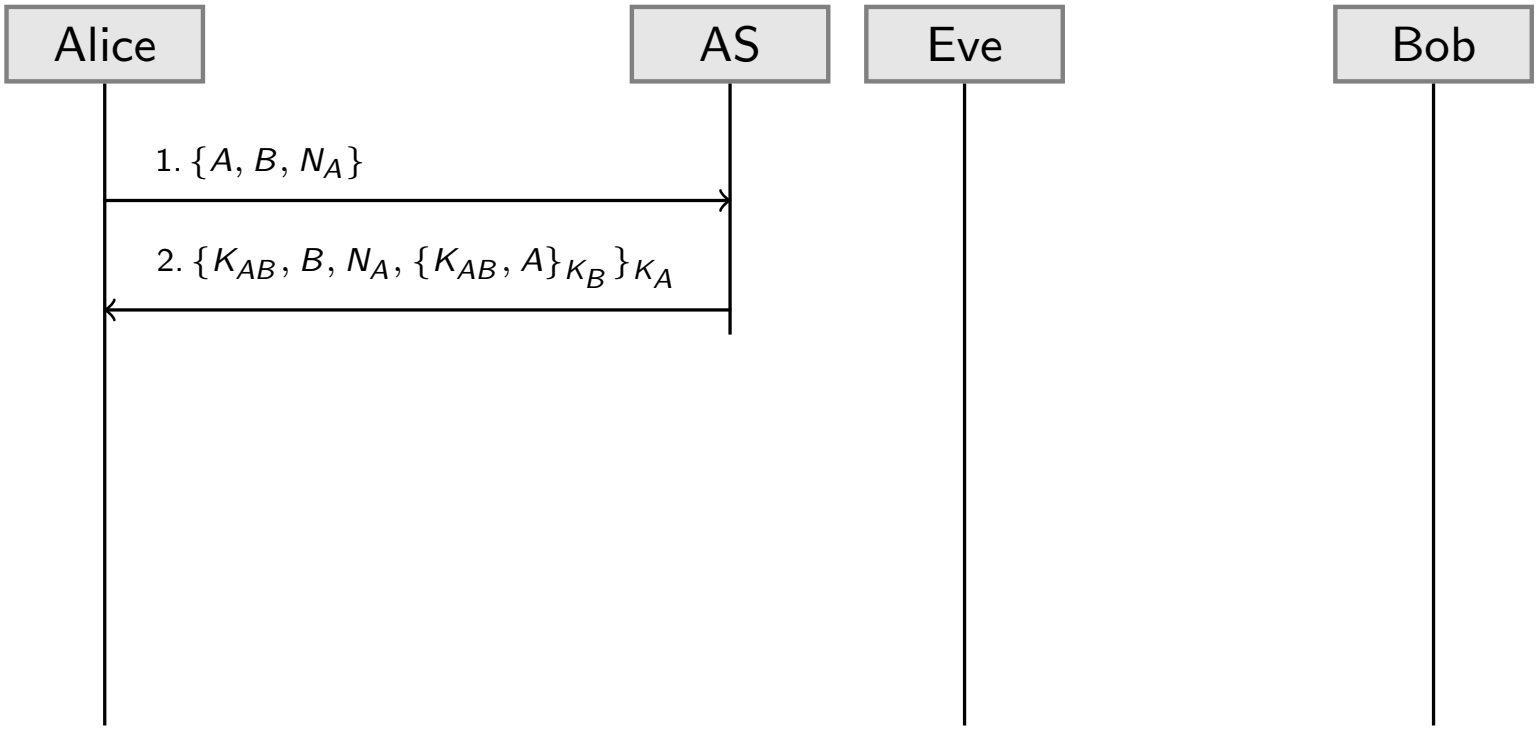
Attack for Variant 3 of the symmetric NSP



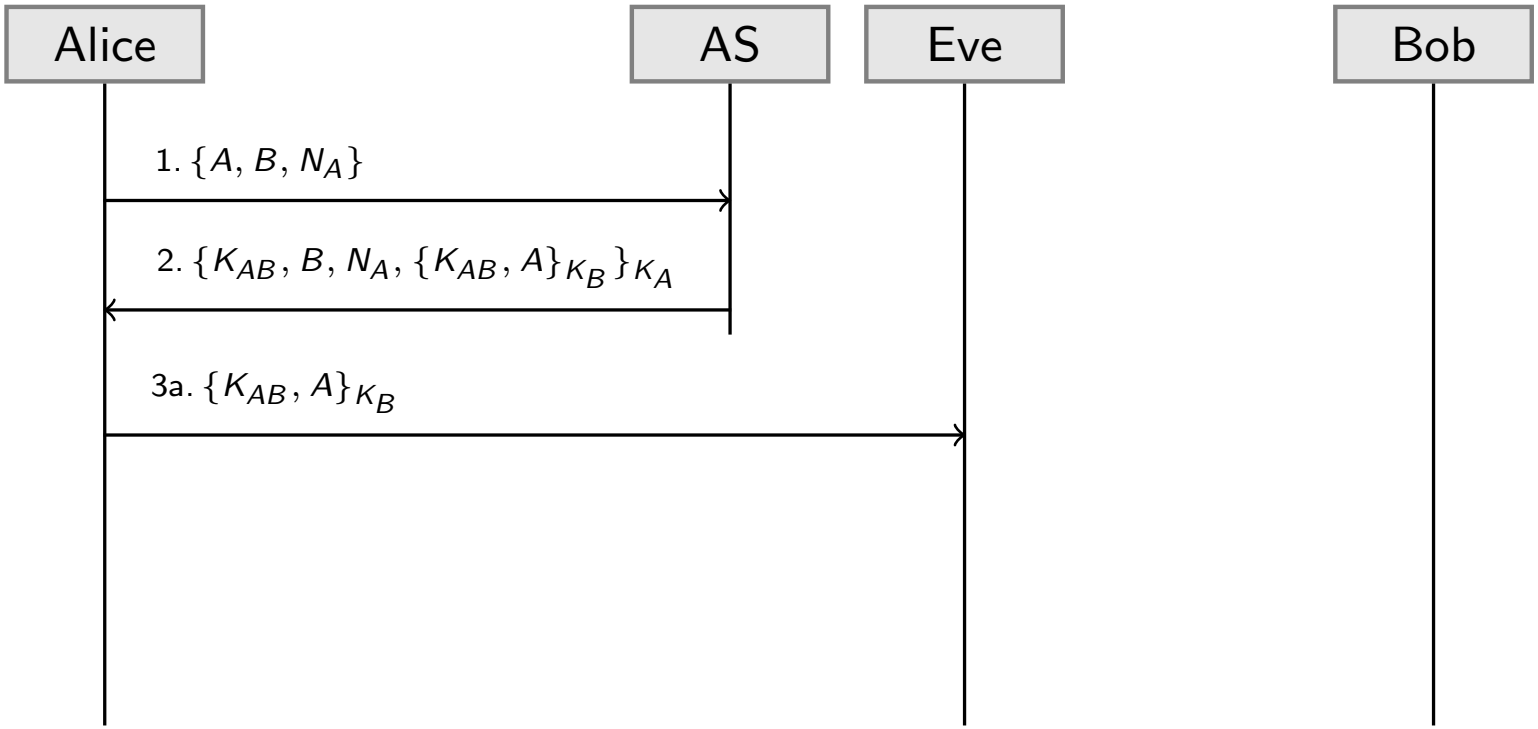
Attack for Variant 3 of the symmetric NSP



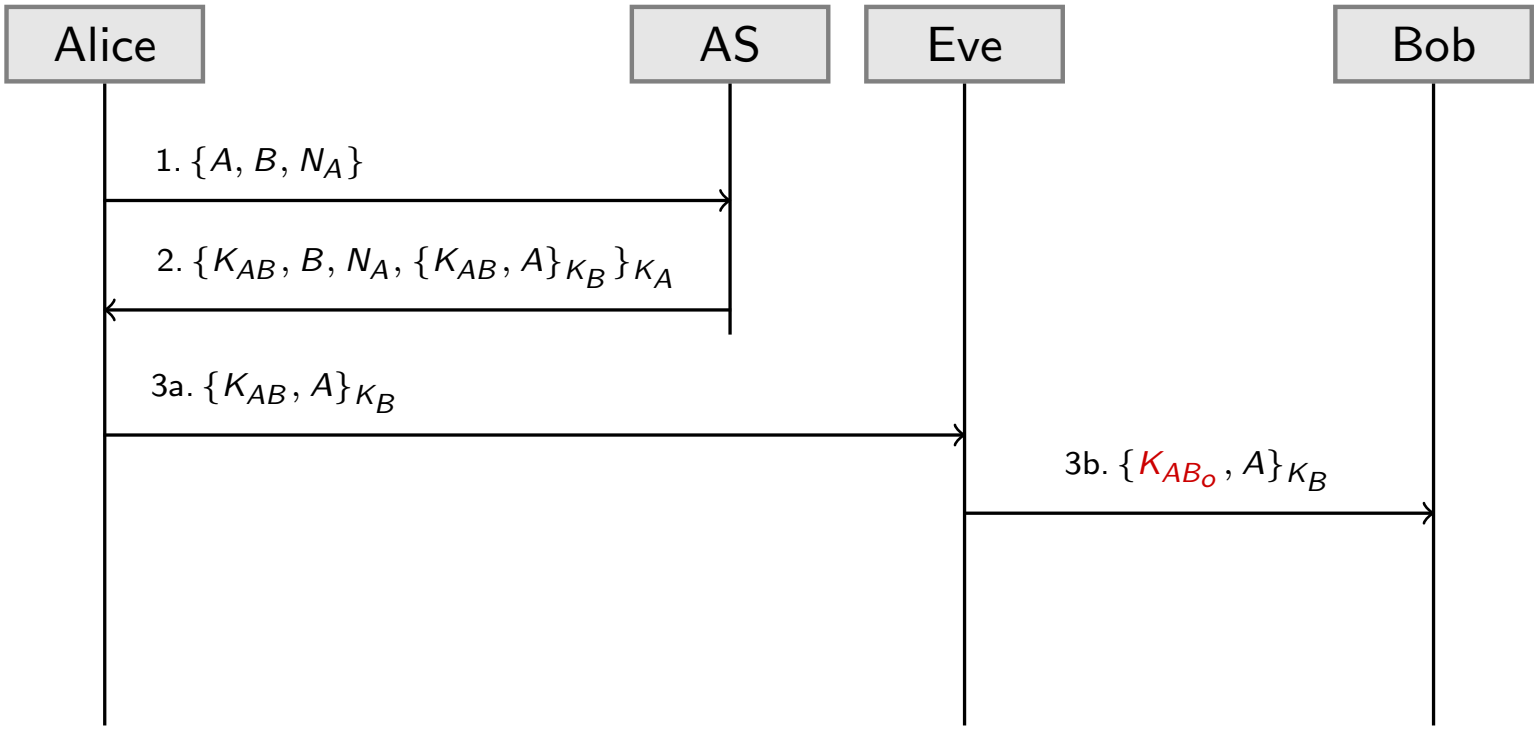
Attack for Variant 3 of the symmetric NSP



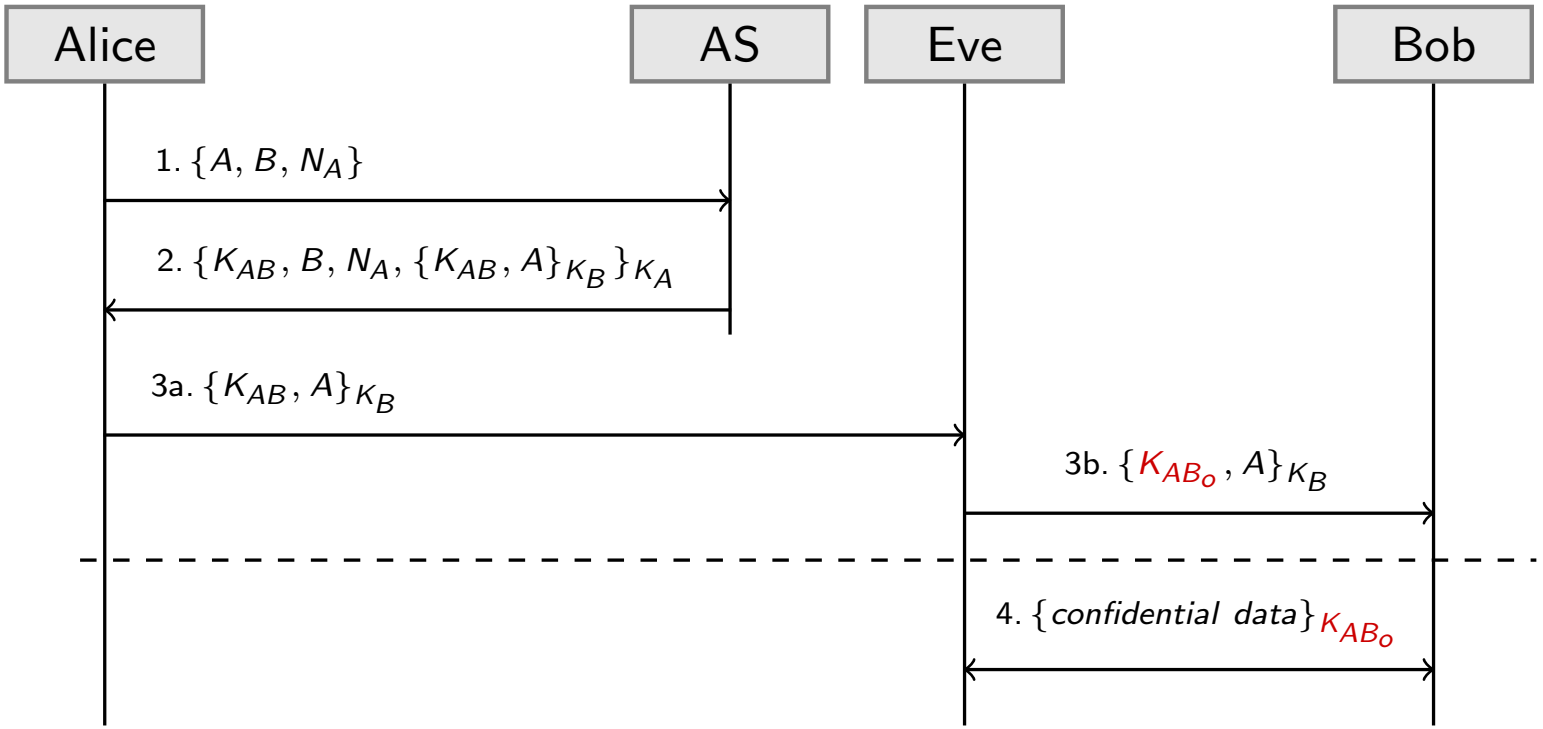
Attack for Variant 3 of the symmetric NSP



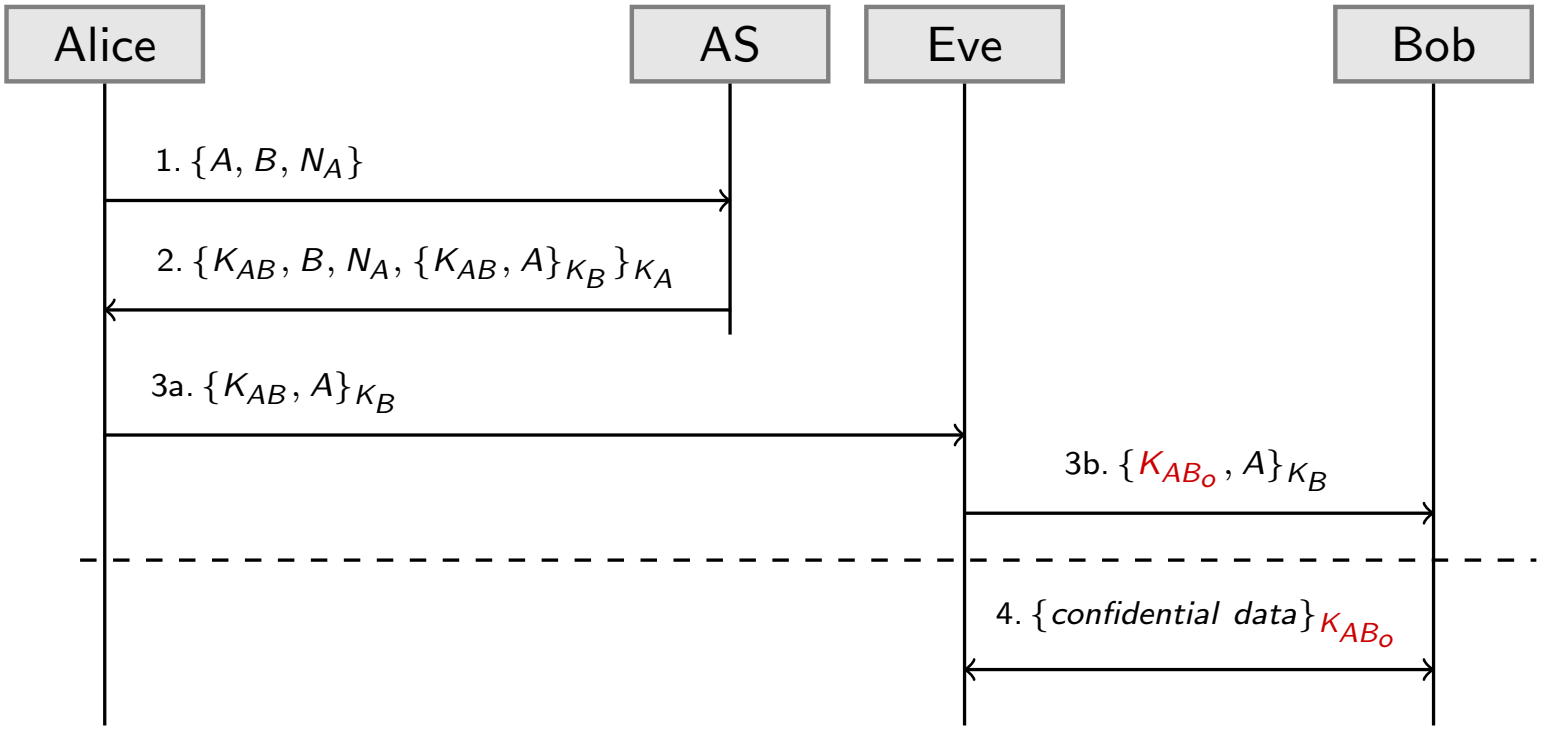
Attack for Variant 3 of the symmetric NSP



Attack for Variant 3 of the symmetric NSP

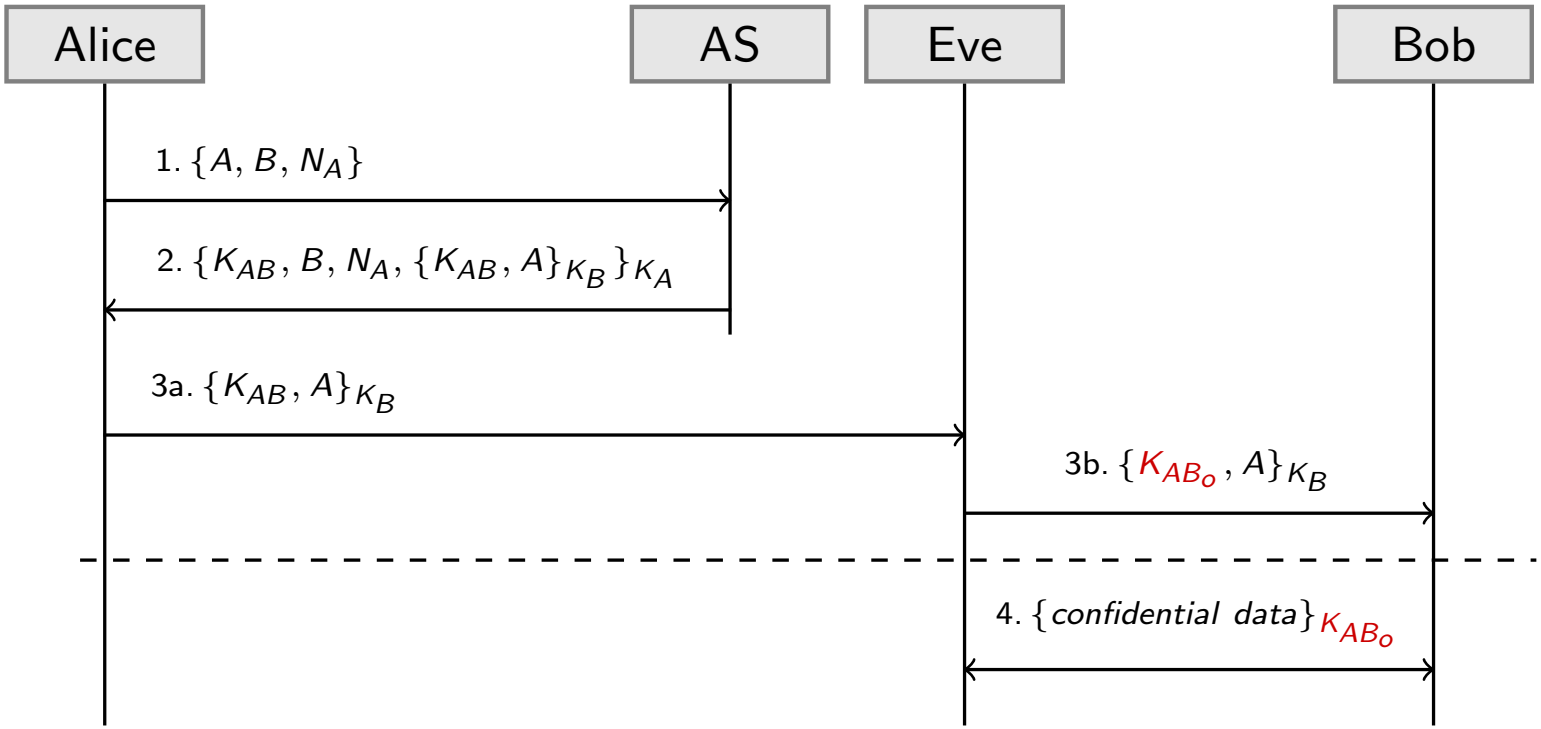


Attack for Variant 3 of the symmetric NSP



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

Attack for Variant 3 of the symmetric NSP

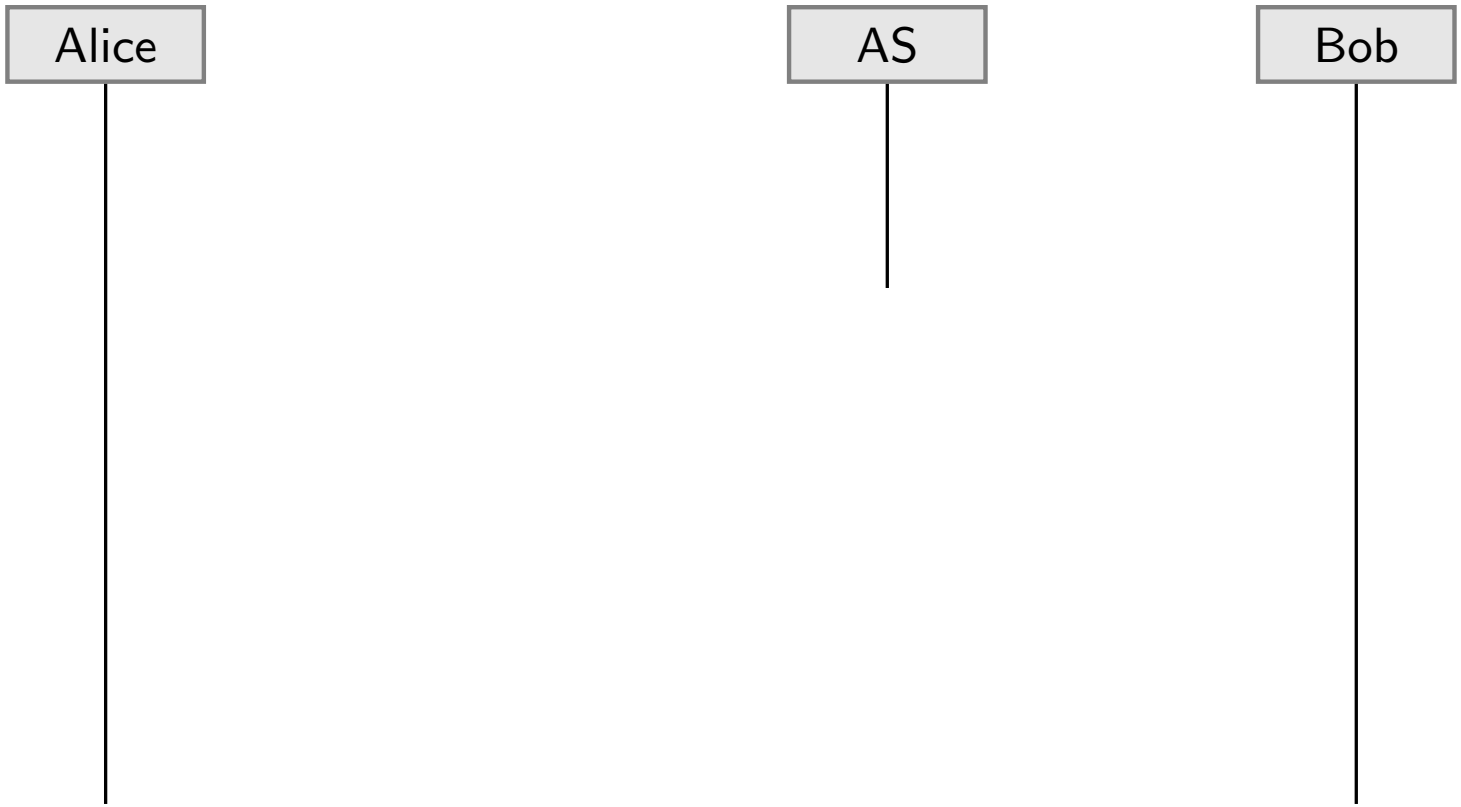


Assumption: Eve knows the old session key K_{AB_o} of Alice & Bob and also the corresponding ticket of Step 3b

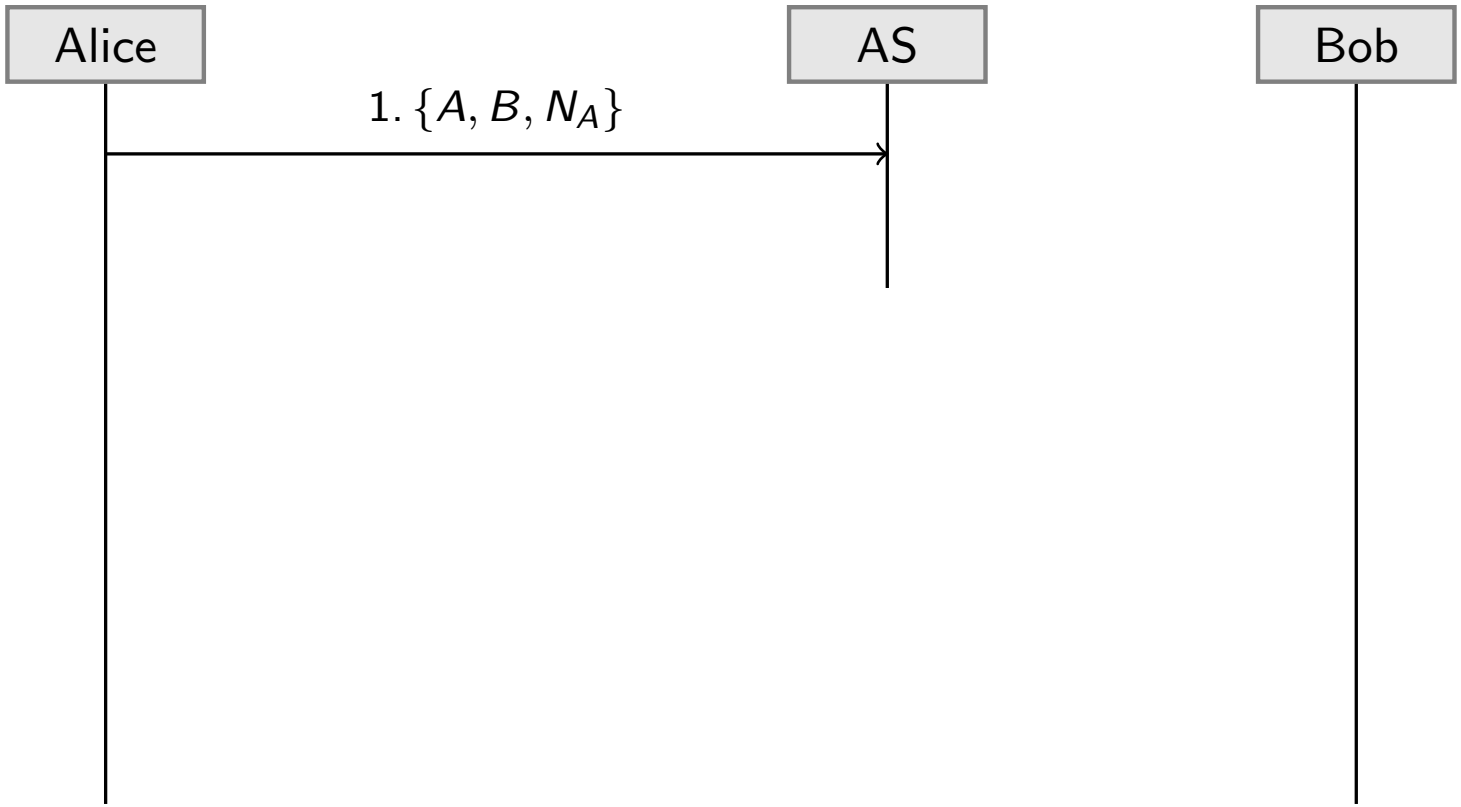
➜ How can this replay attack against Bob be prevented?

Specify a man-in-the-middle and a replay attack for the NSP example.

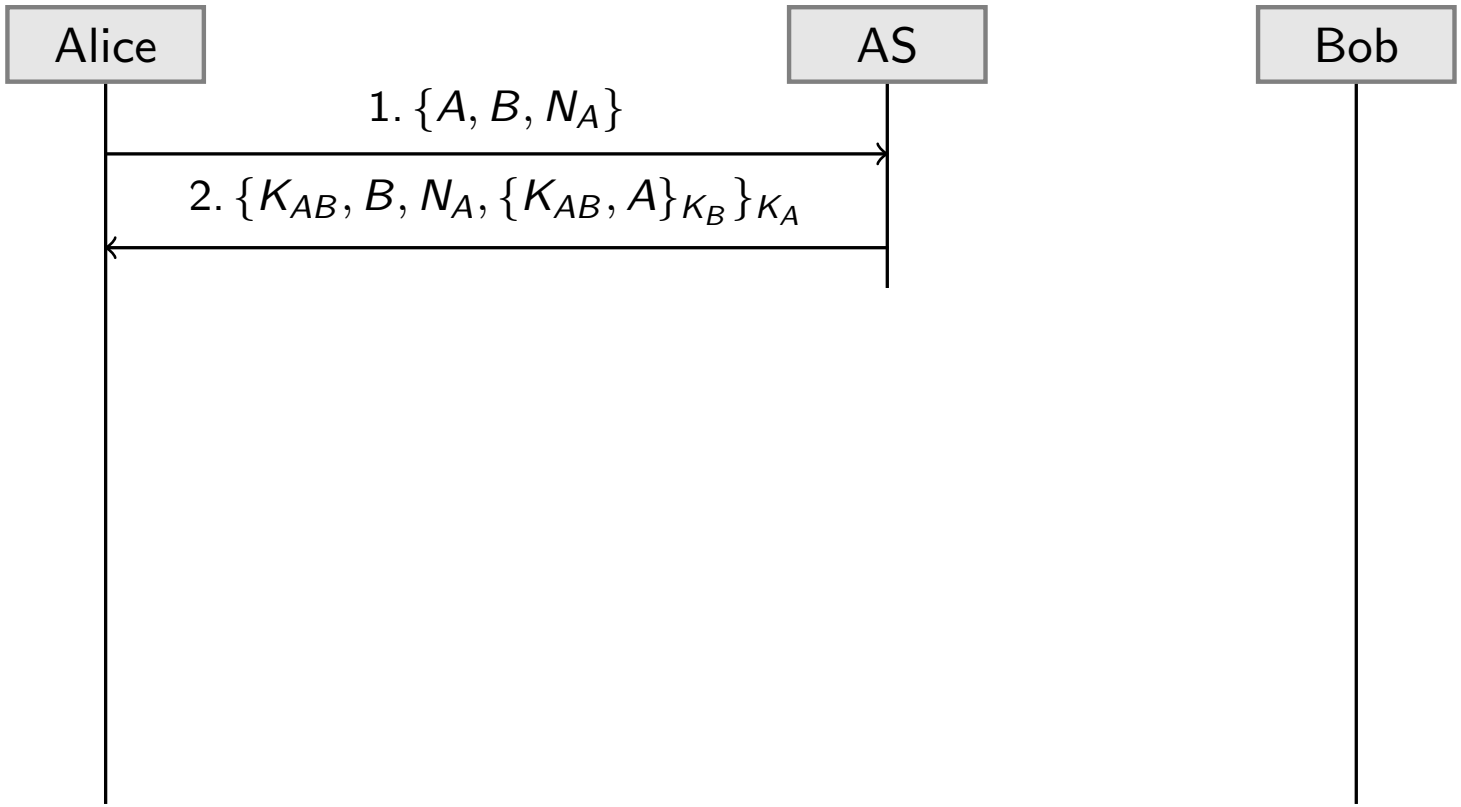
Variant 4: Symmetric NSP with Handshake



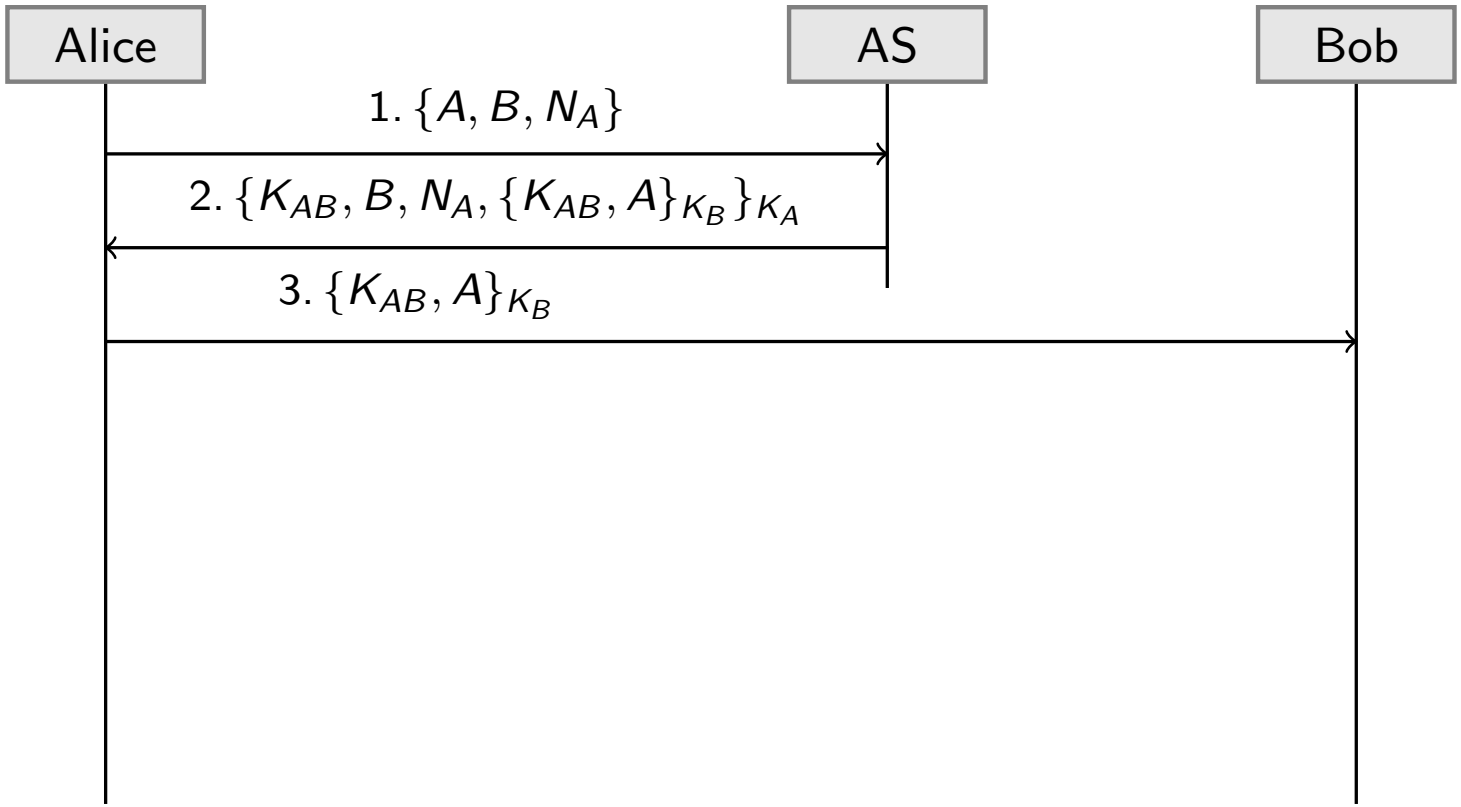
Variant 4: Symmetric NSP with Handshake



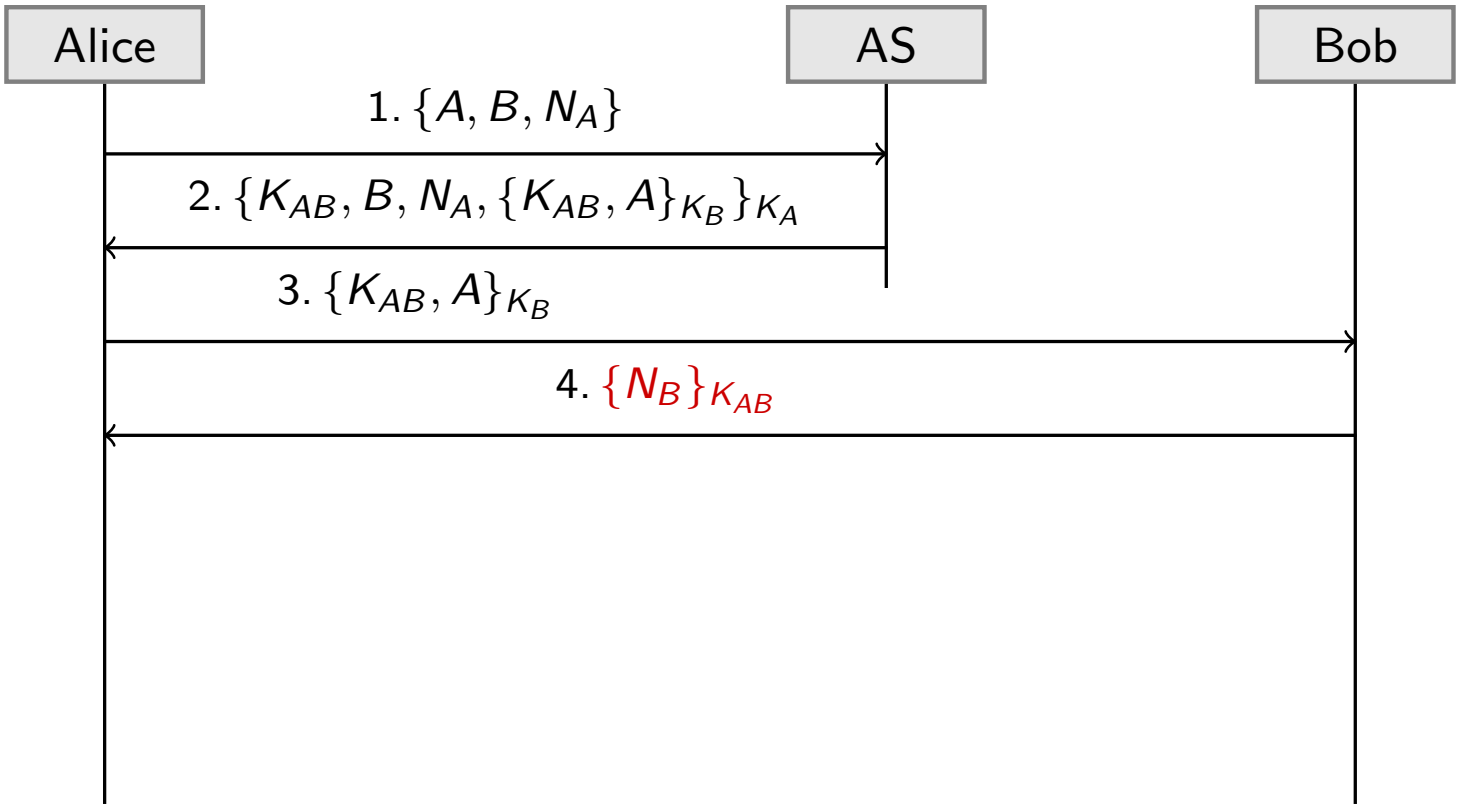
Variant 4: Symmetric NSP with Handshake



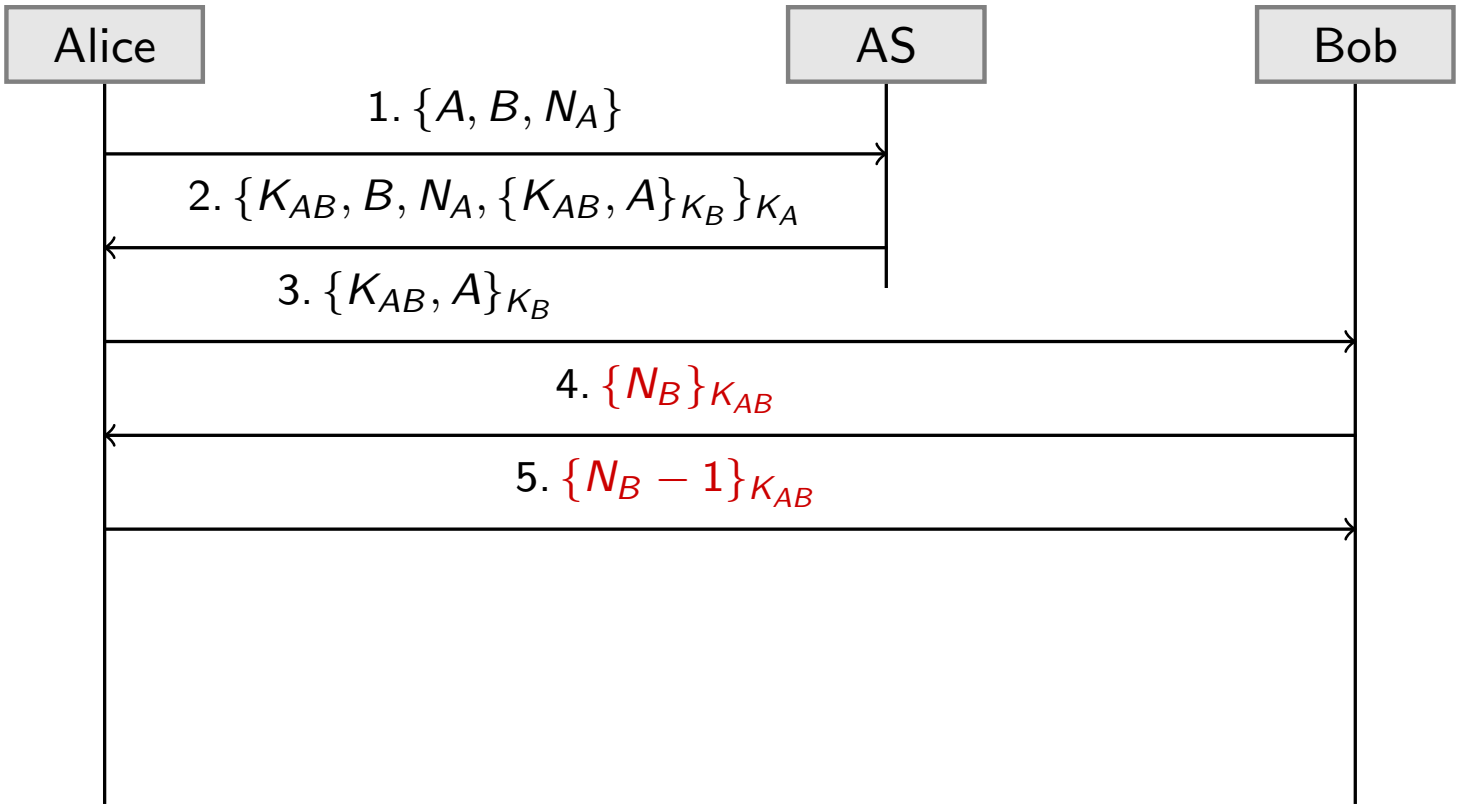
Variant 4: Symmetric NSP with Handshake



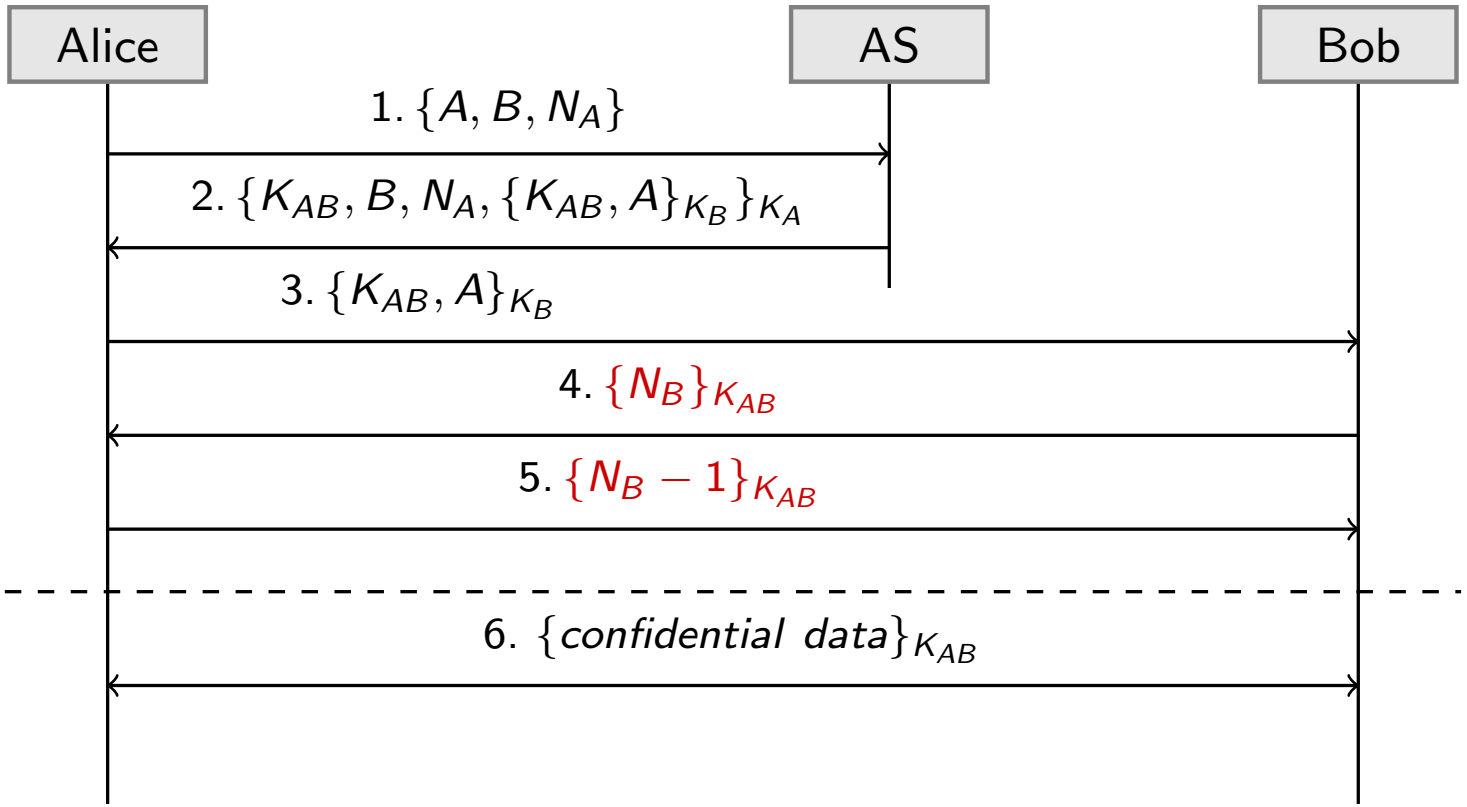
Variant 4: Symmetric NSP with Handshake



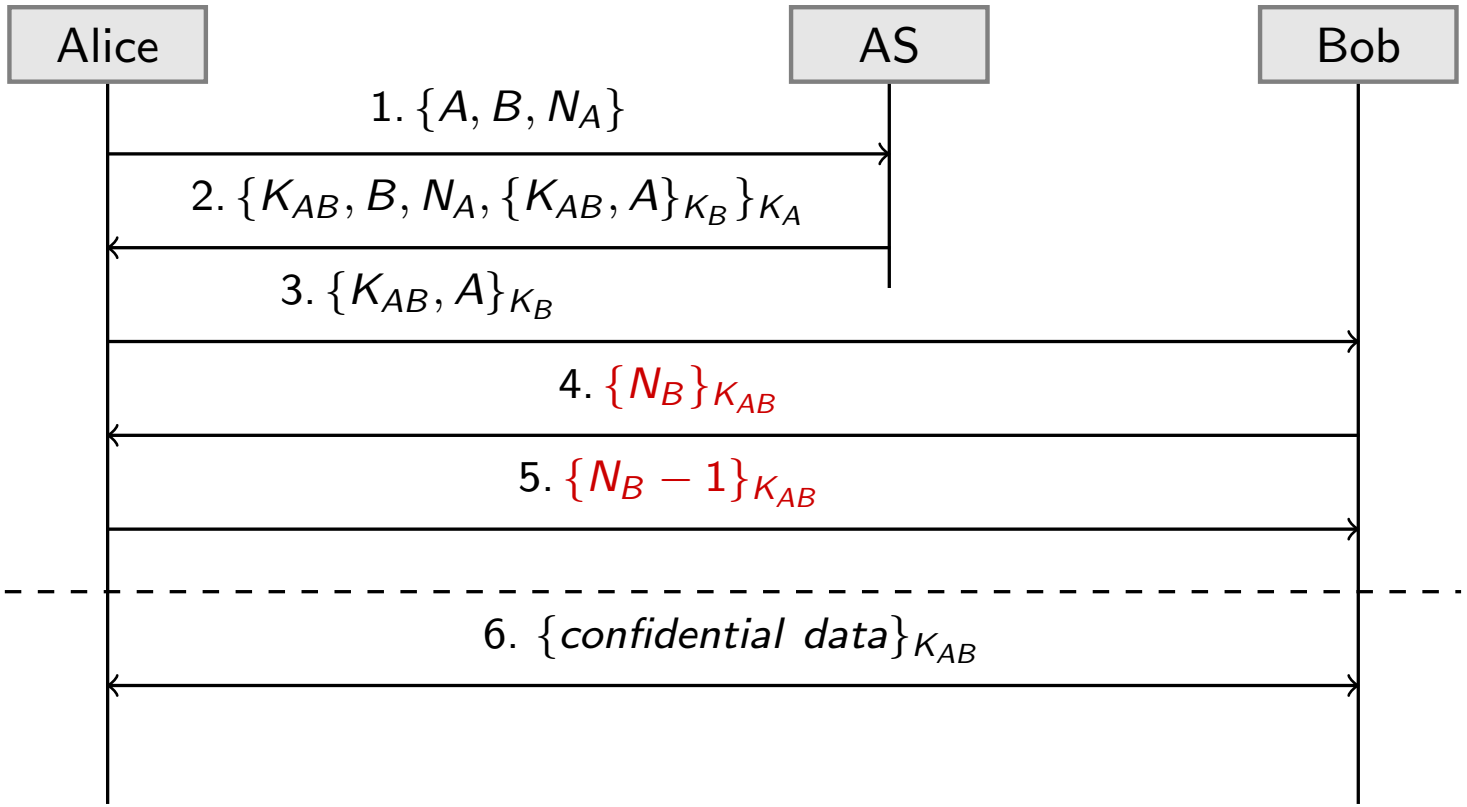
Variant 4: Symmetric NSP with Handshake



Variant 4: Symmetric NSP with Handshake

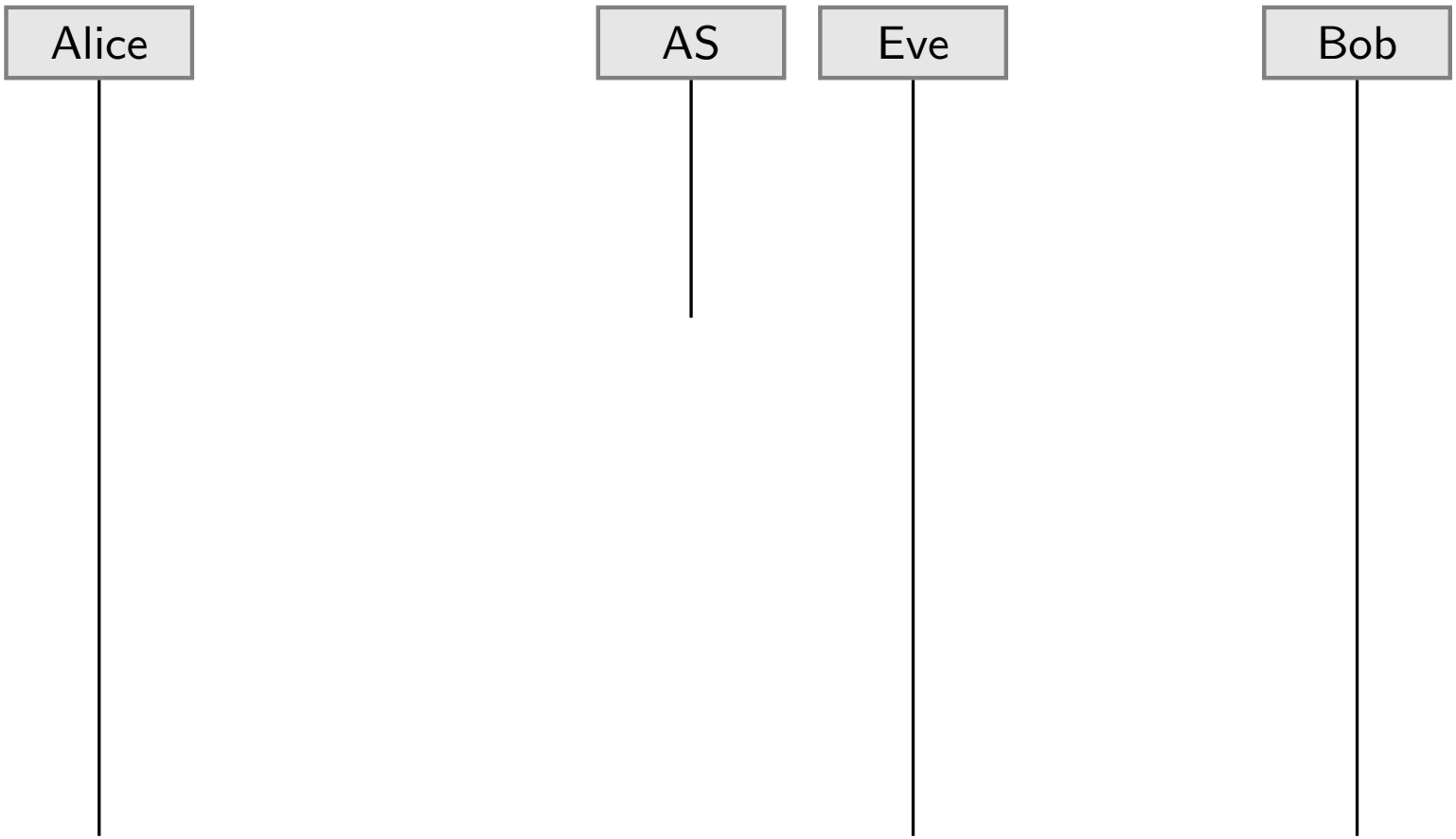


Variant 4: Symmetric NSP with Handshake

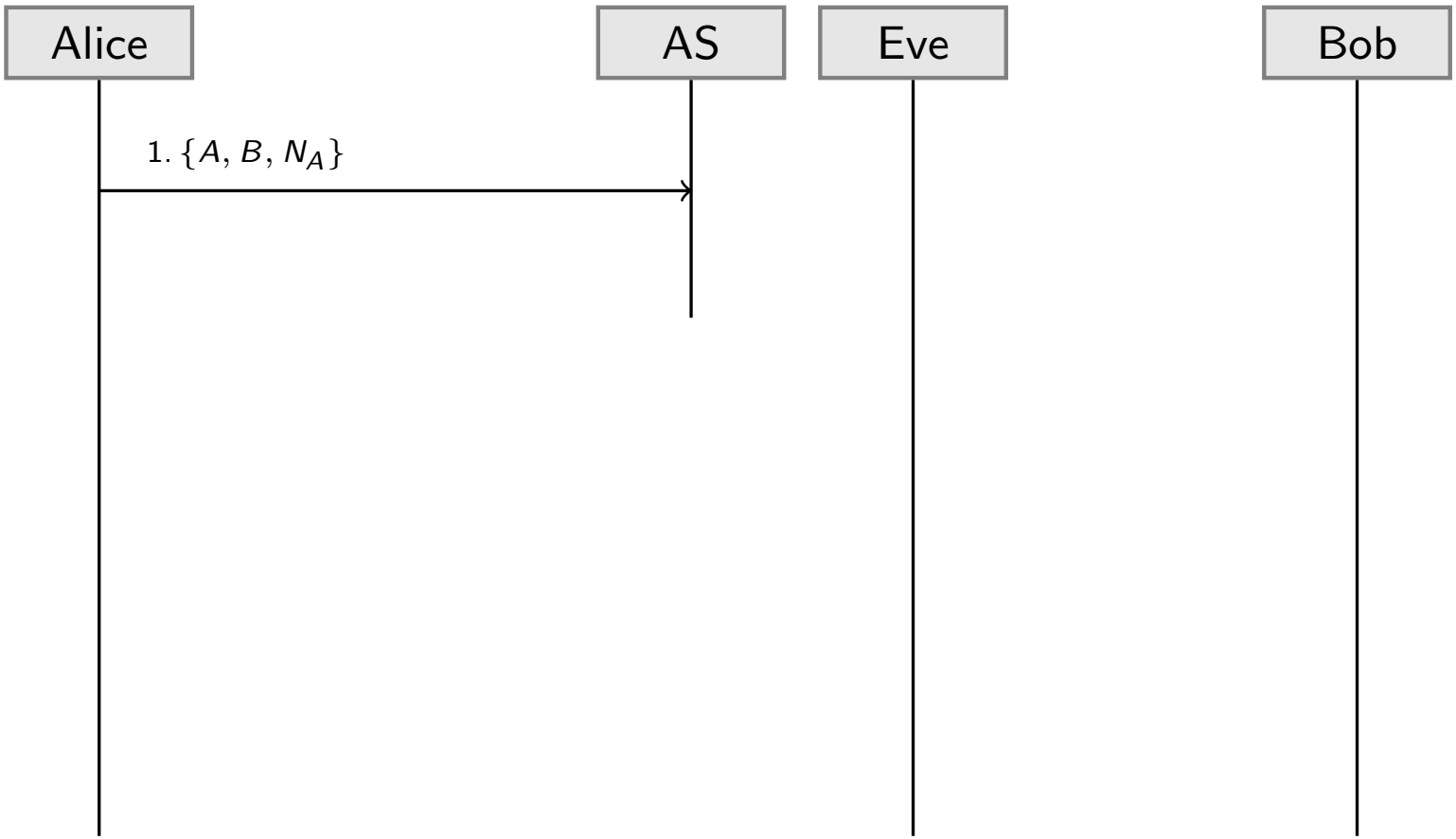


The handshake implemented in the original NSP does *not offer* Bob additional protection against replay attacks! Why?

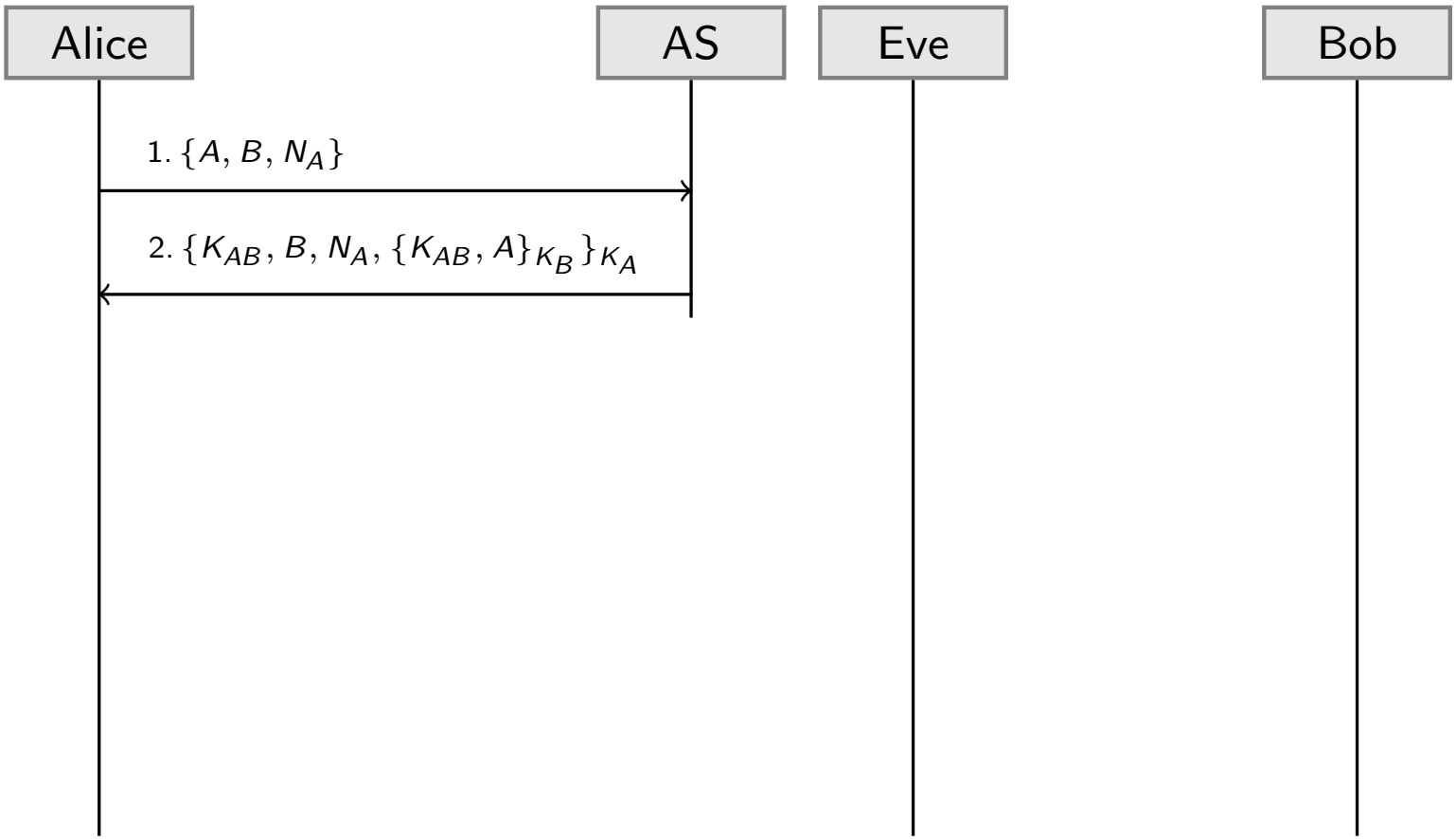
Attack of Variant 4 of the Symmetric NSP



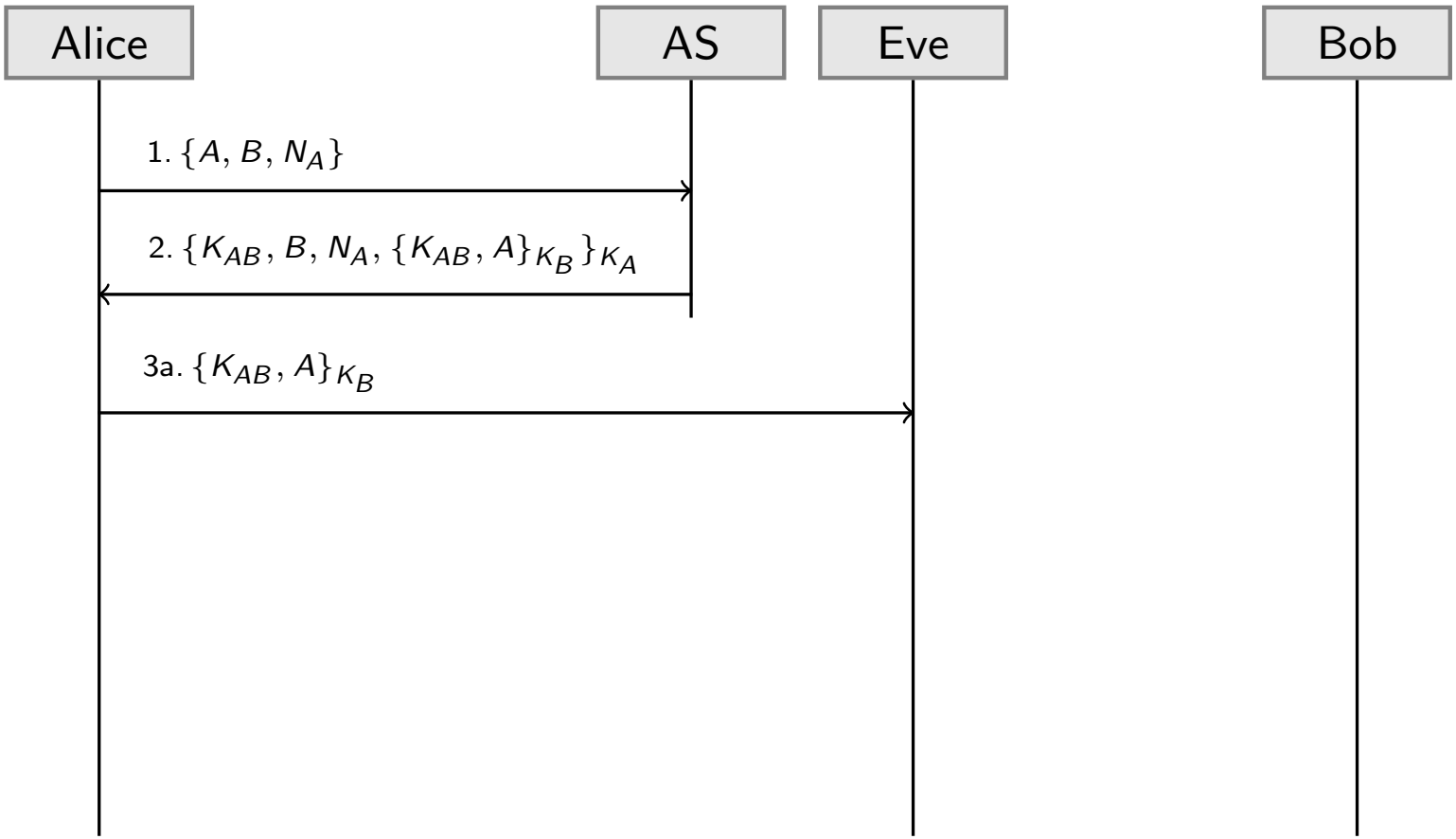
Attack of Variant 4 of the Symmetric NSP



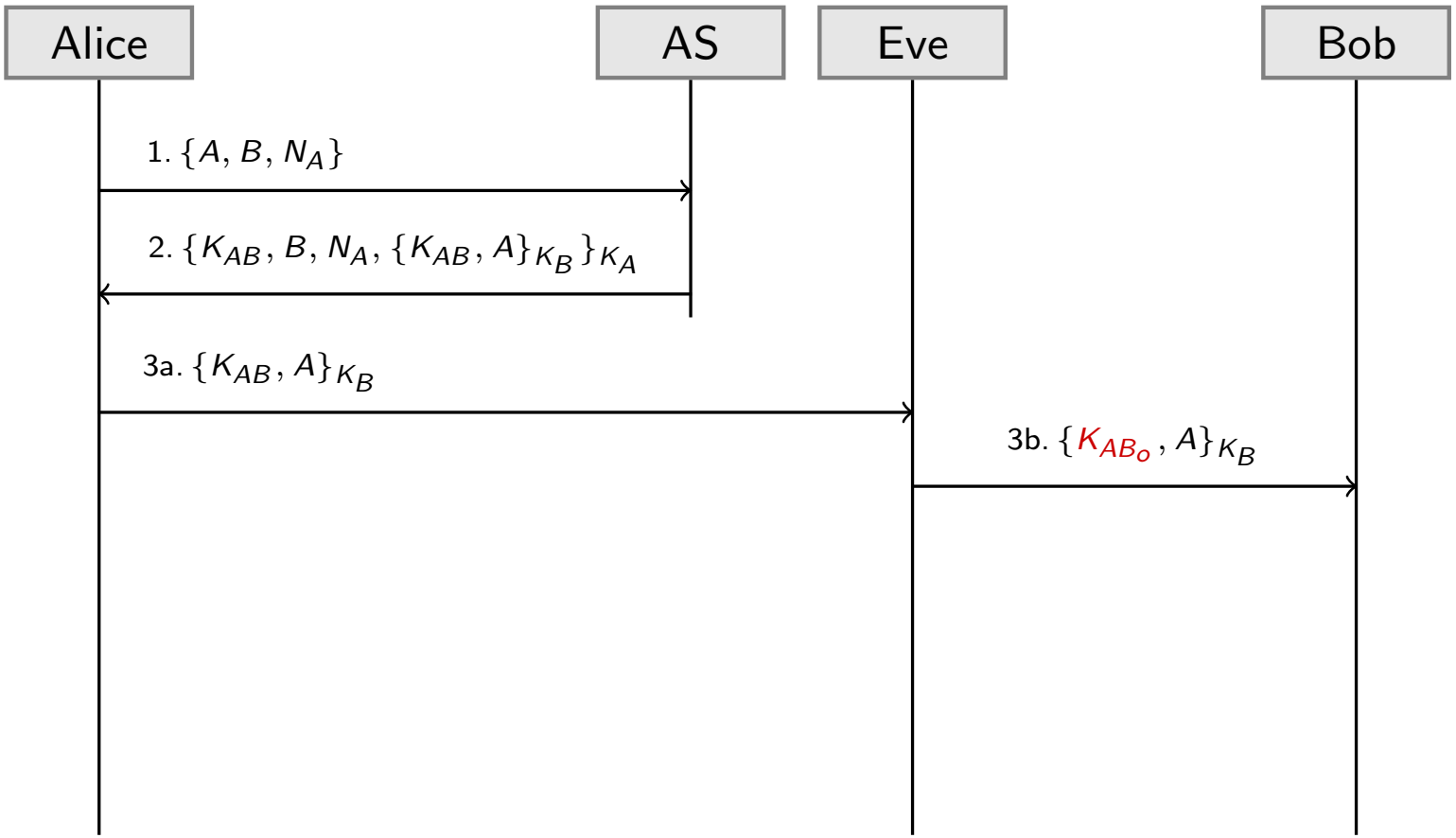
Attack of Variant 4 of the Symmetric NSP



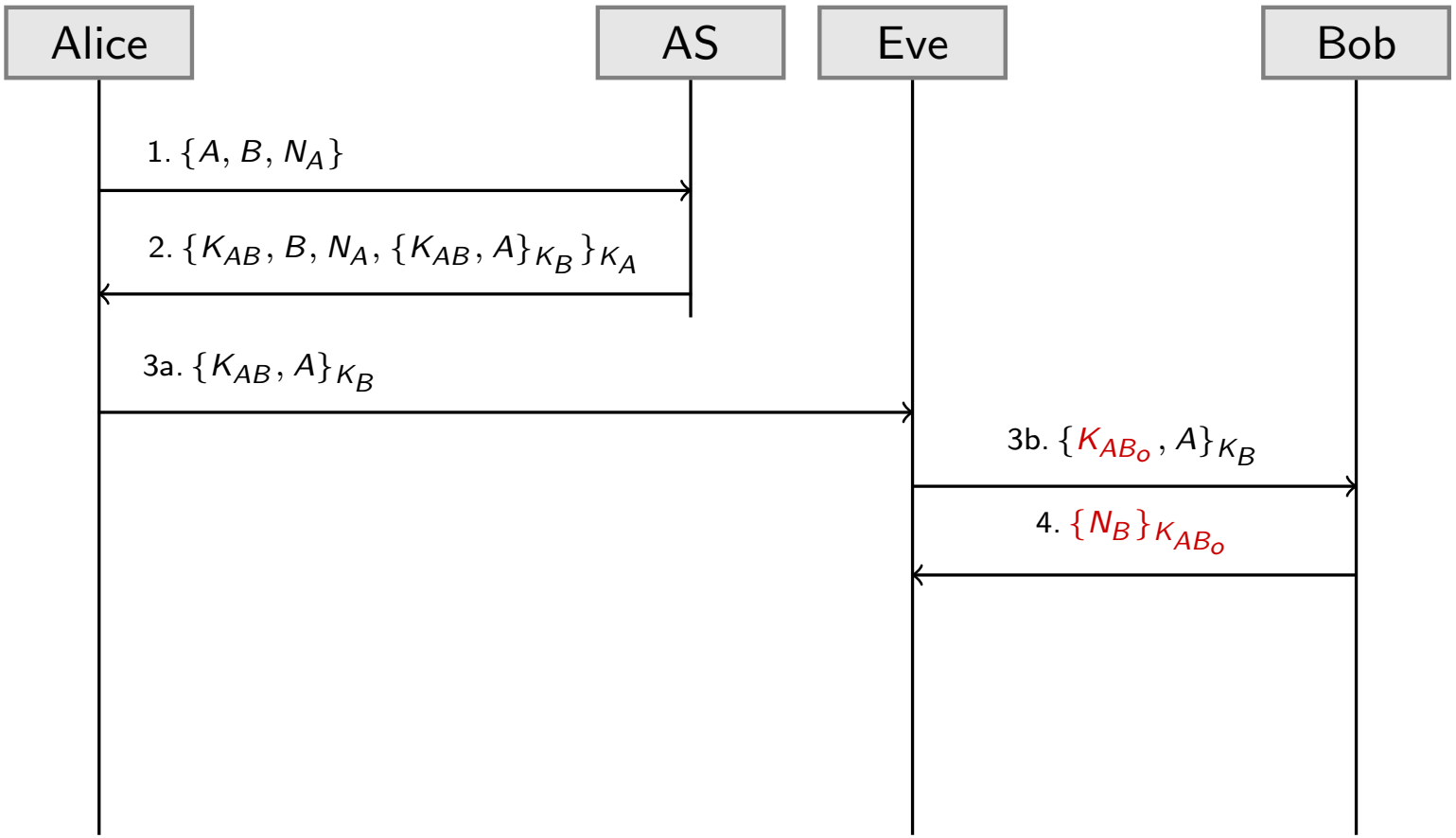
Attack of Variant 4 of the Symmetric NSP



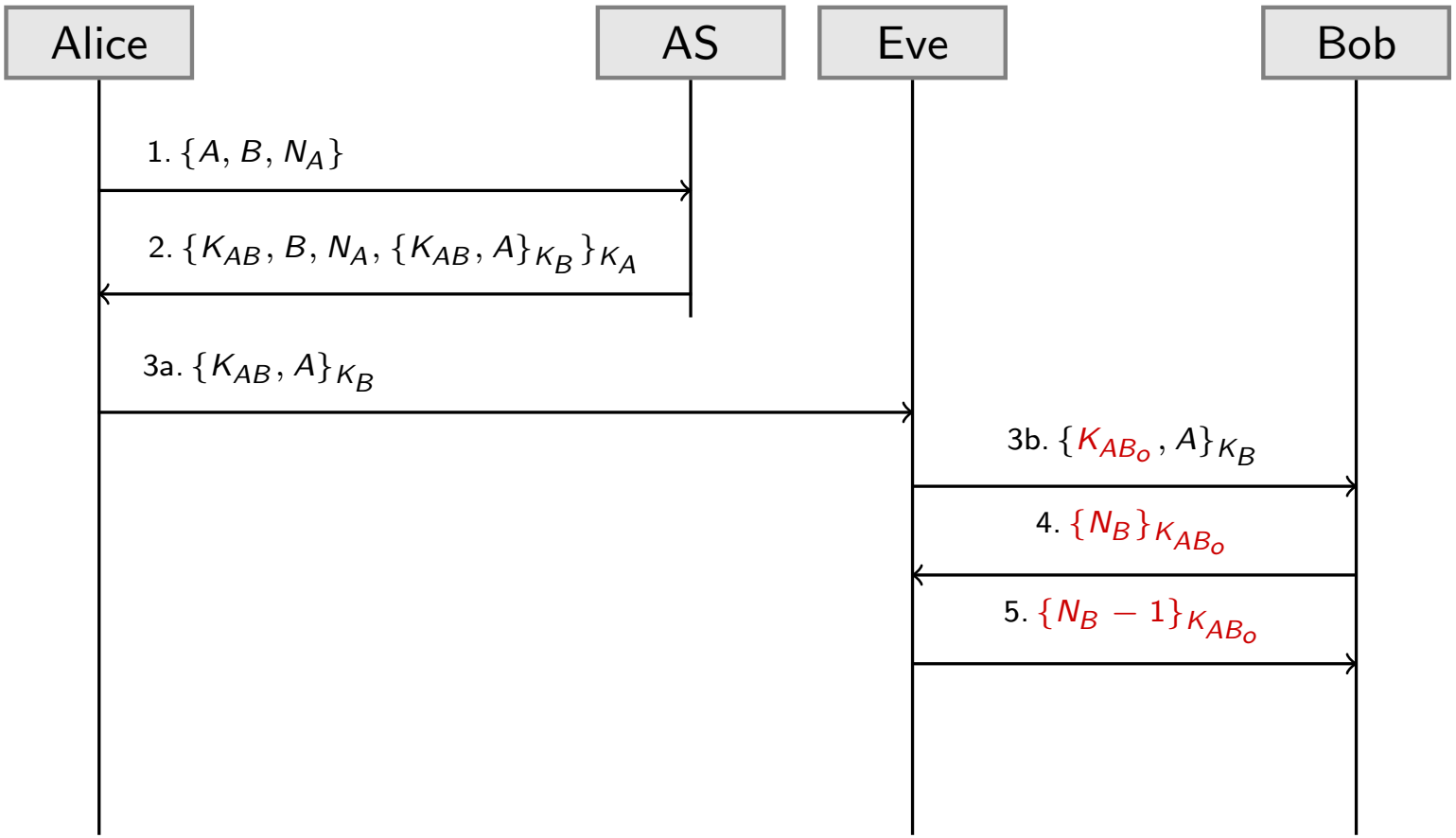
Attack of Variant 4 of the Symmetric NSP



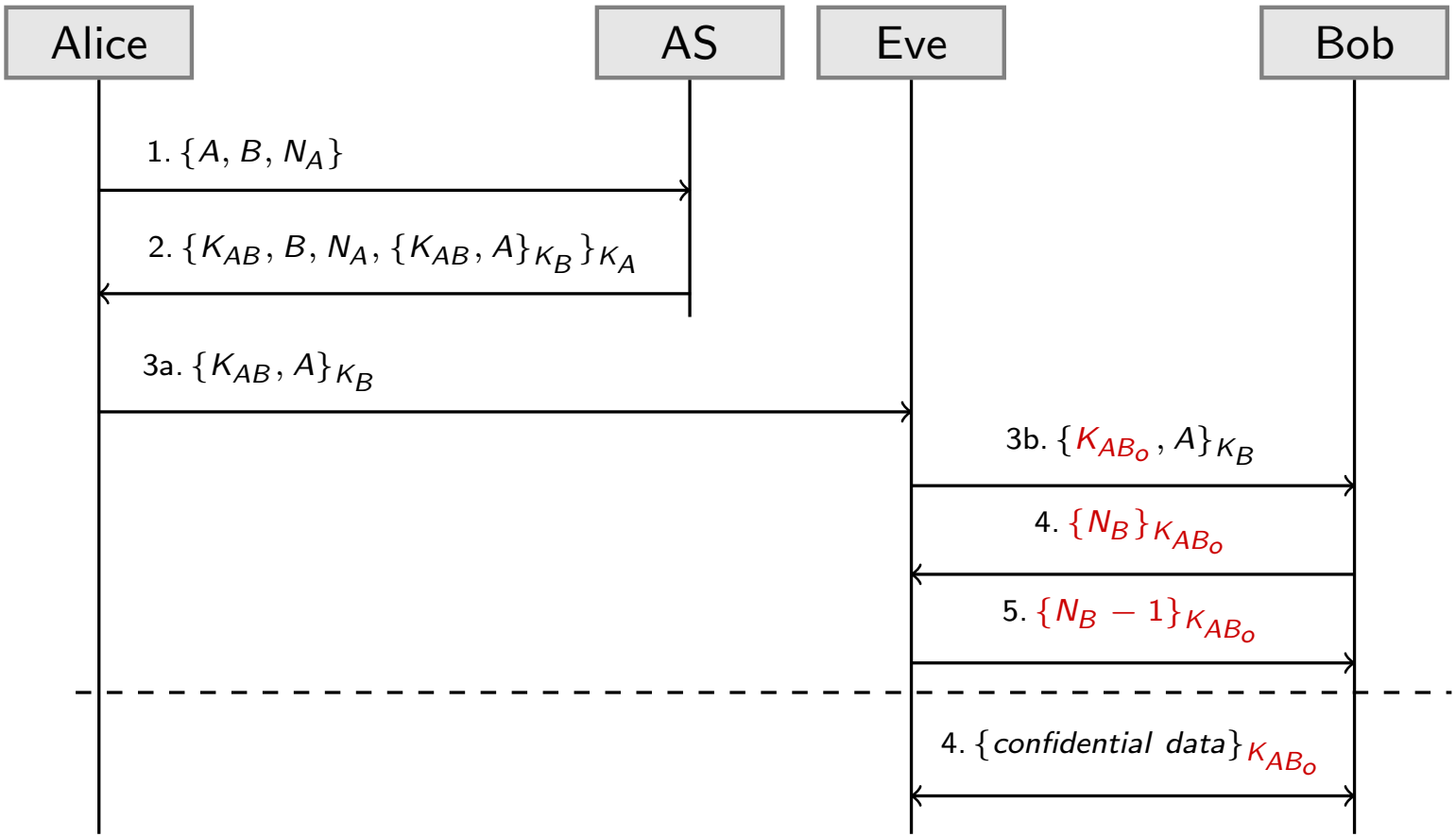
Attack of Variant 4 of the Symmetric NSP



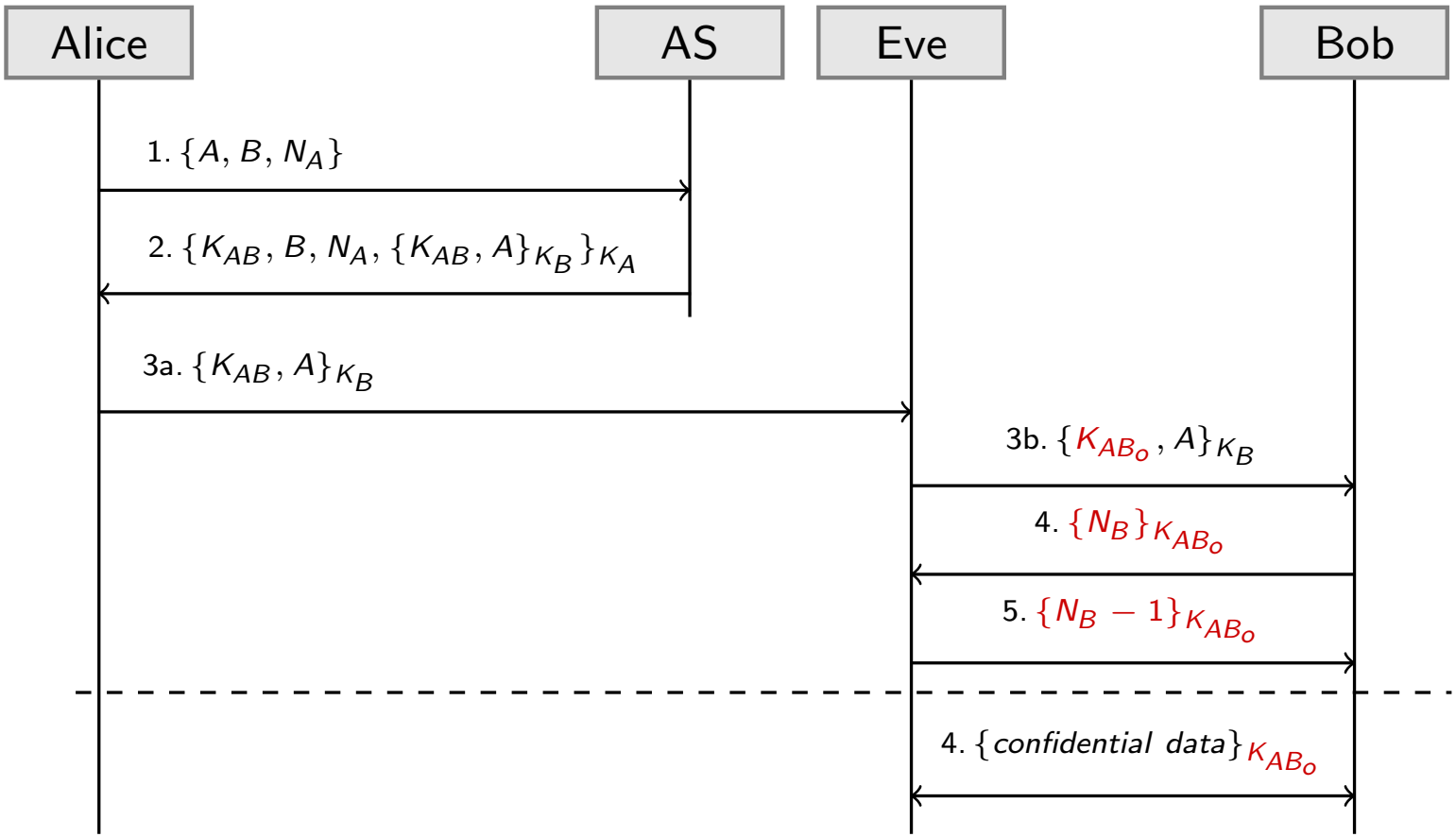
Attack of Variant 4 of the Symmetric NSP



Attack of Variant 4 of the Symmetric NSP

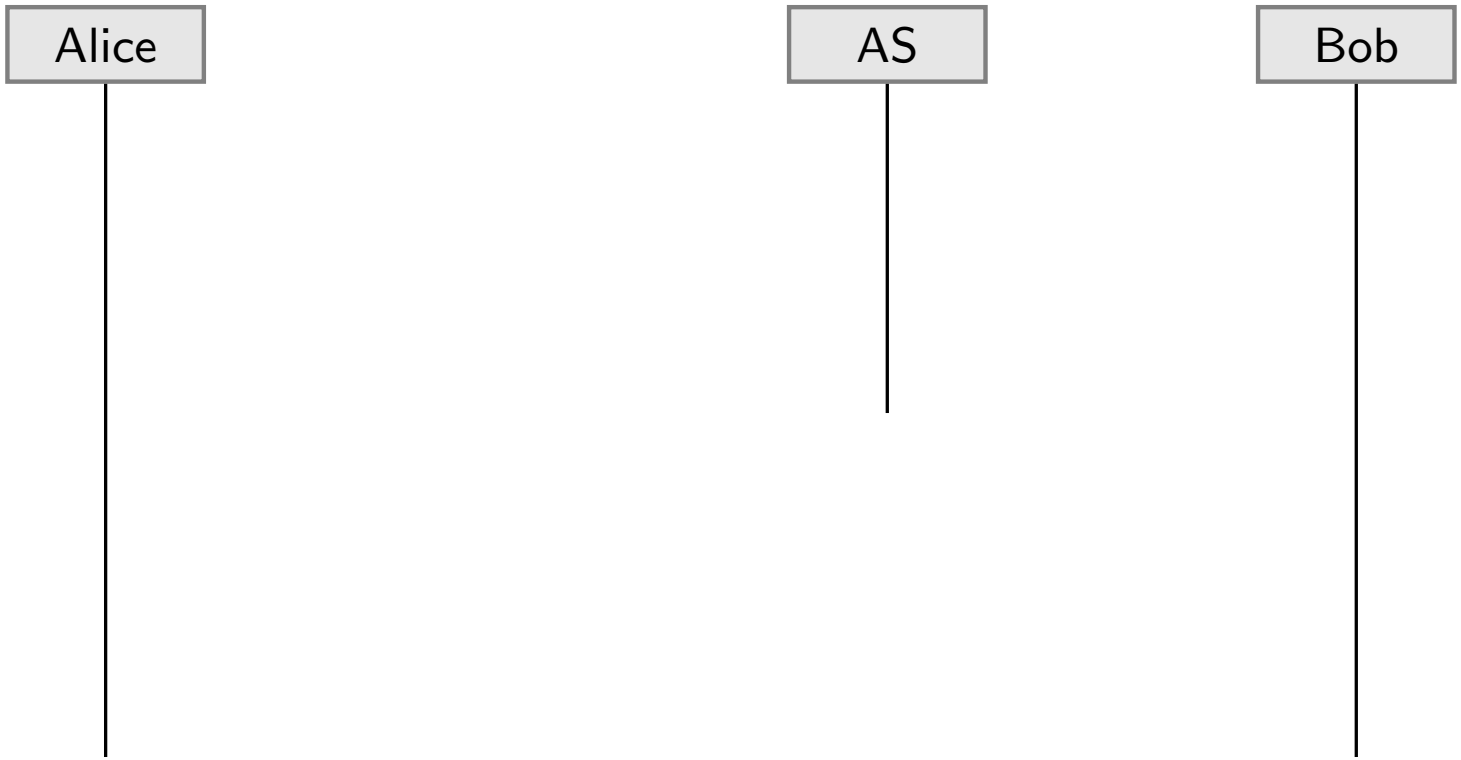


Attack of Variant 4 of the Symmetric NSP

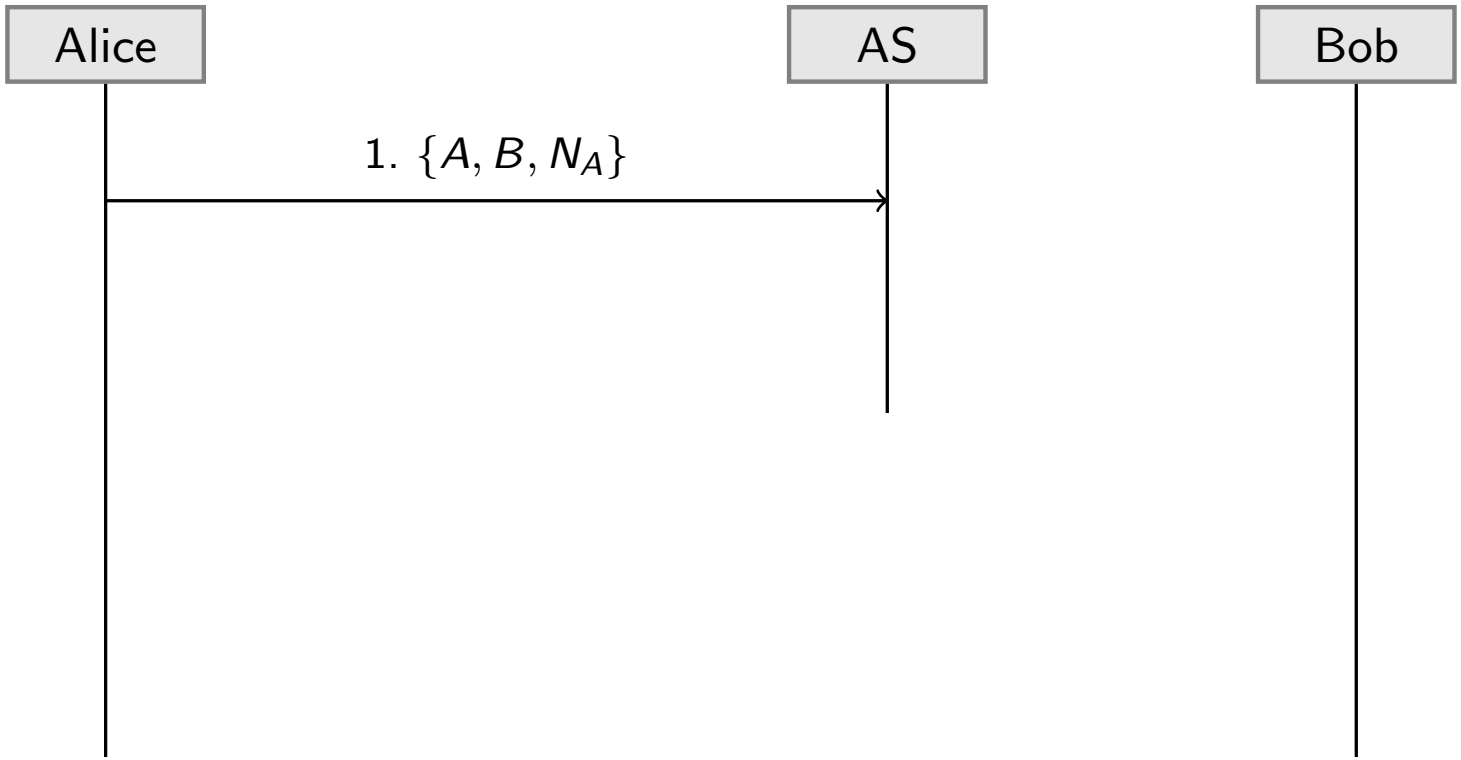


How to uncover the replay attack against Bob?

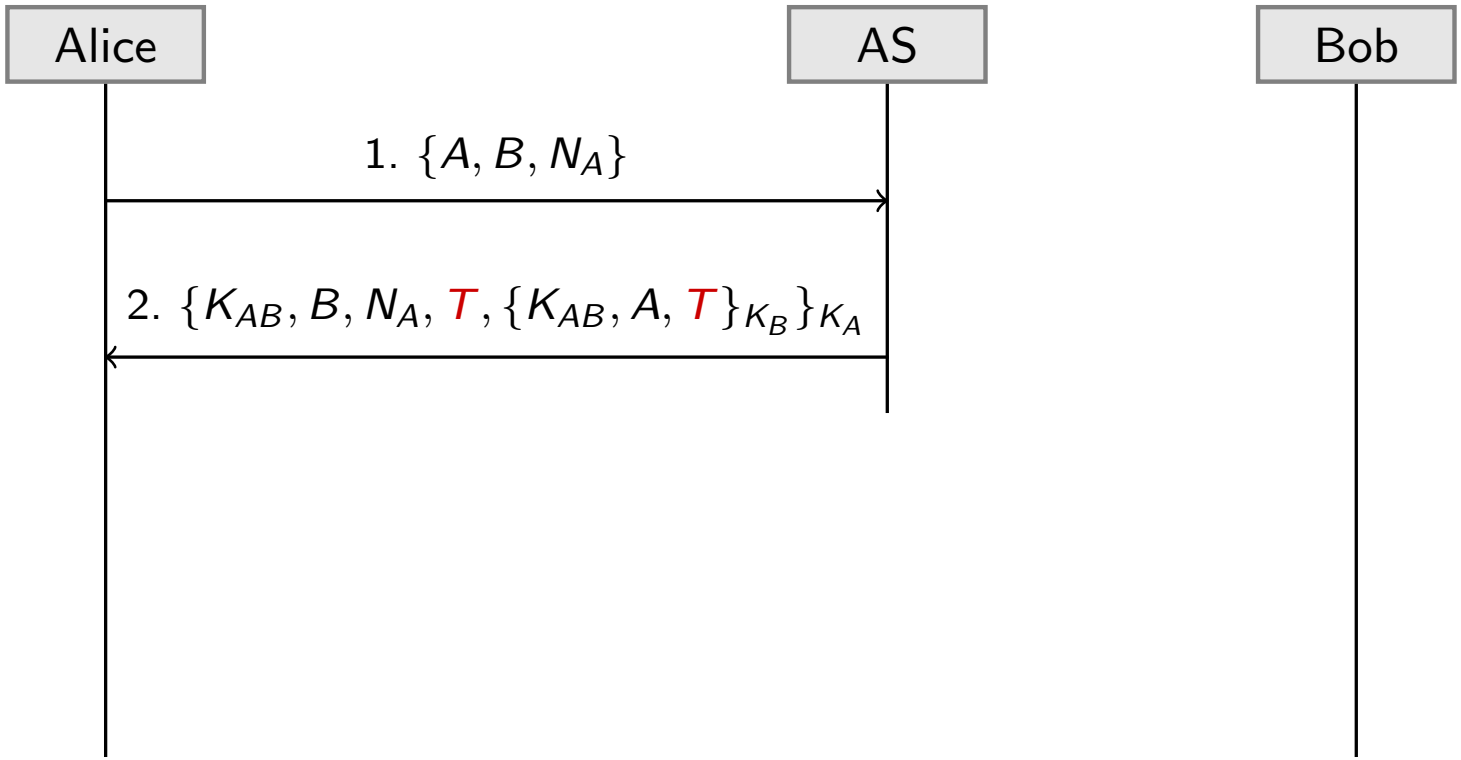
Variant 5: Symmetric NSP & Time Stamps



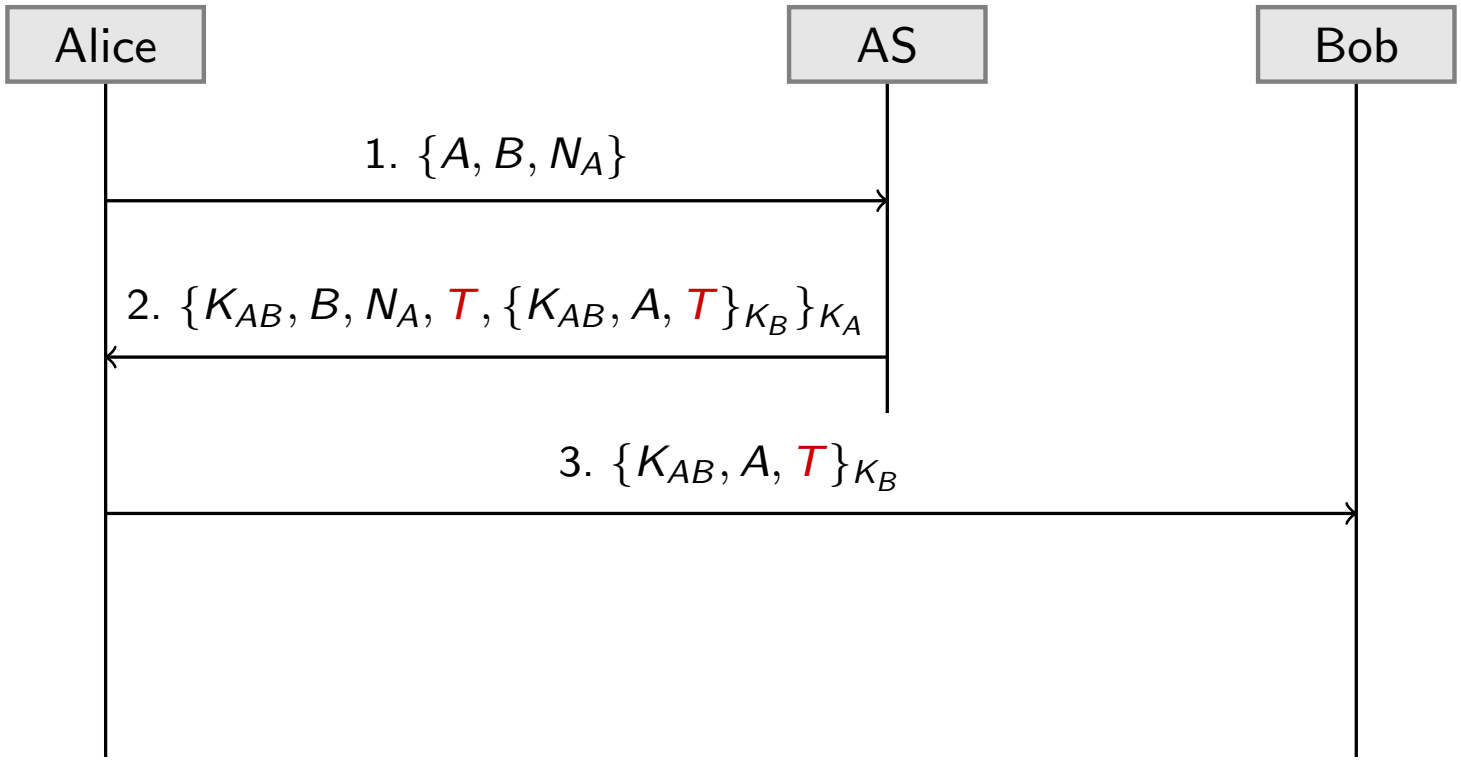
Variant 5: Symmetric NSP & Time Stamps



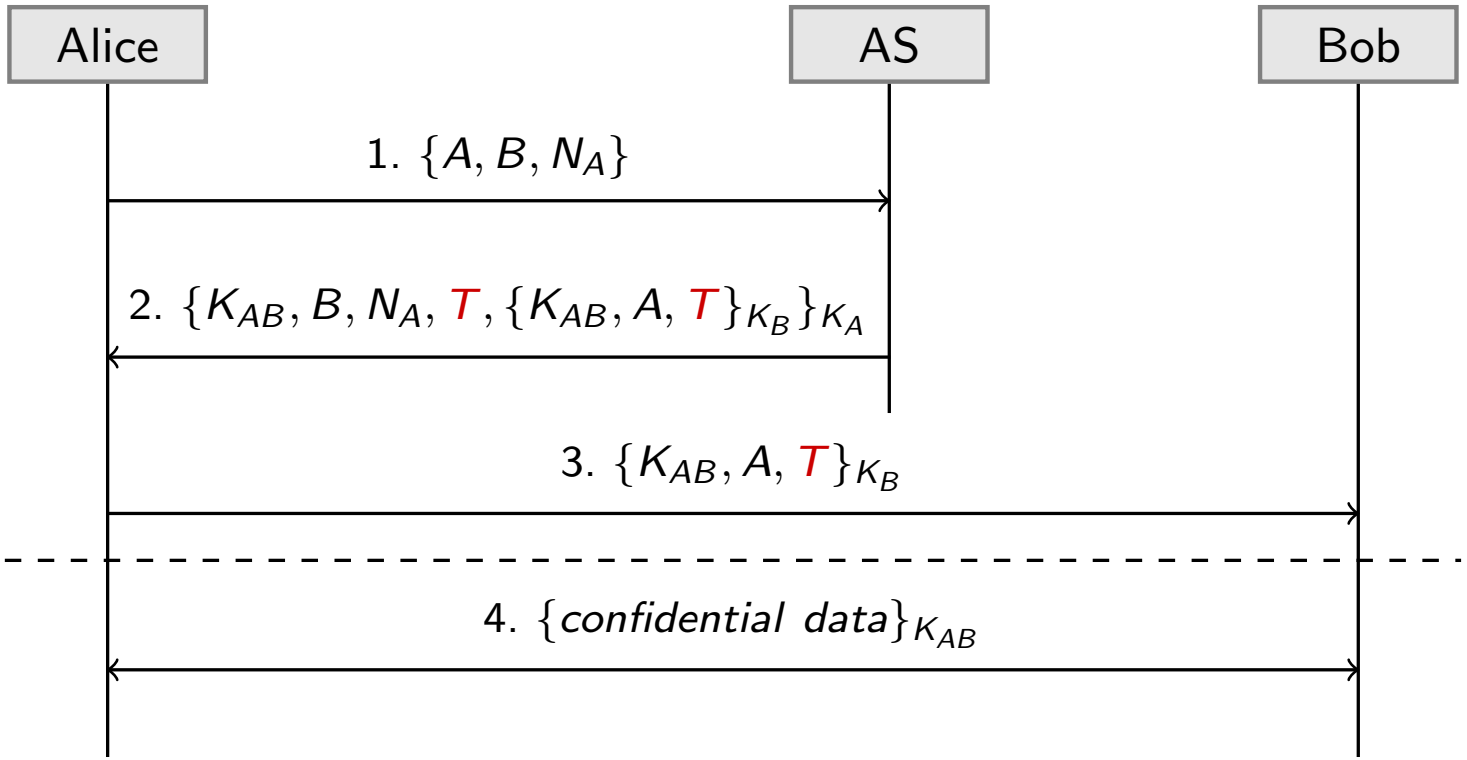
Variant 5: Symmetric NSP & Time Stamps



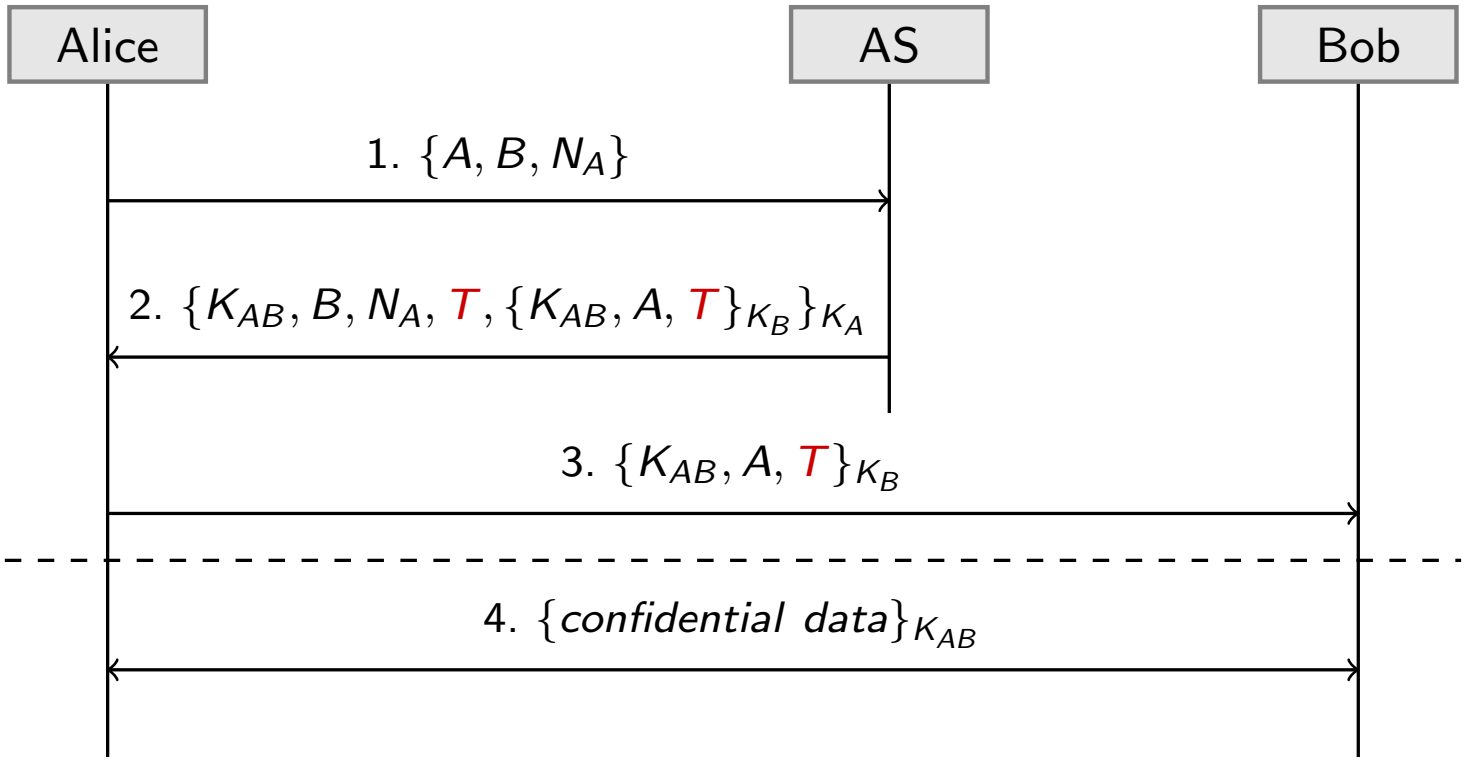
Variant 5: Symmetric NSP & Time Stamps



Variant 5: Symmetric NSP & Time Stamps

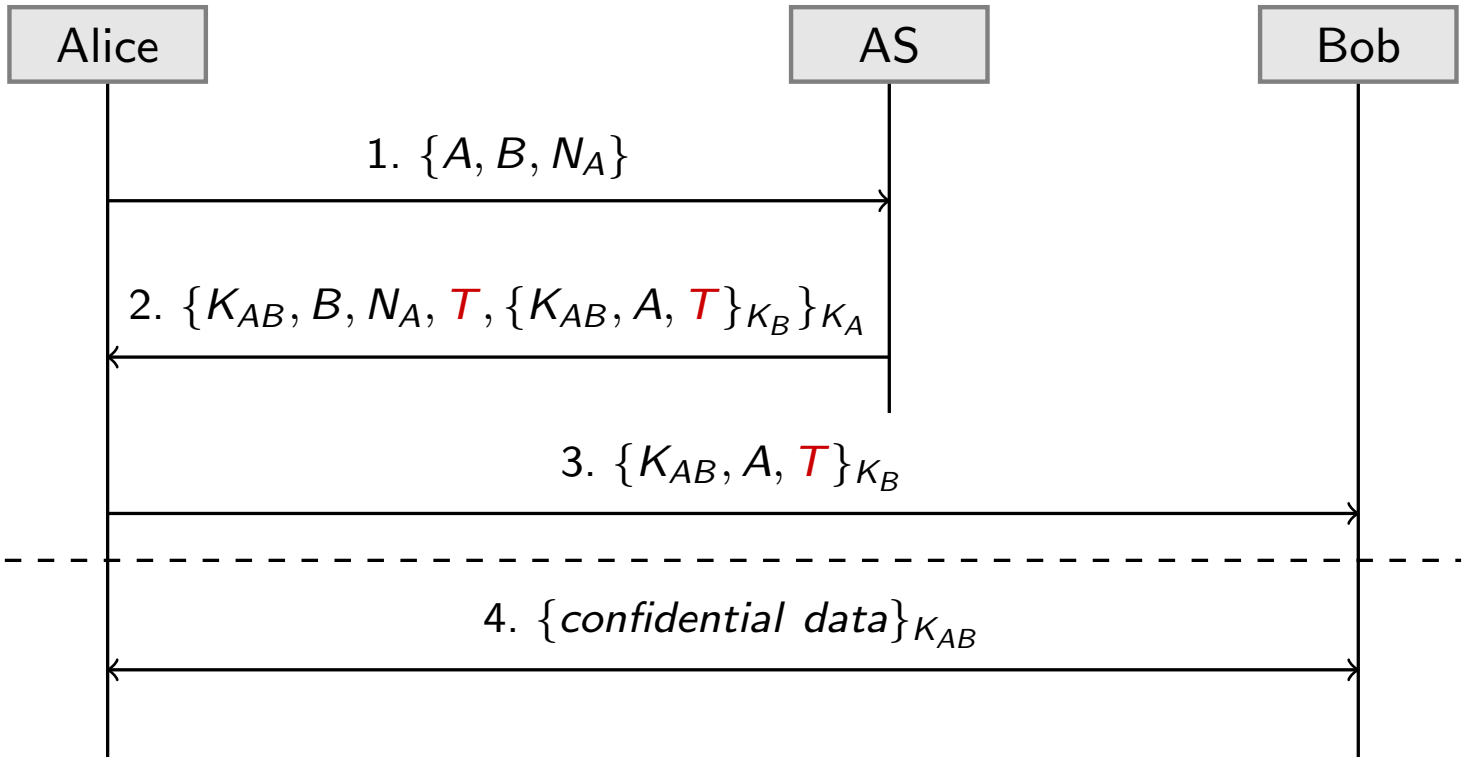


Variant 5: Symmetric NSP & Time Stamps



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

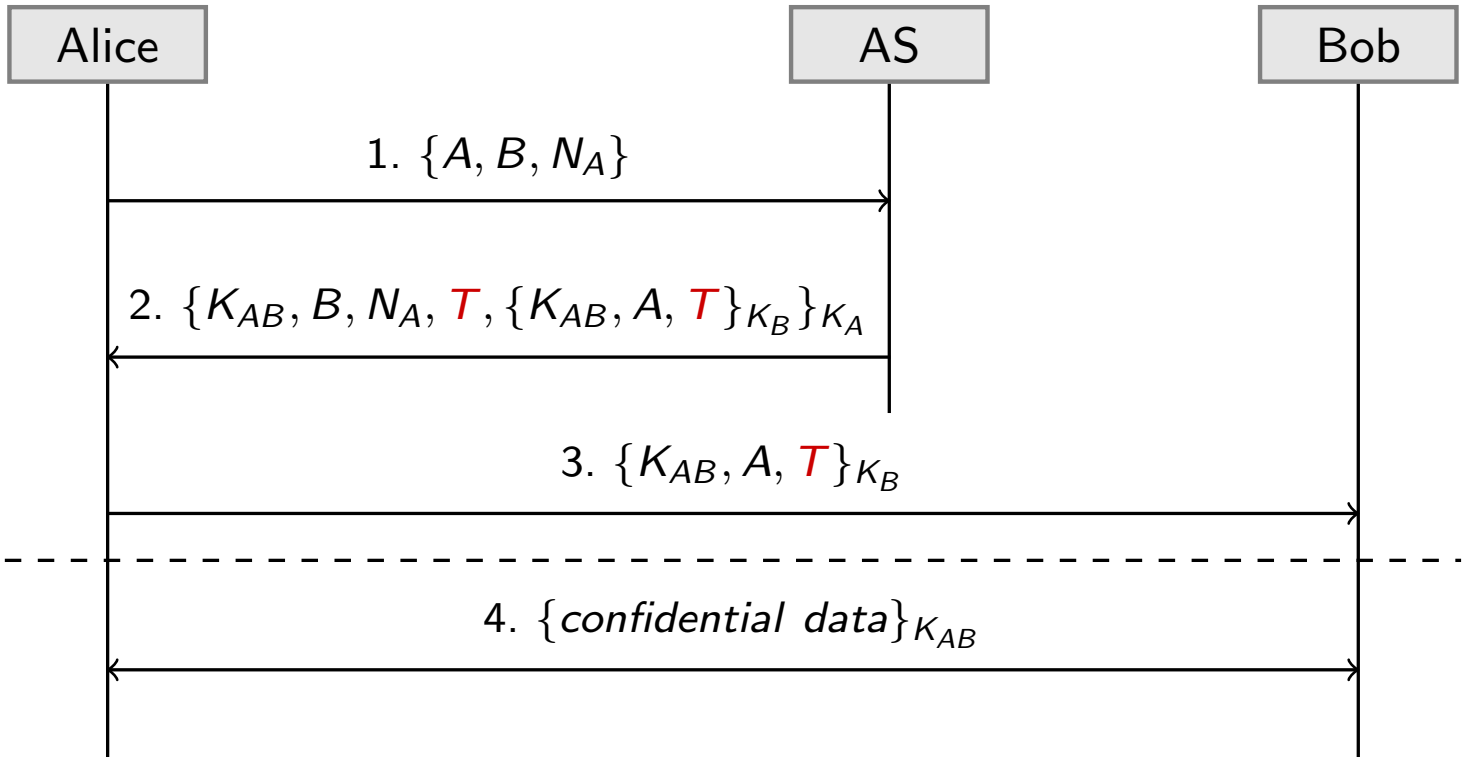
Variant 5: Symmetric NSP & Time Stamps



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

➜ Is Bob protected now?

Variant 5: Symmetric NSP & Time Stamps



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

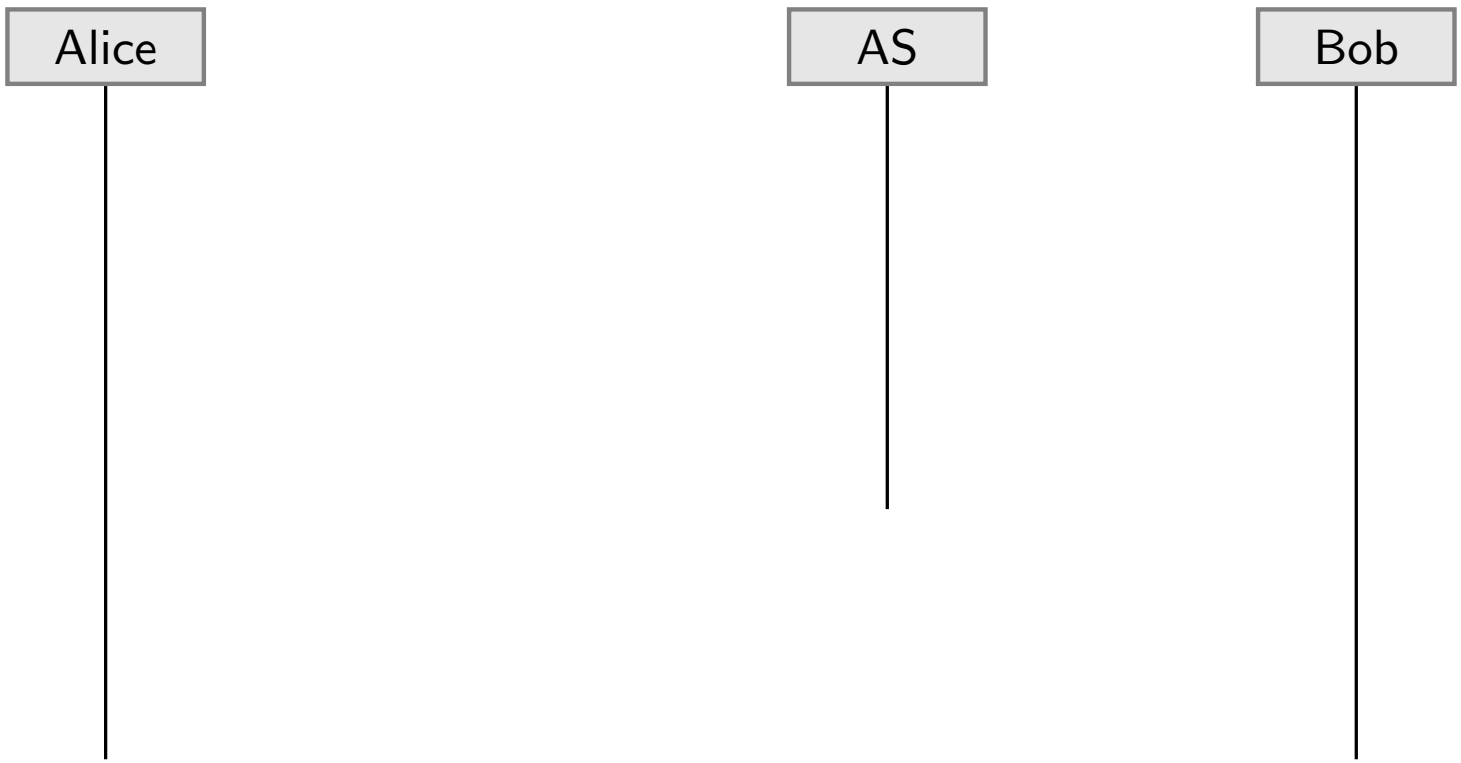
1. Modify the time of your target system
2. Perform a replay attack

How to protect?

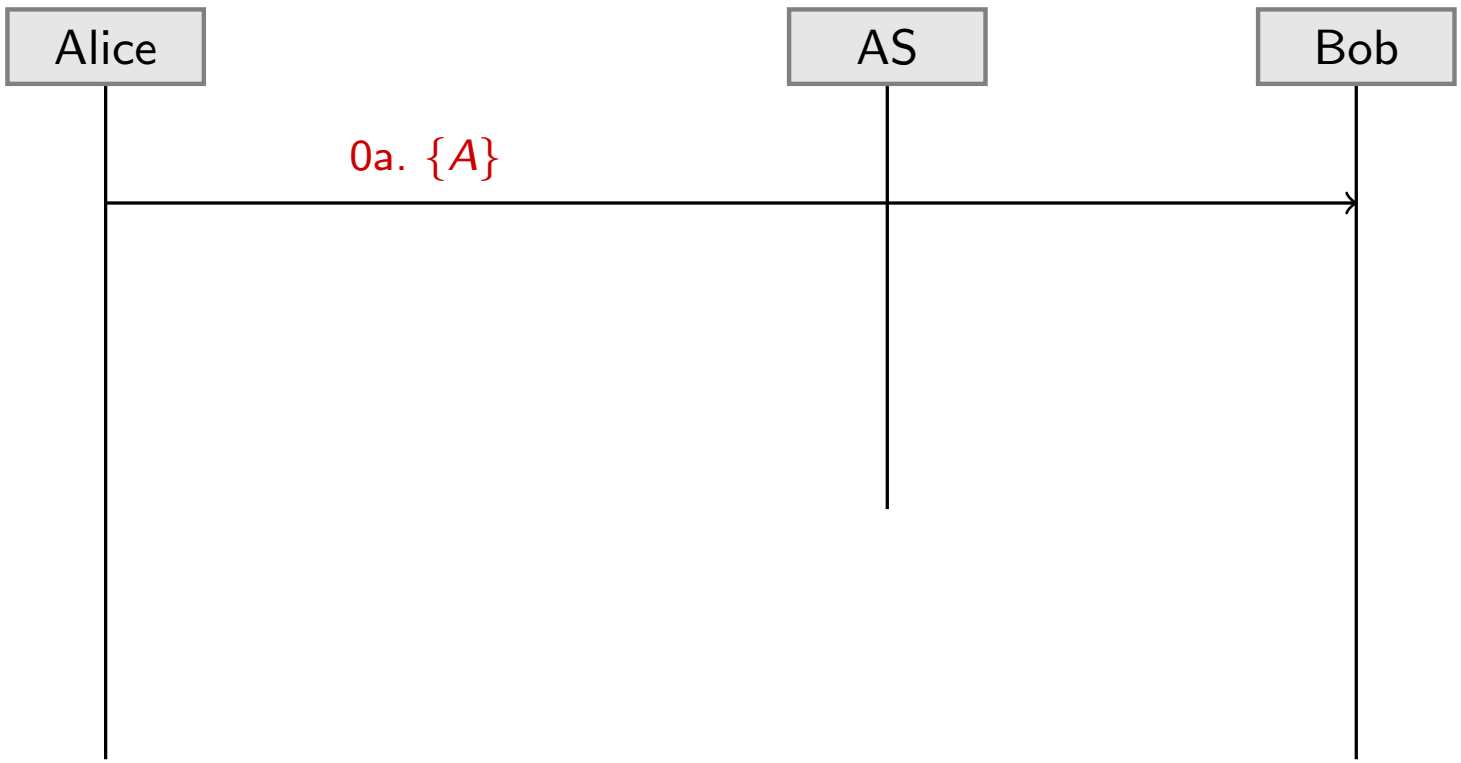
➜ Use of previously negotiated nonces also for Bob

➜ Disadvantage: The protocol is getting more complicated

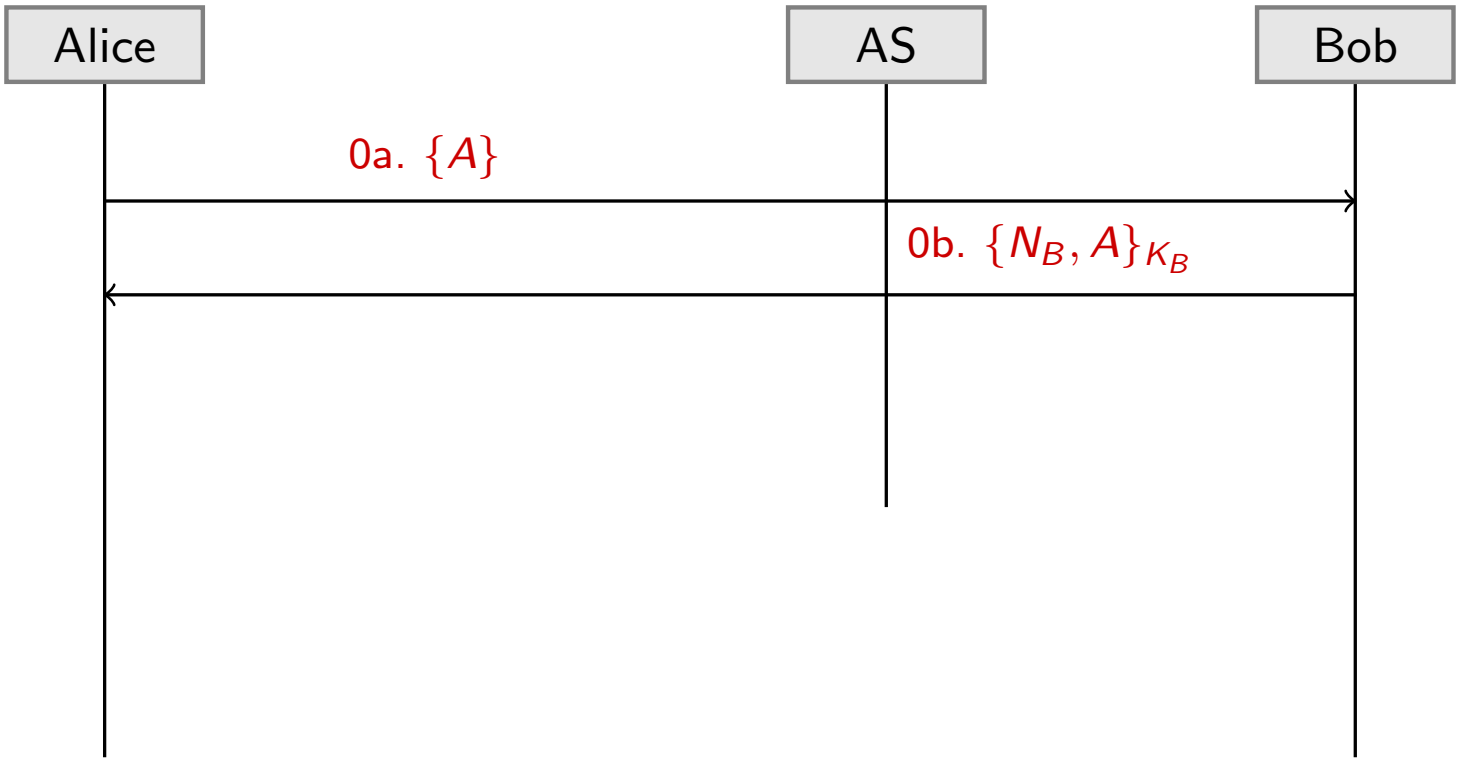
Variant 6 of the Symmetric NSP with Nonces



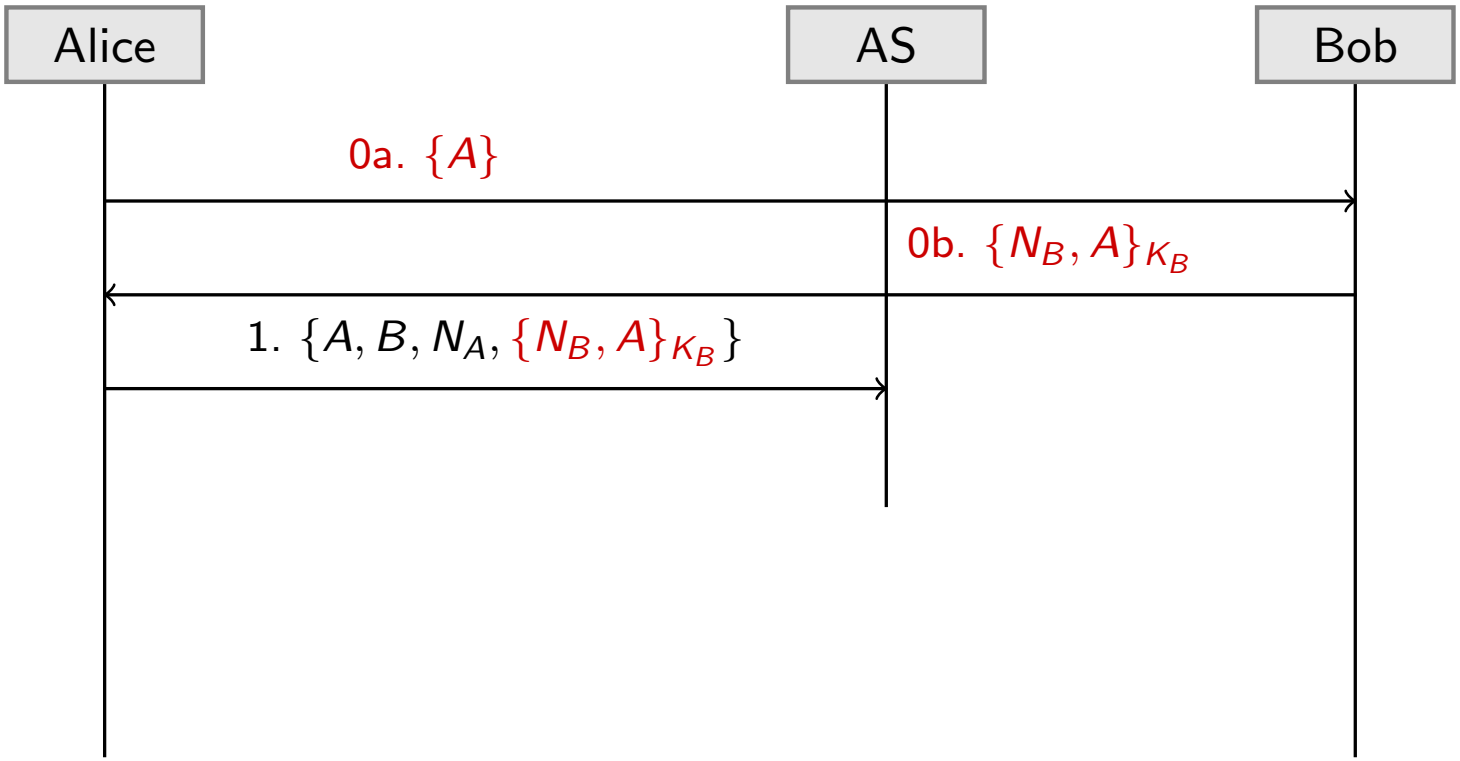
Variant 6 of the Symmetric NSP with Nonces



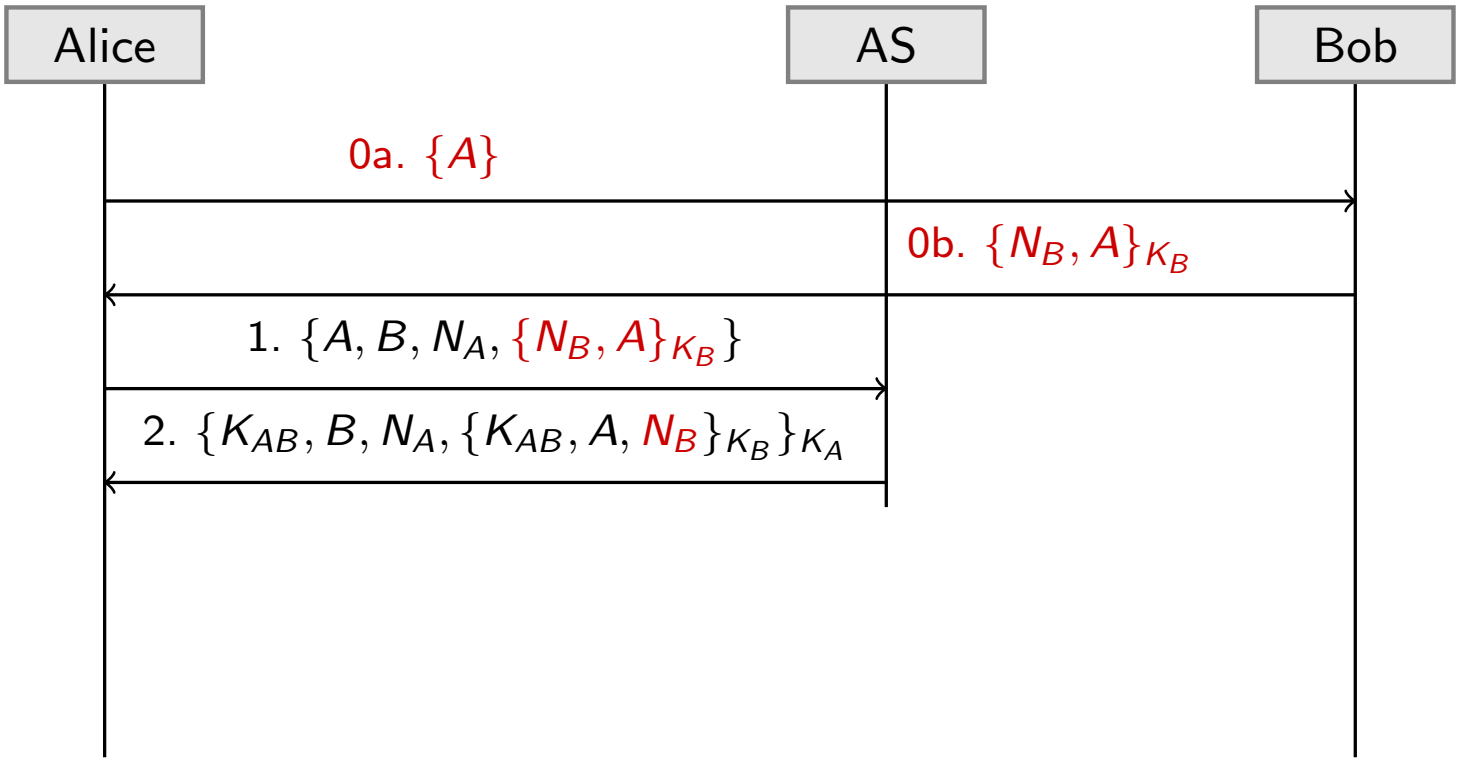
Variant 6 of the Symmetric NSP with Nonces



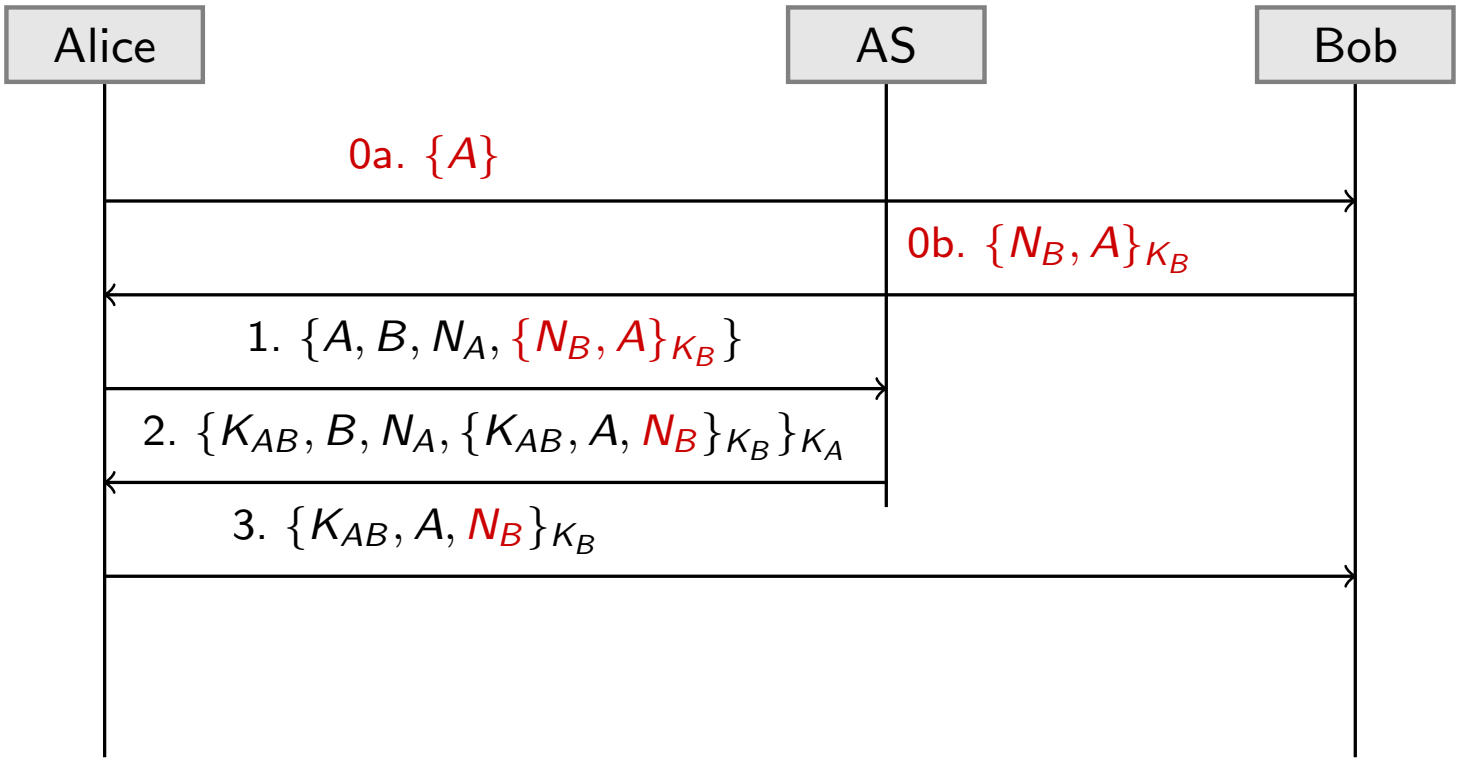
Variant 6 of the Symmetric NSP with Nonces



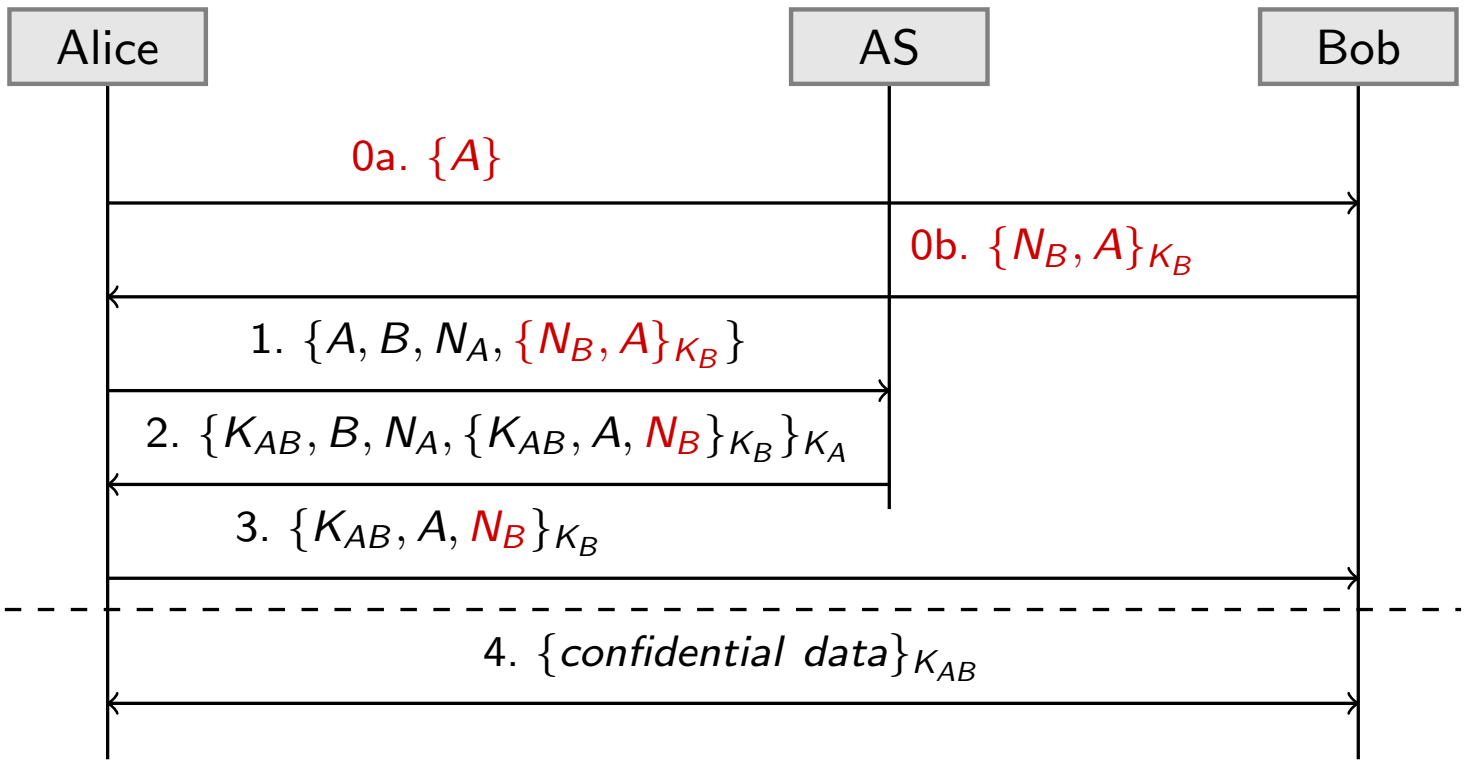
Variant 6 of the Symmetric NSP with Nonces



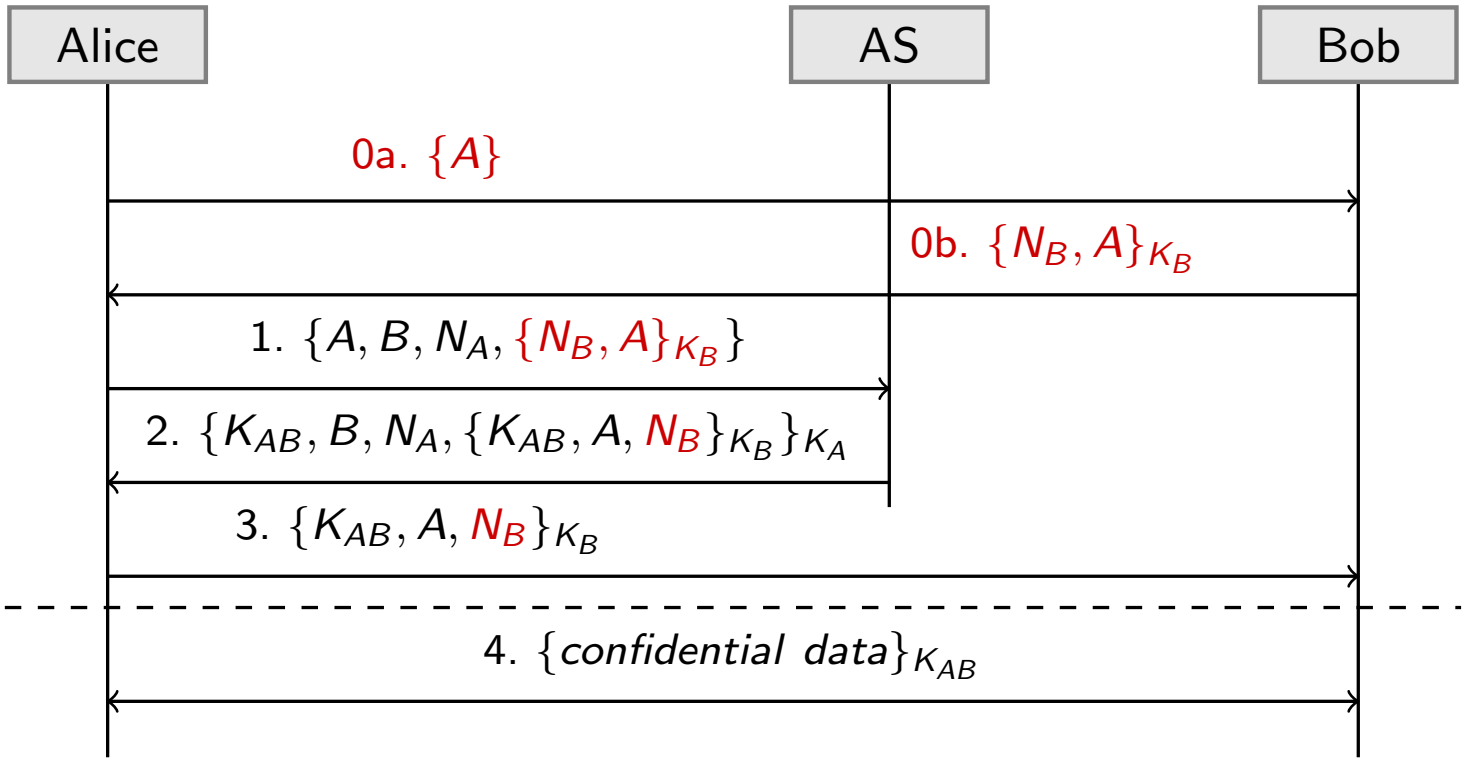
Variant 6 of the Symmetric NSP with Nonces



Variant 6 of the Symmetric NSP with Nonces



Variant 6 of the Symmetric NSP with Nonces



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

Which countermeasures exist to prevent these 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

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

Preliminary Specifications

Given Keys

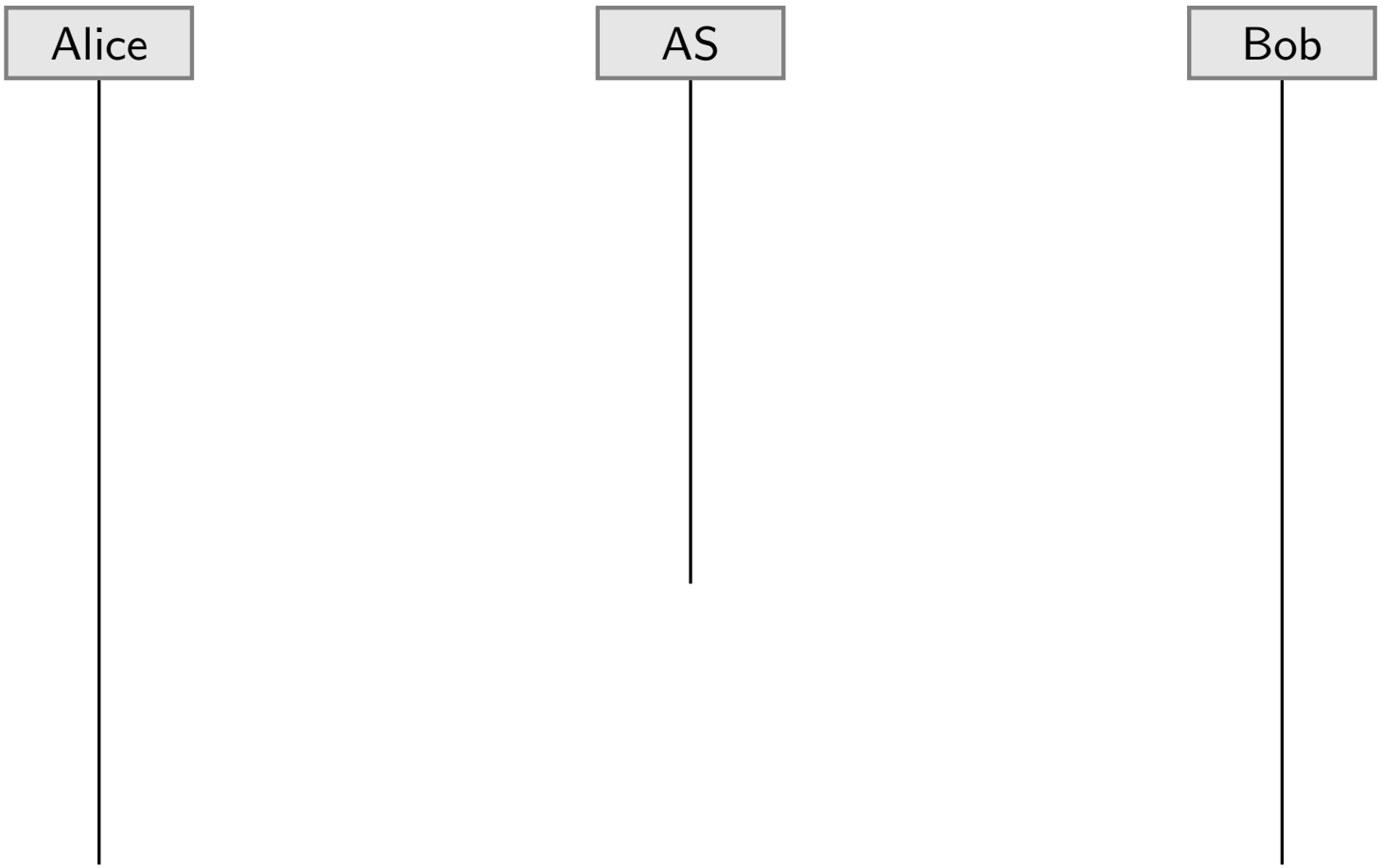
1. PK_{AS}: Public key of the authentication server AS
2. SK_{AS}: Secret key of the authentication server AS
3. PK_A and PK_B: Public keys of Alice and Bob
4. SK_A and SK_B: Secret keys of Alice and Bob

Assumptions

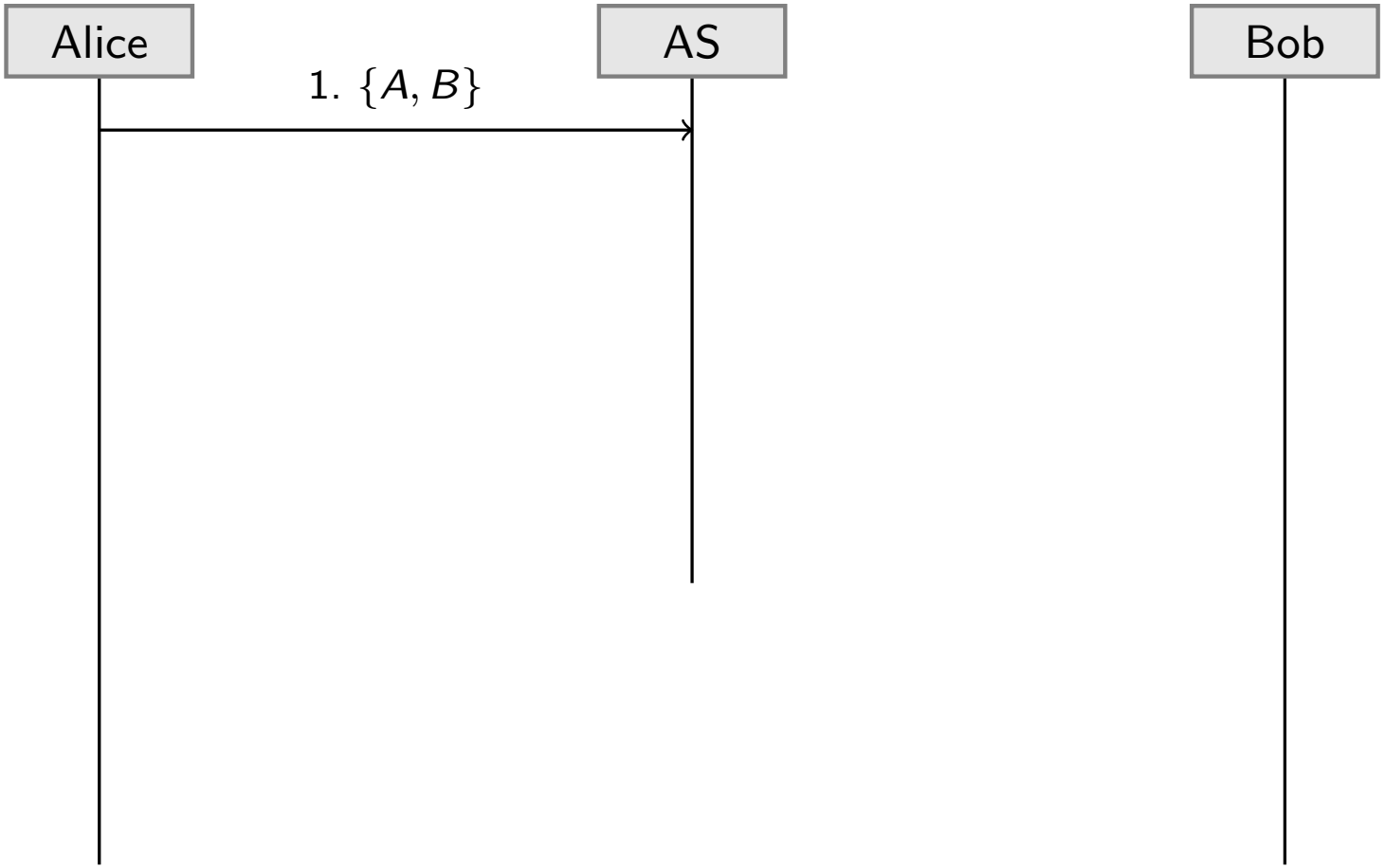
- AS knows the public keys of all participants
- All participants only know the public key PK_{AS} before the protocol is started

➜ Participants must request all other required keys from AS

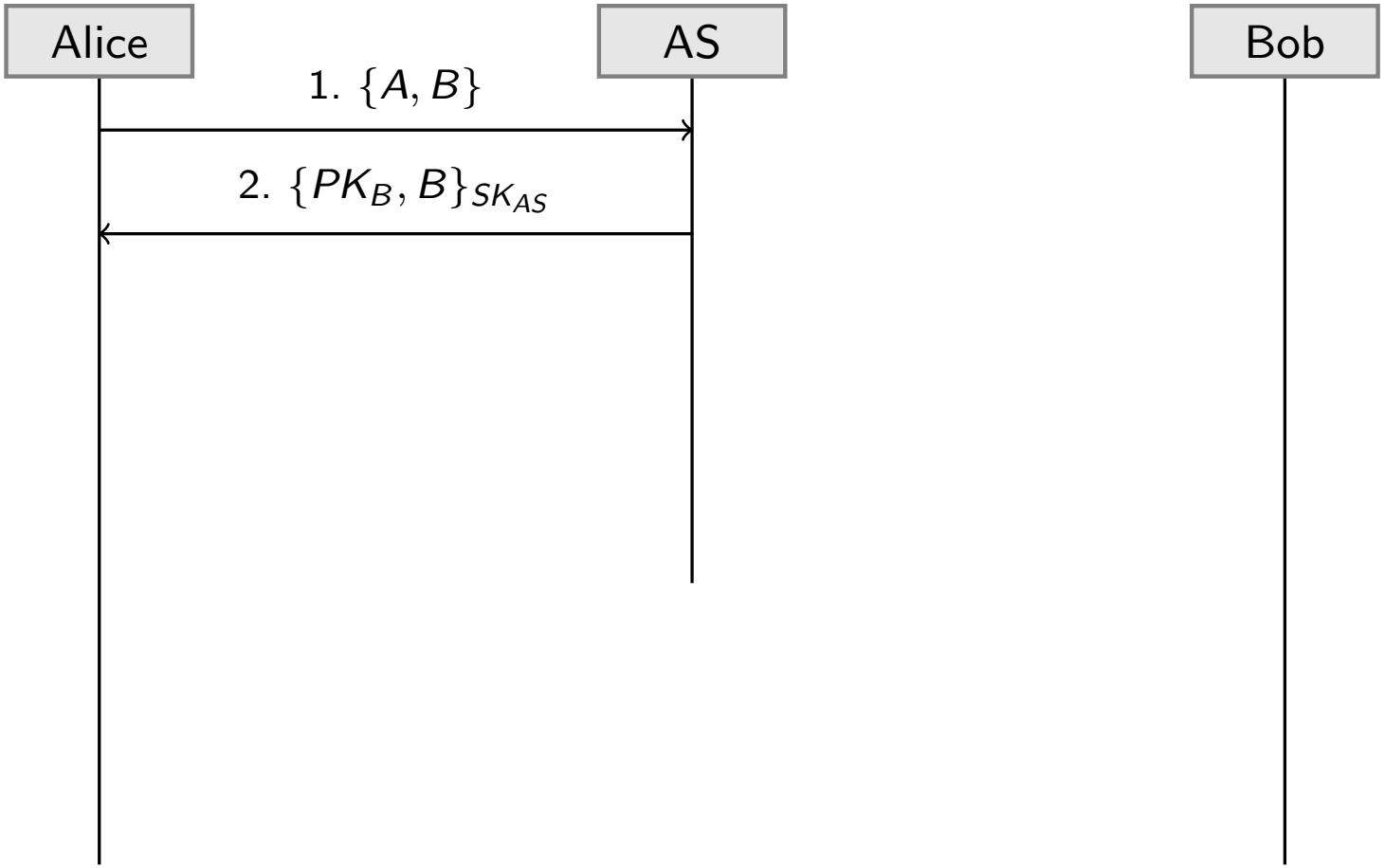
Asymmetric Variant of the NSP



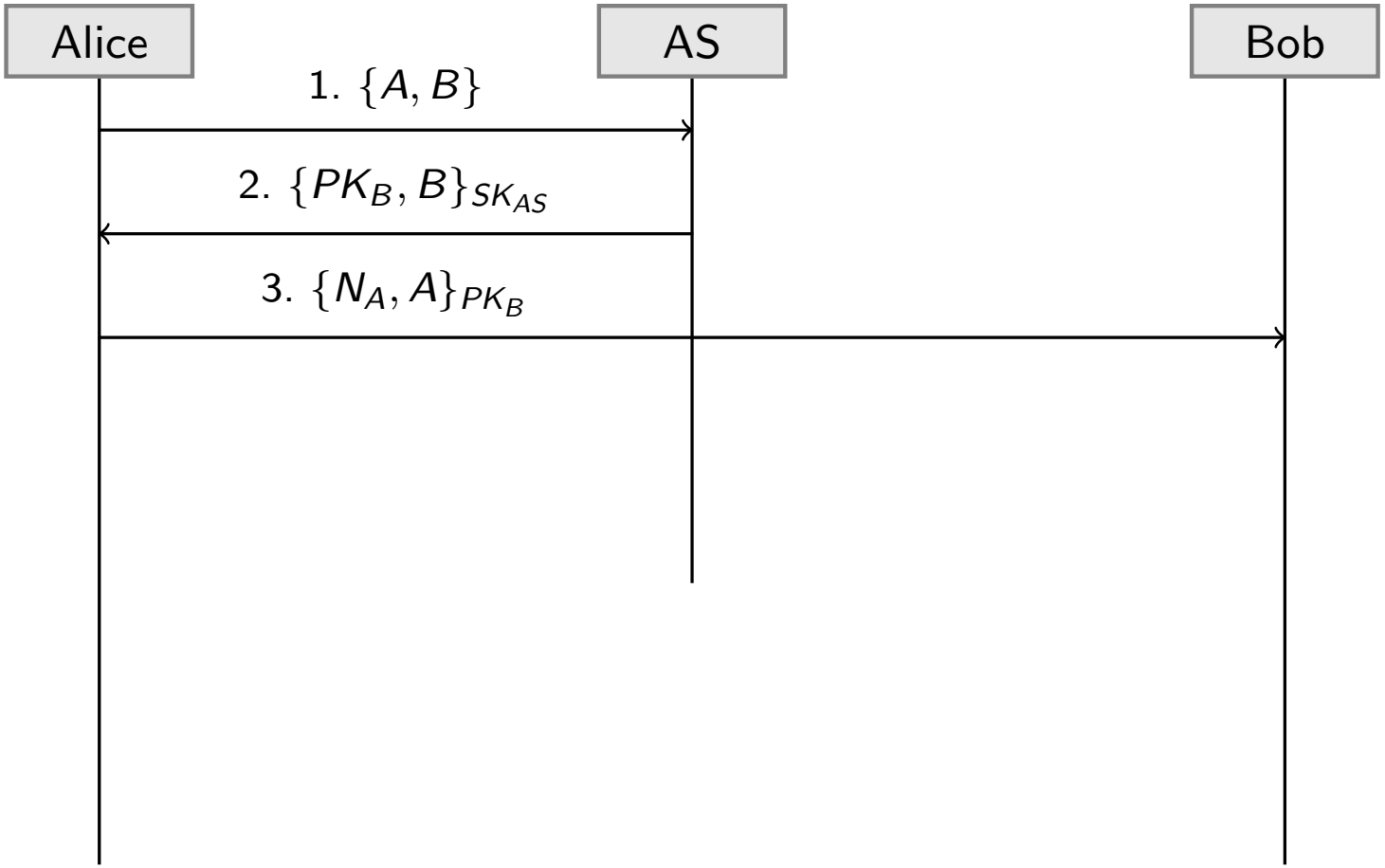
Asymmetric Variant of the NSP



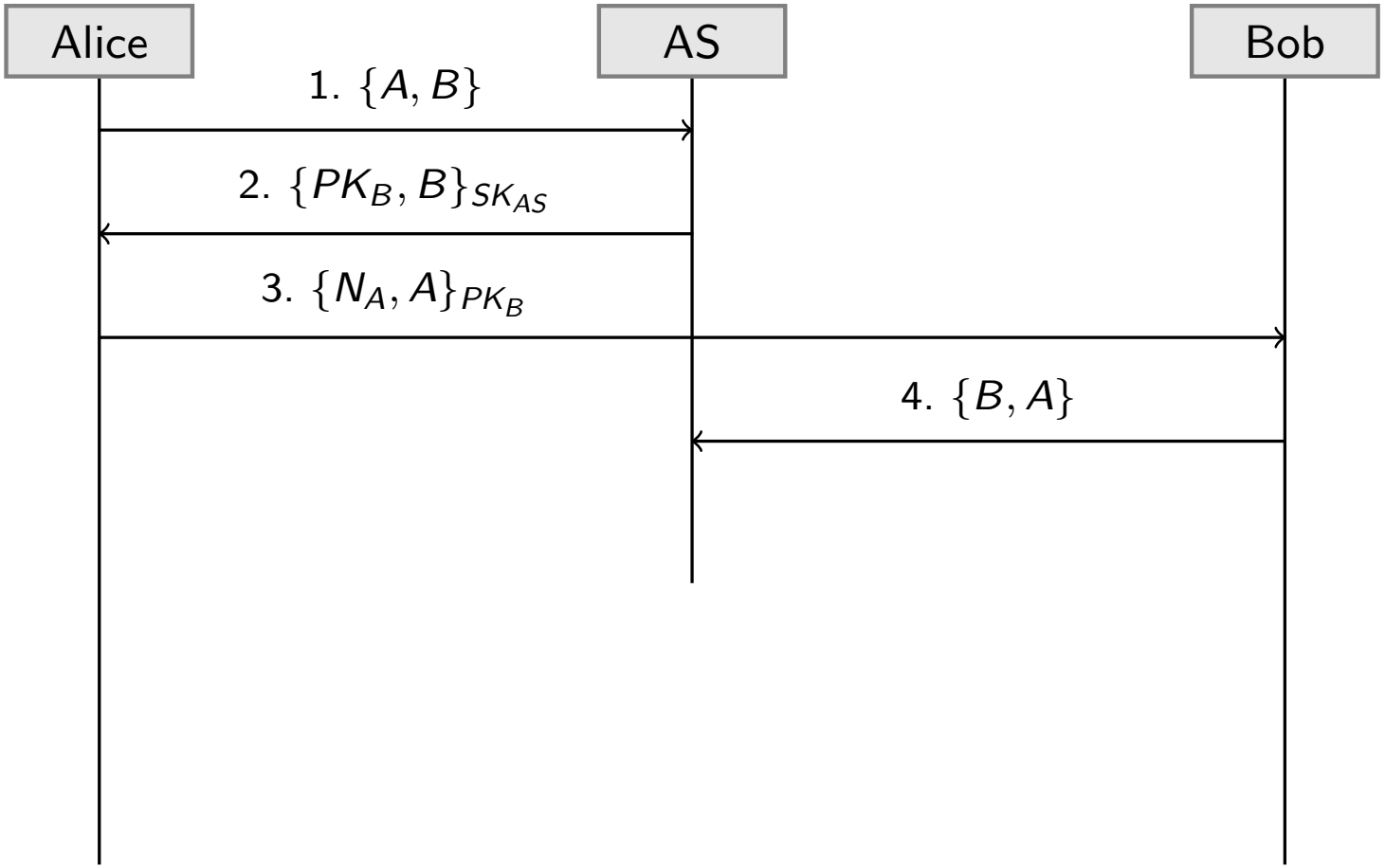
Asymmetric Variant of the NSP



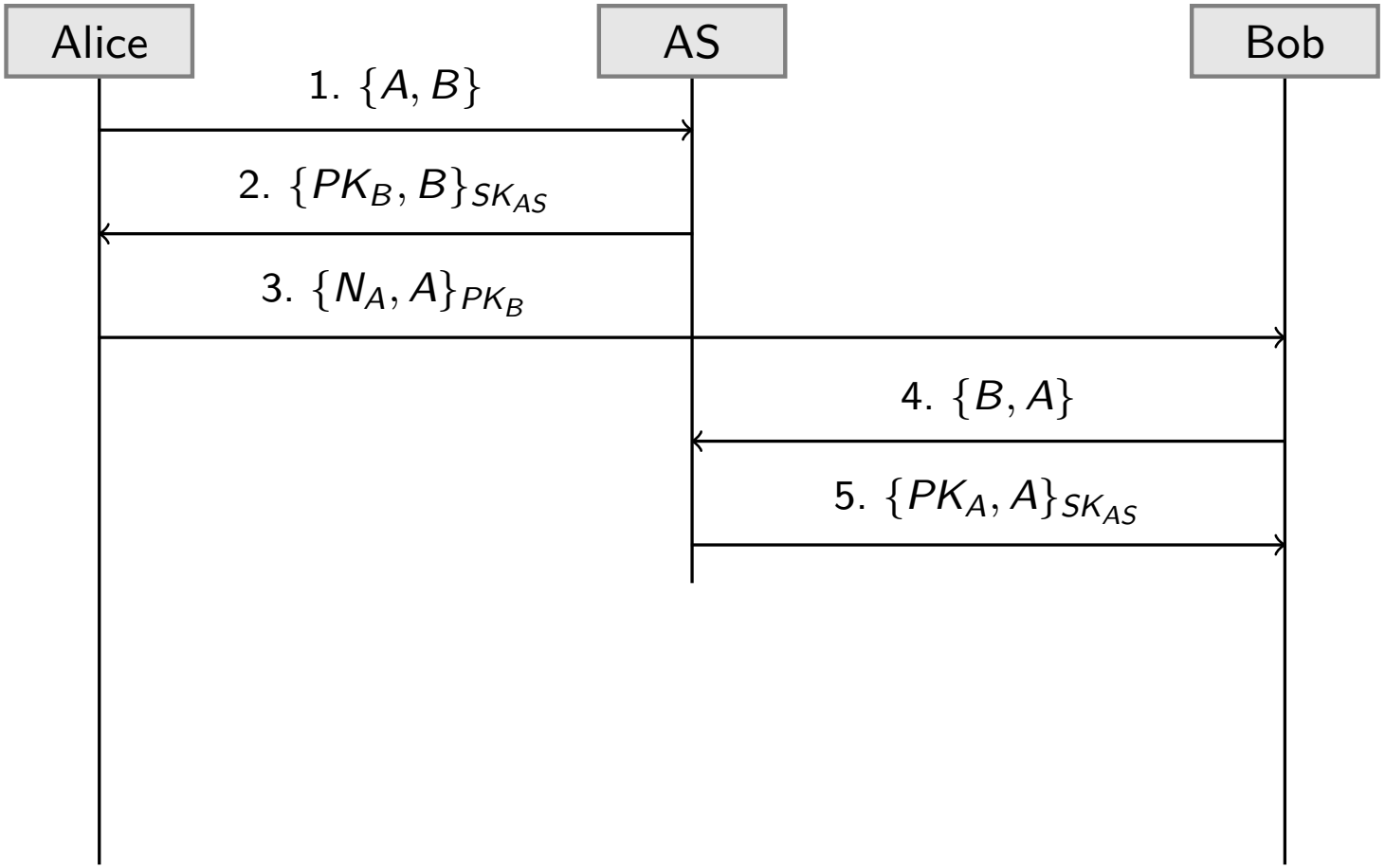
Asymmetric Variant of the NSP



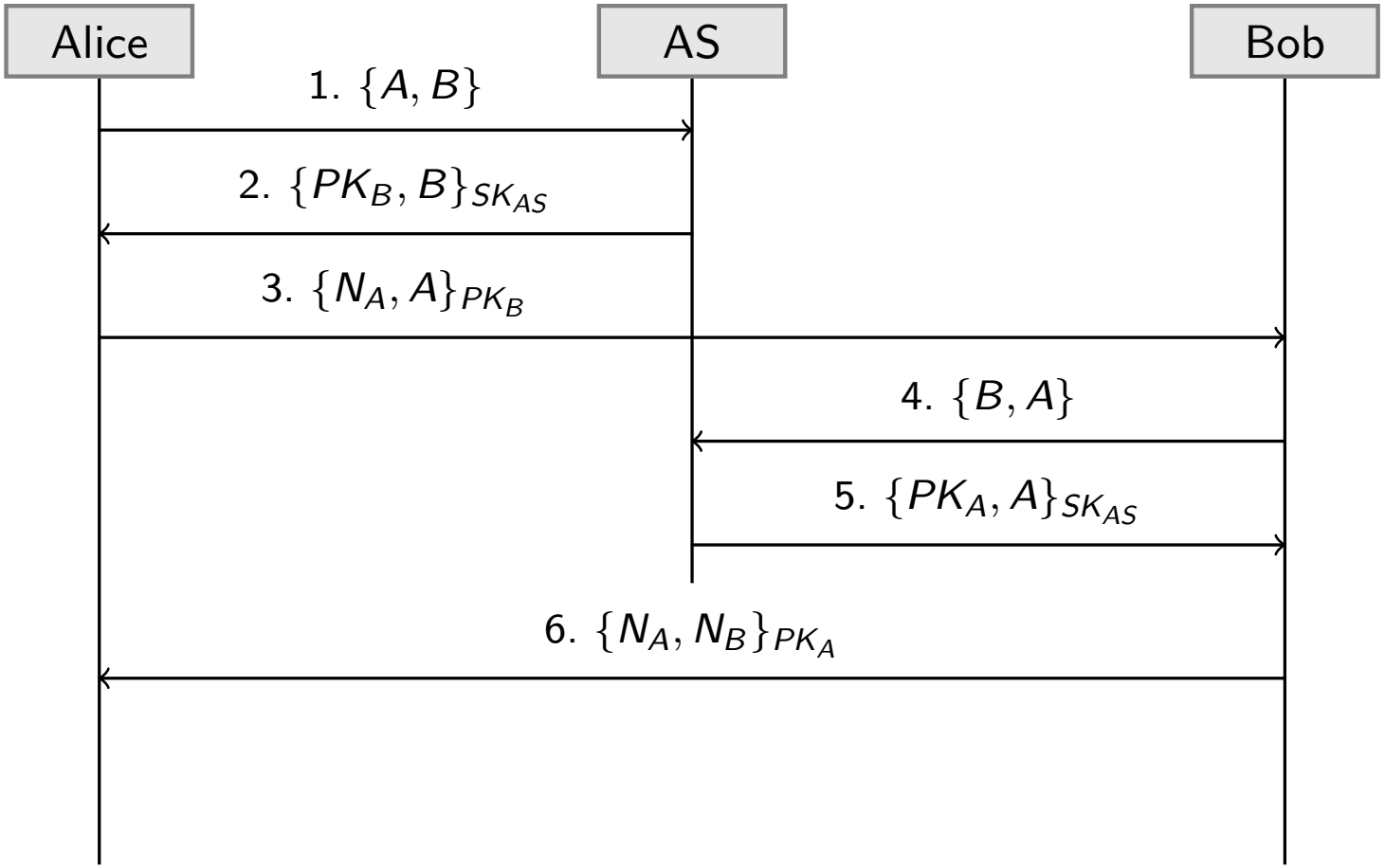
Asymmetric Variant of the NSP



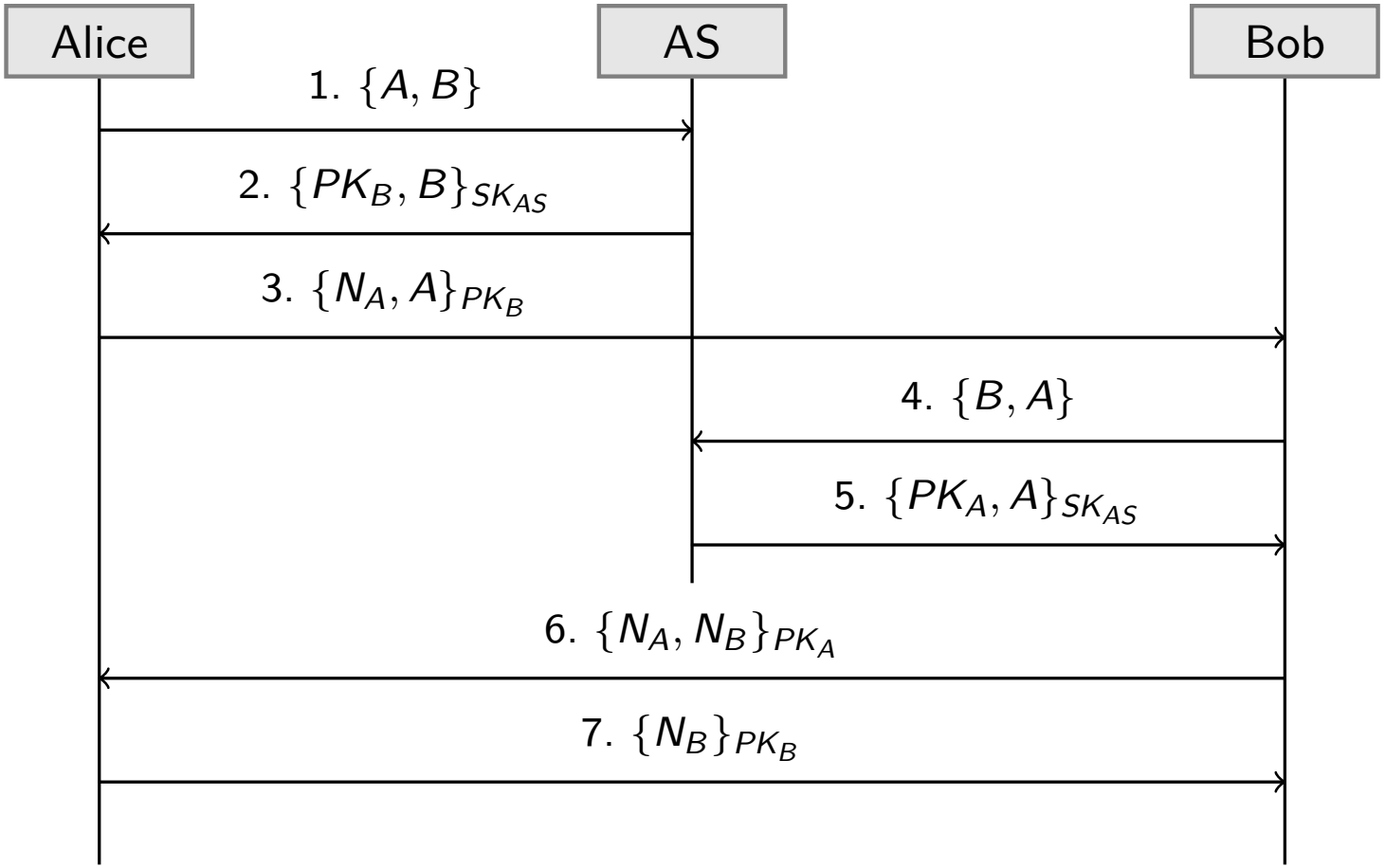
Asymmetric Variant of the NSP



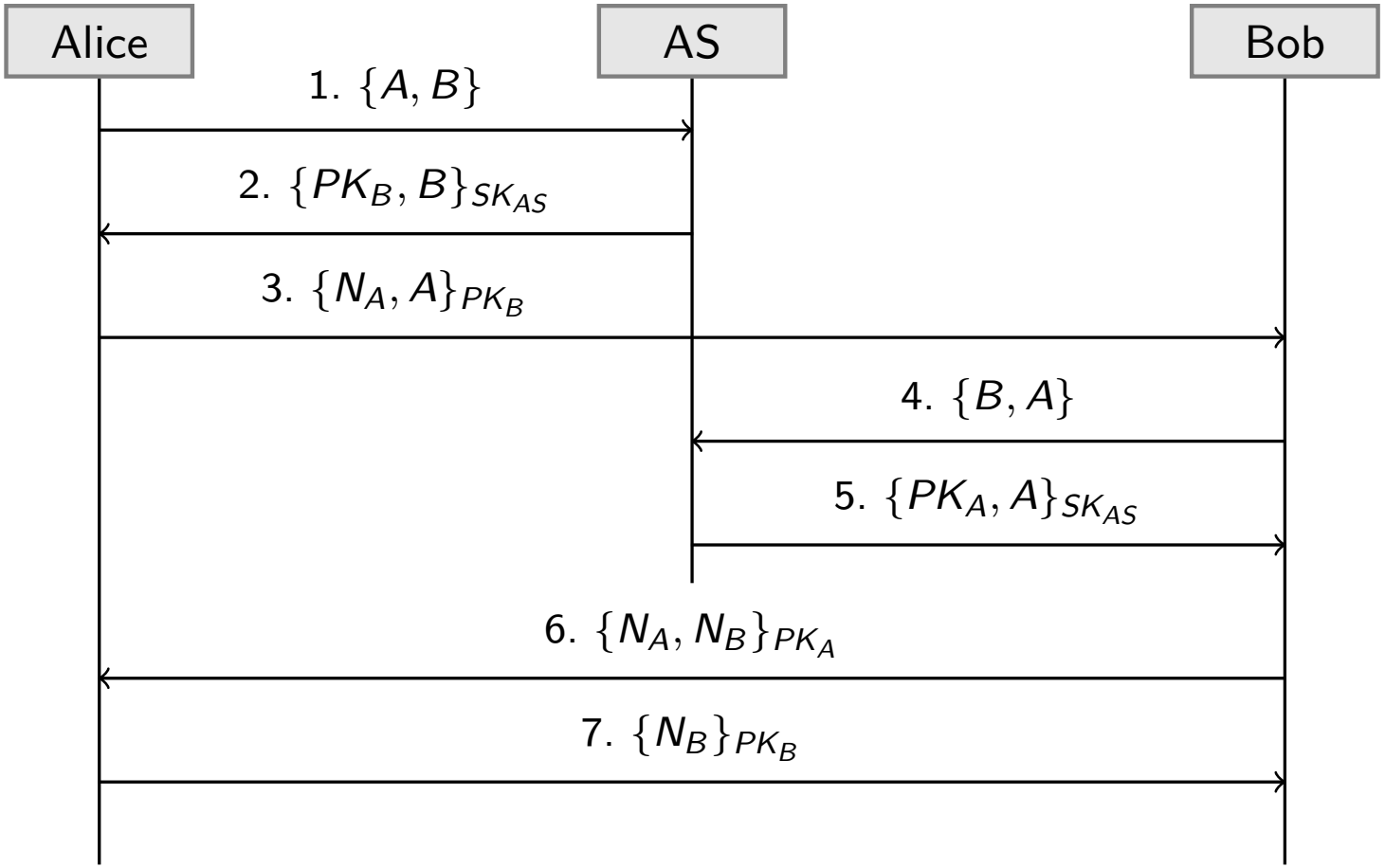
Asymmetric Variant of the NSP



Asymmetric Variant of the NSP



Asymmetric Variant of the NSP



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

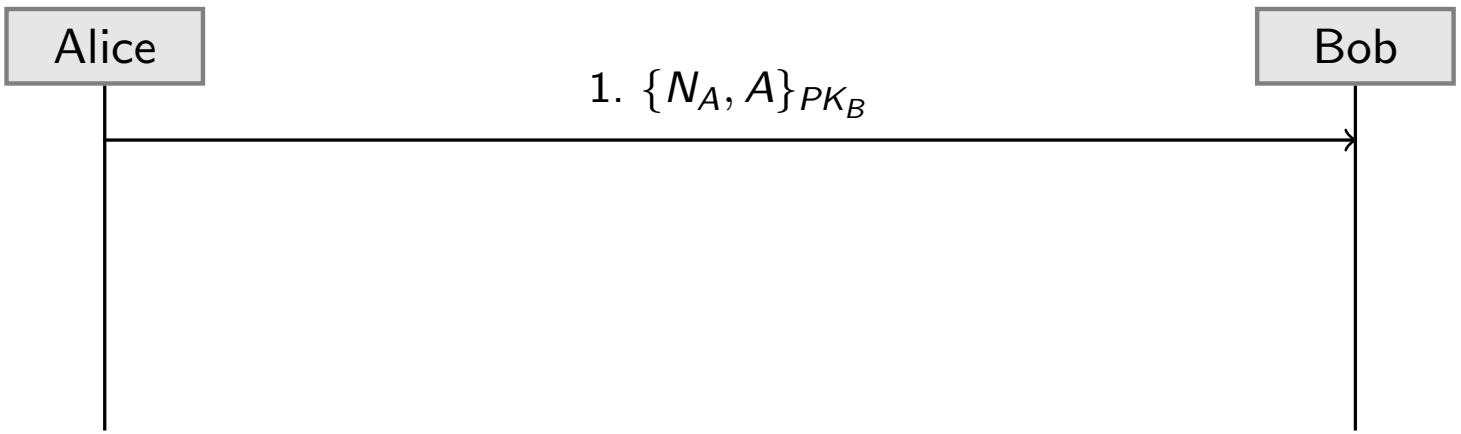
Specify the asymmetric variant of the Needham-Schroeder protocol. What attack for NSP has not been detected for many years?

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

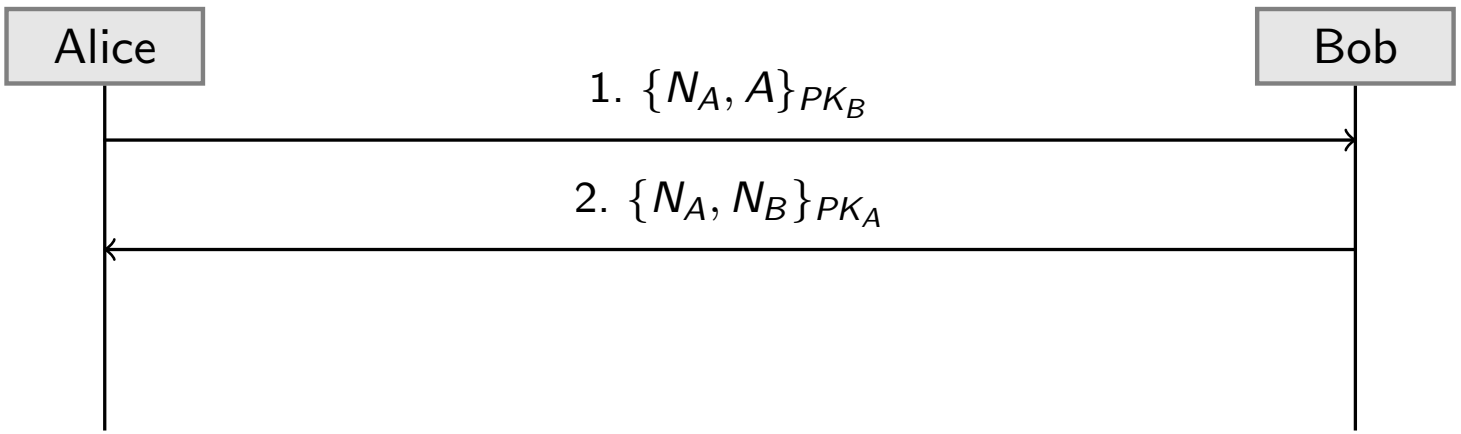
Simplified Version of the Asymmetric NSP

- ➜ Assumption: Participants have already received all required public keys from the AS
- ➜ Therefore, Steps 1,2,4 & 5 can be omitted



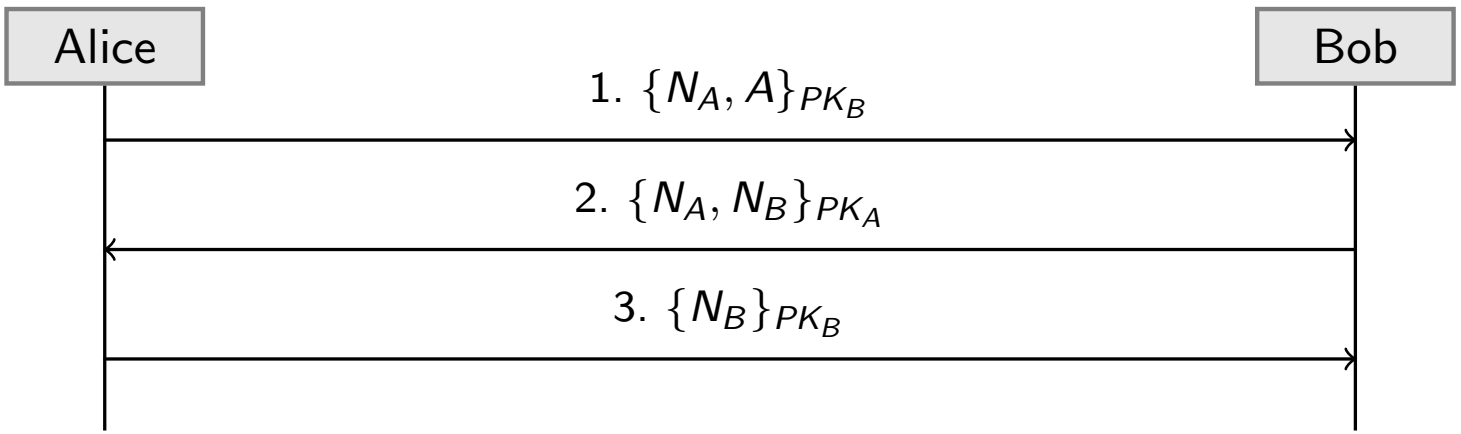
Simplified Version of the Asymmetric NSP

- ➜ Assumption: Participants have already received all required public keys from the AS
- ➜ Therefore, Steps 1,2,4 & 5 can be omitted



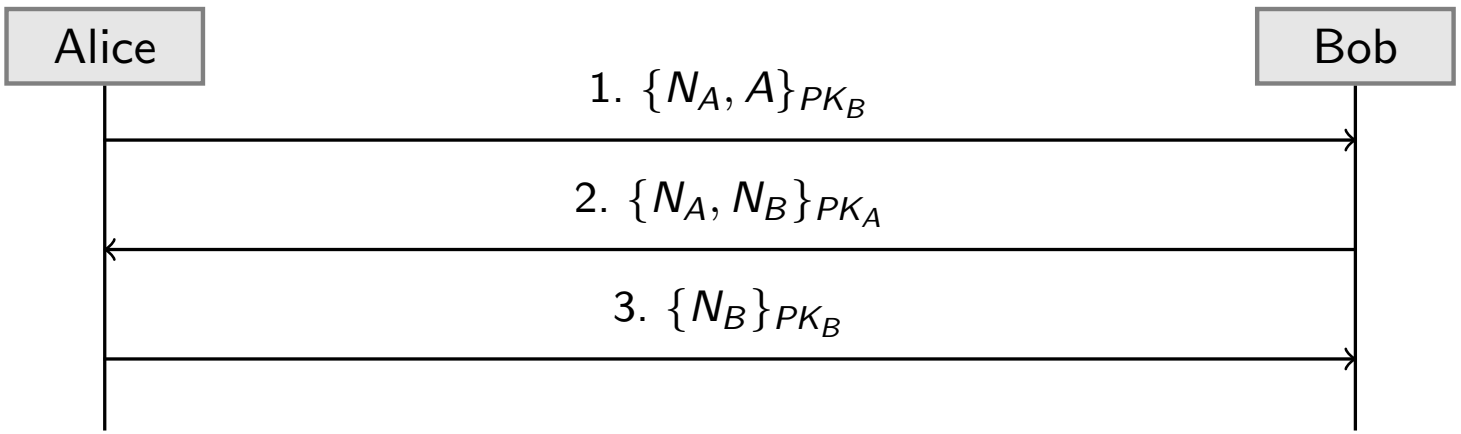
Simplified Version of the Asymmetric NSP

- ➜ Assumption: Participants have already received all required public keys from the AS
- ➜ Therefore, Steps 1,2,4 & 5 can be omitted



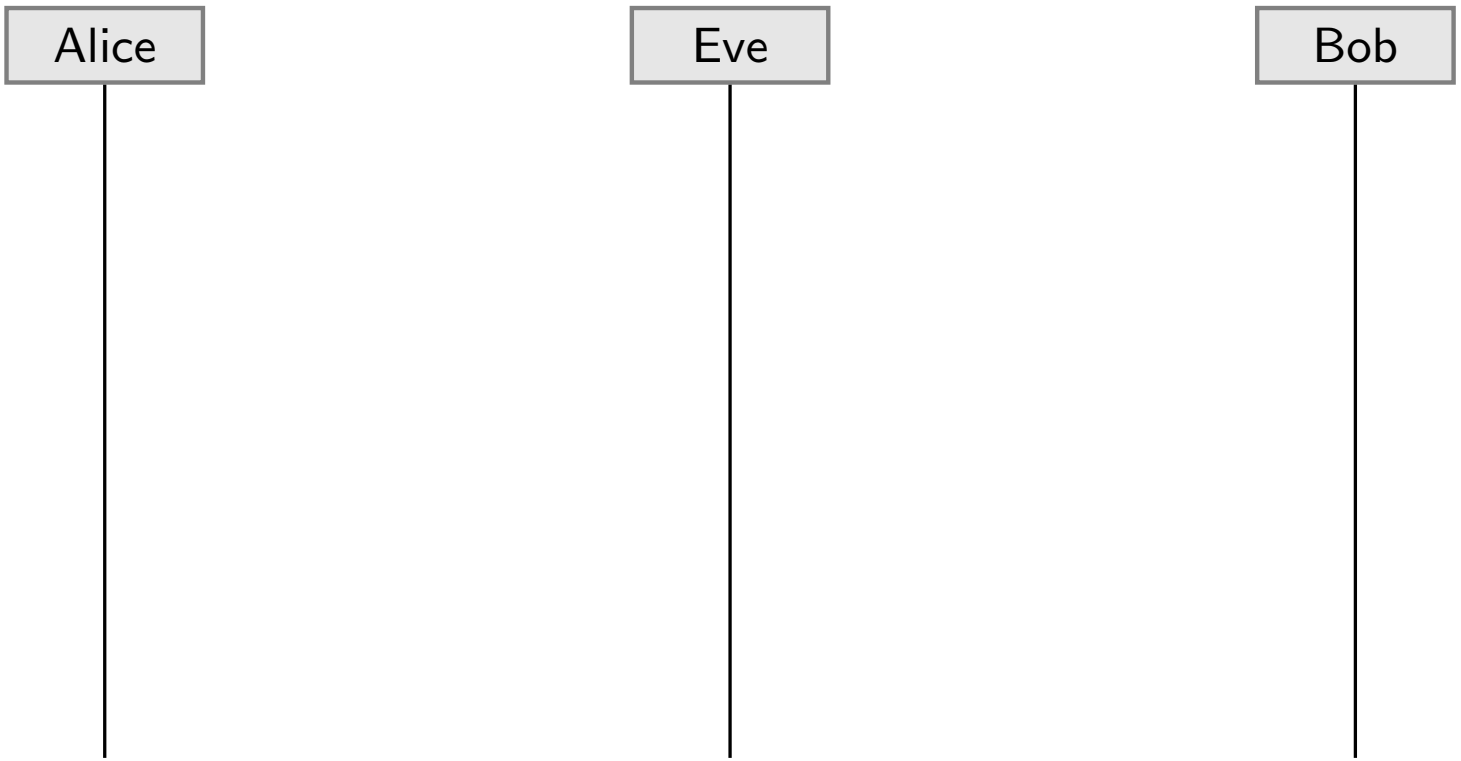
Simplified Version of the Asymmetric NSP

- ➜ Assumption: Participants have already received all required public keys from the AS
- ➜ Therefore, Steps 1,2,4 & 5 can be omitted

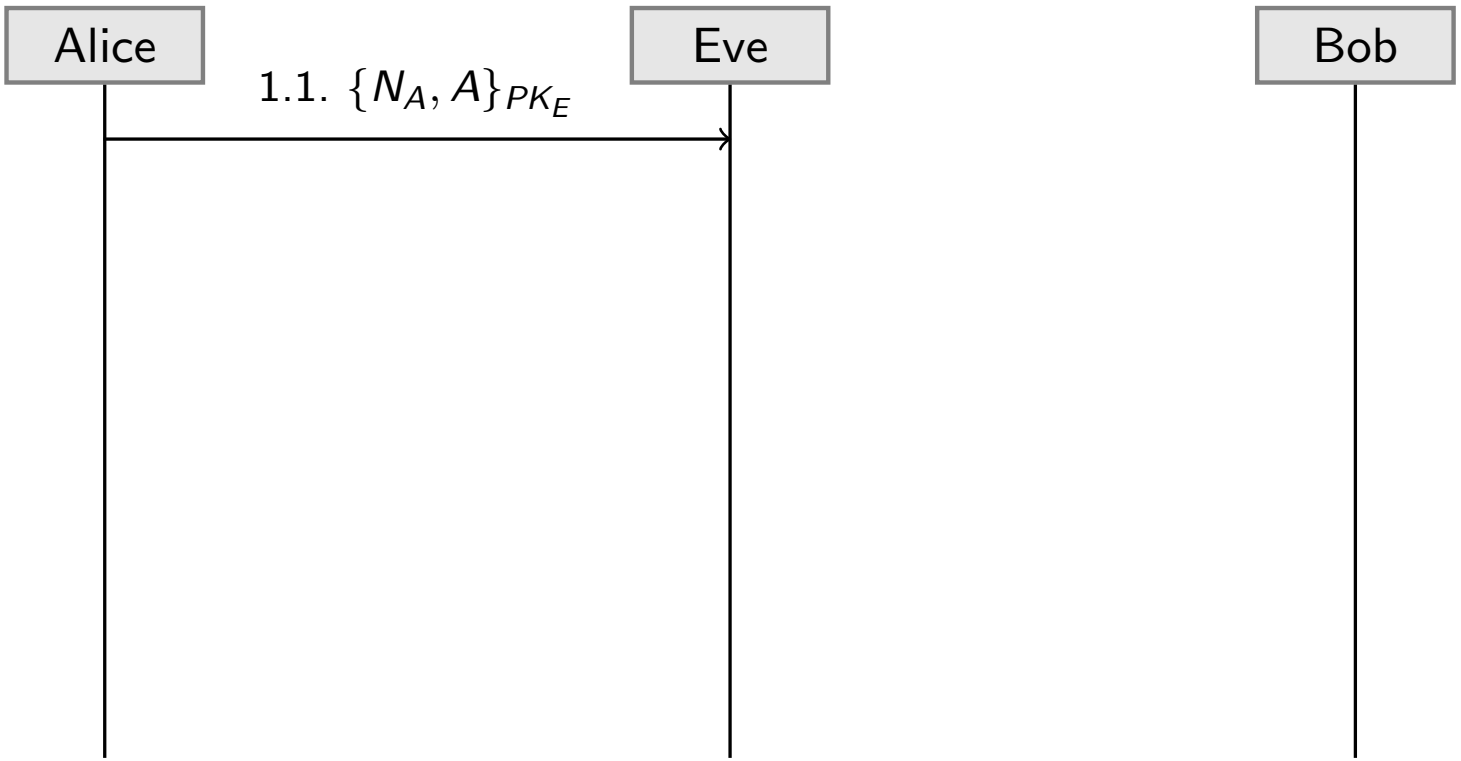


Note: The attacker Eve executes two of these protocol sessions in parallel to perform the attack!

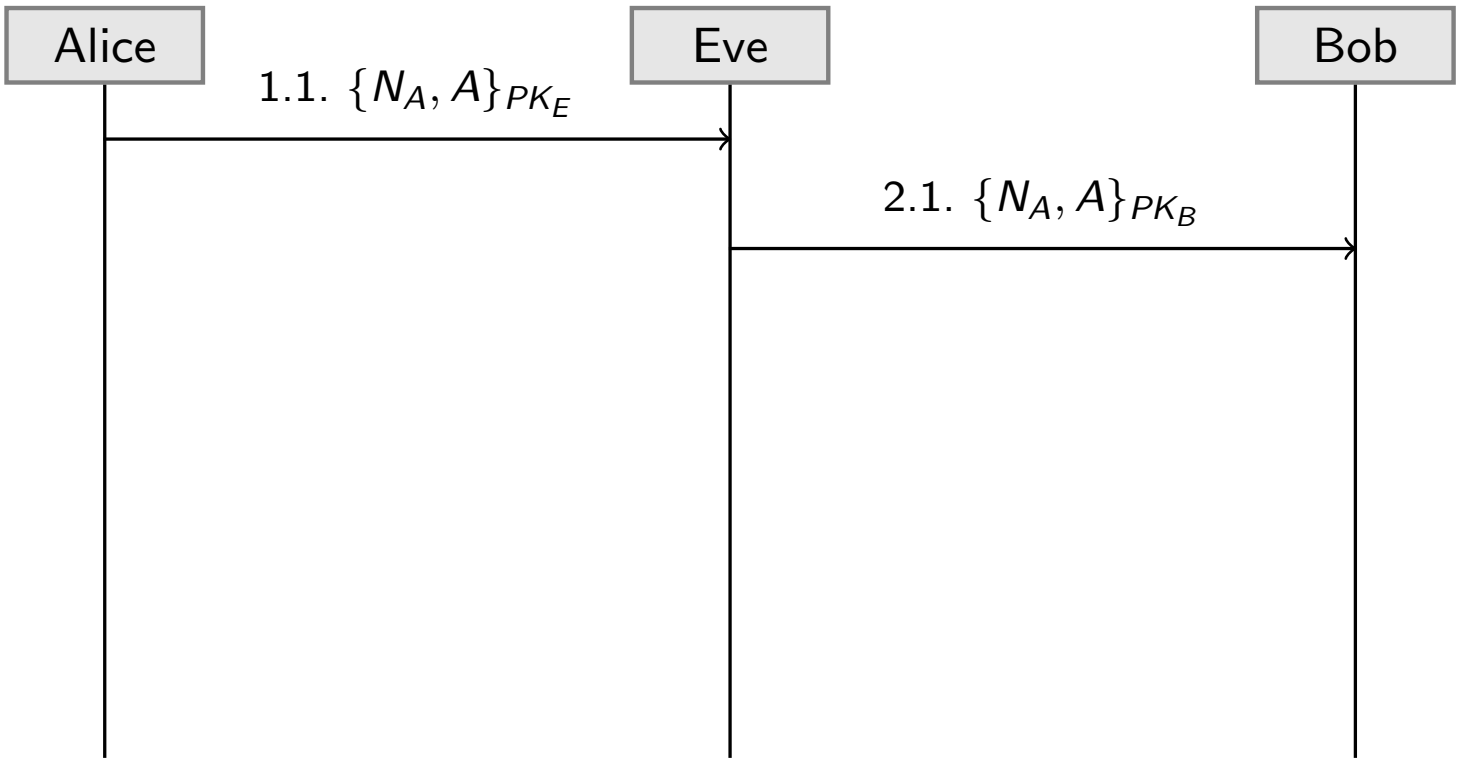
Attack for the Asymmetric Variant of the NSP



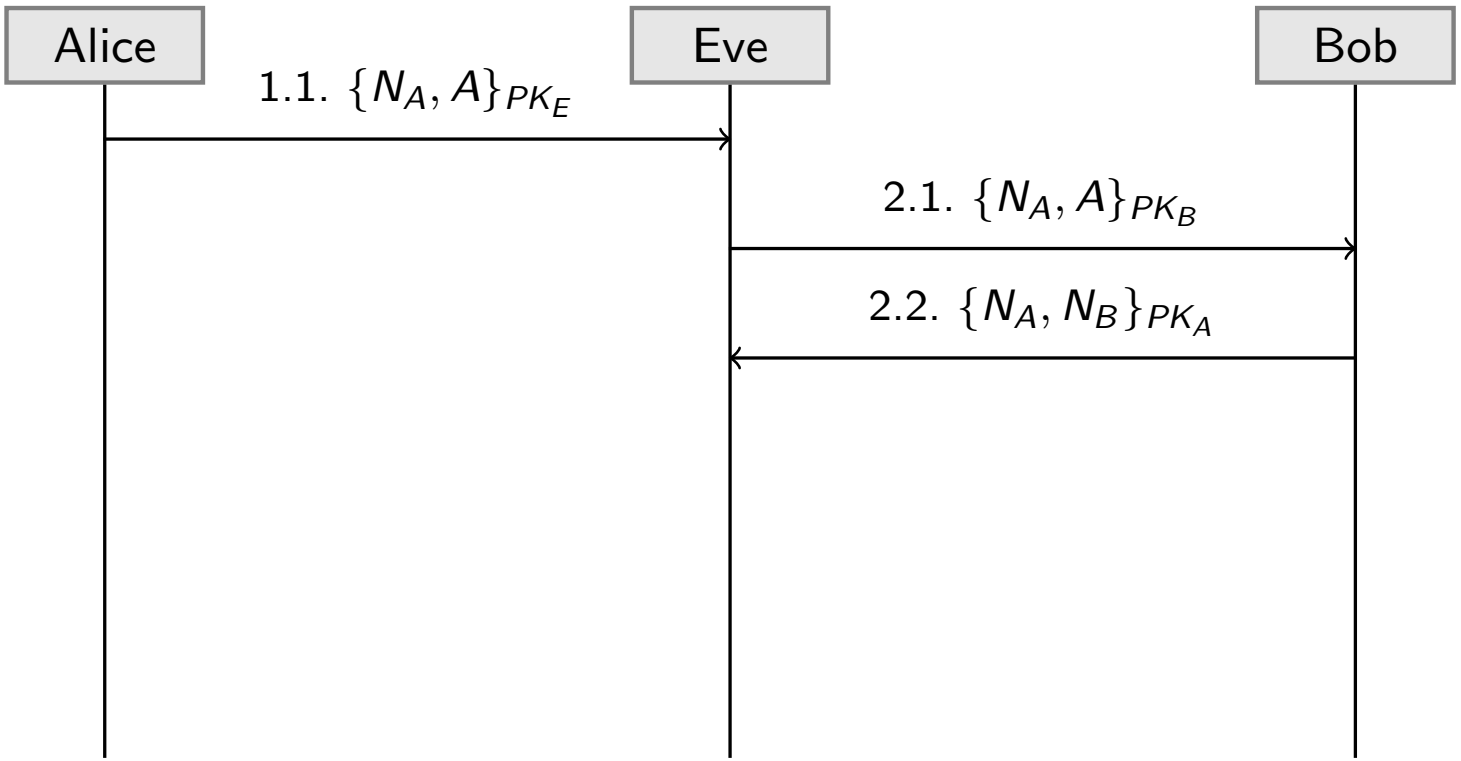
Attack for the Asymmetric Variant of the NSP



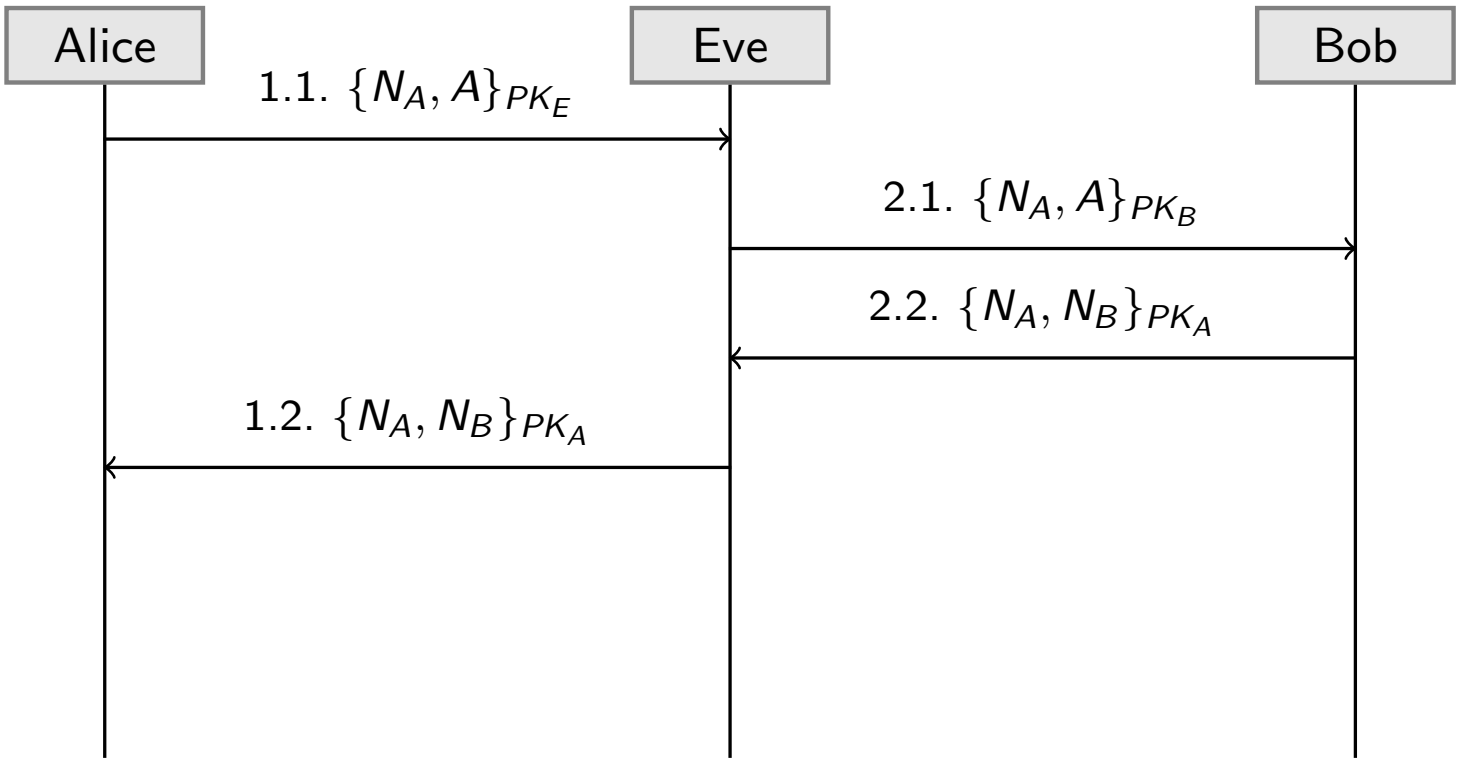
Attack for the Asymmetric Variant of the NSP



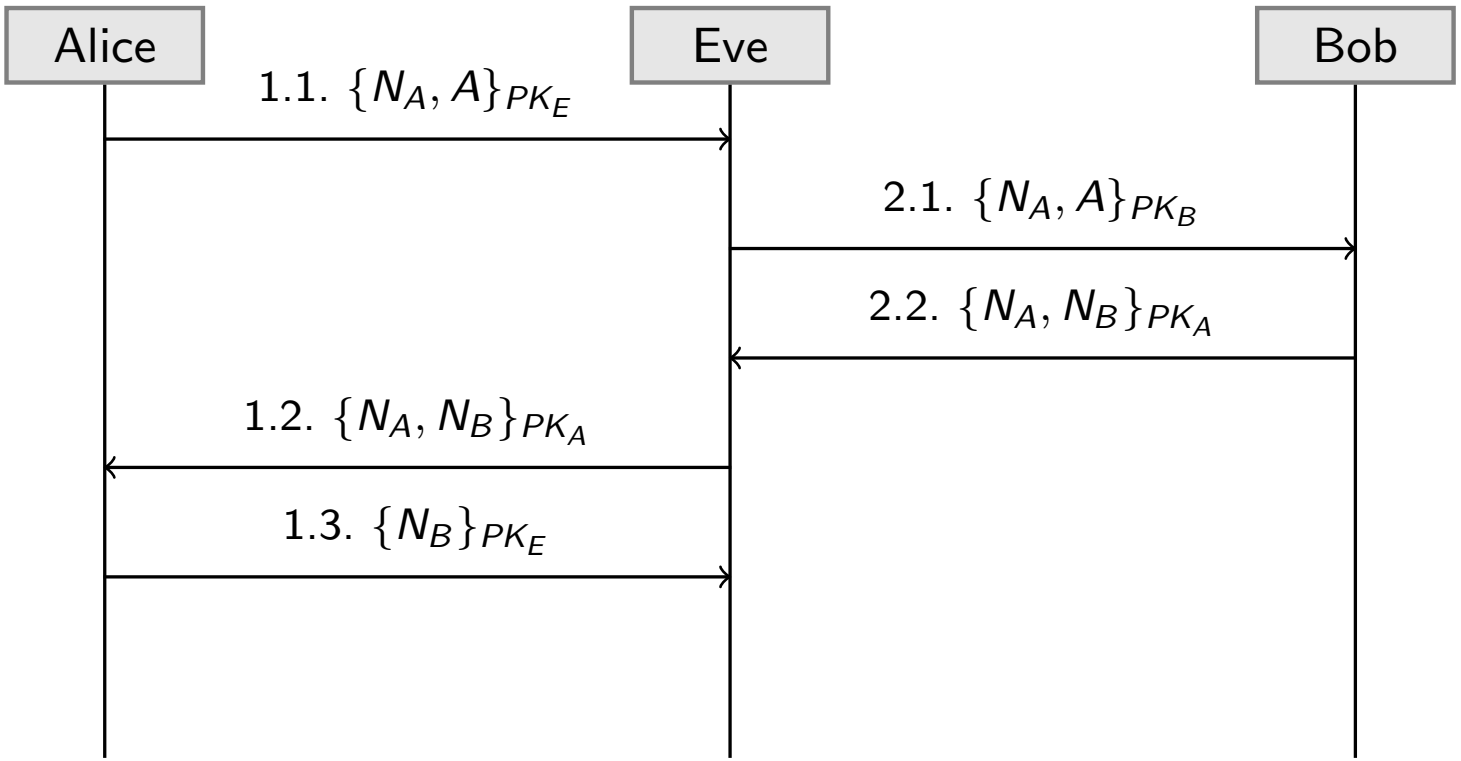
Attack for the Asymmetric Variant of the NSP



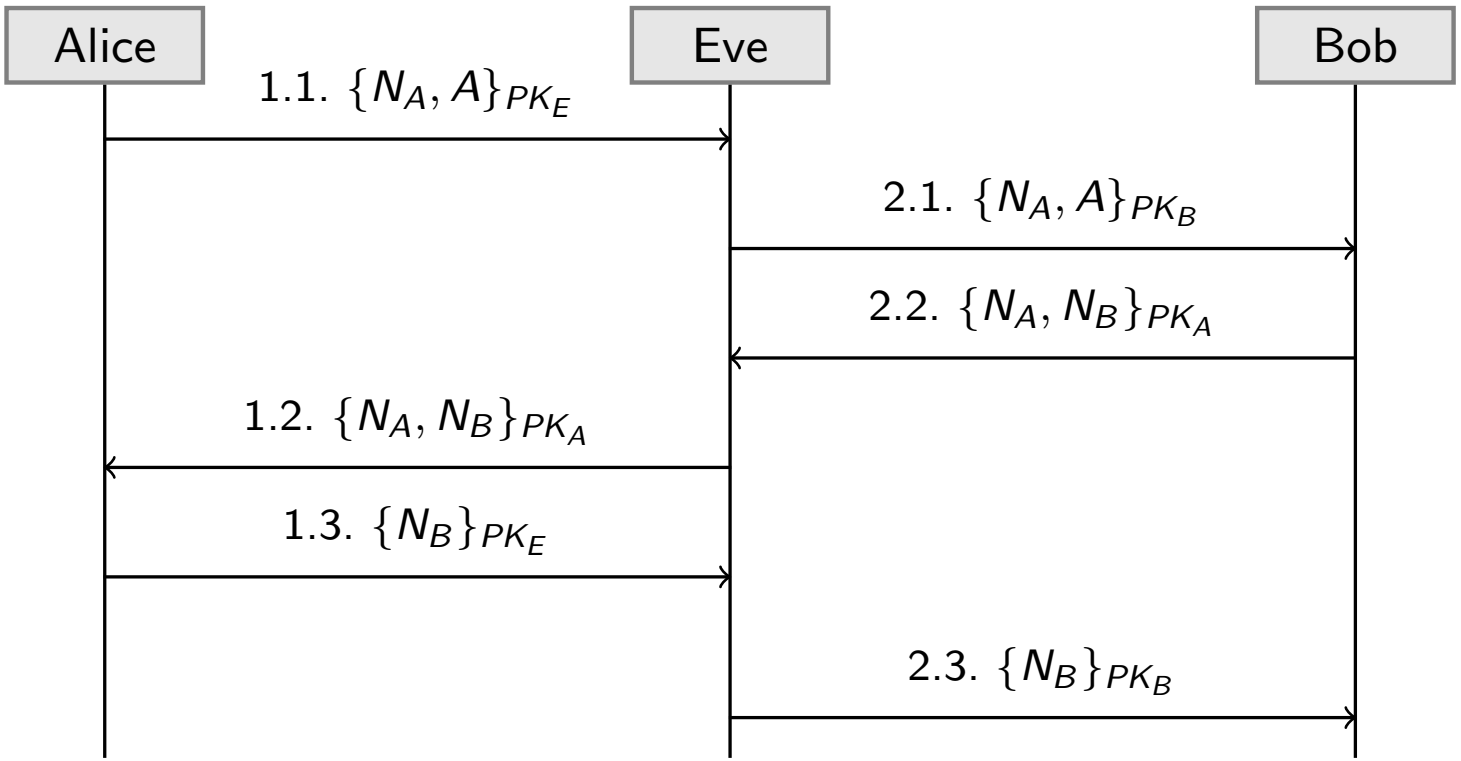
Attack for the Asymmetric Variant of the NSP



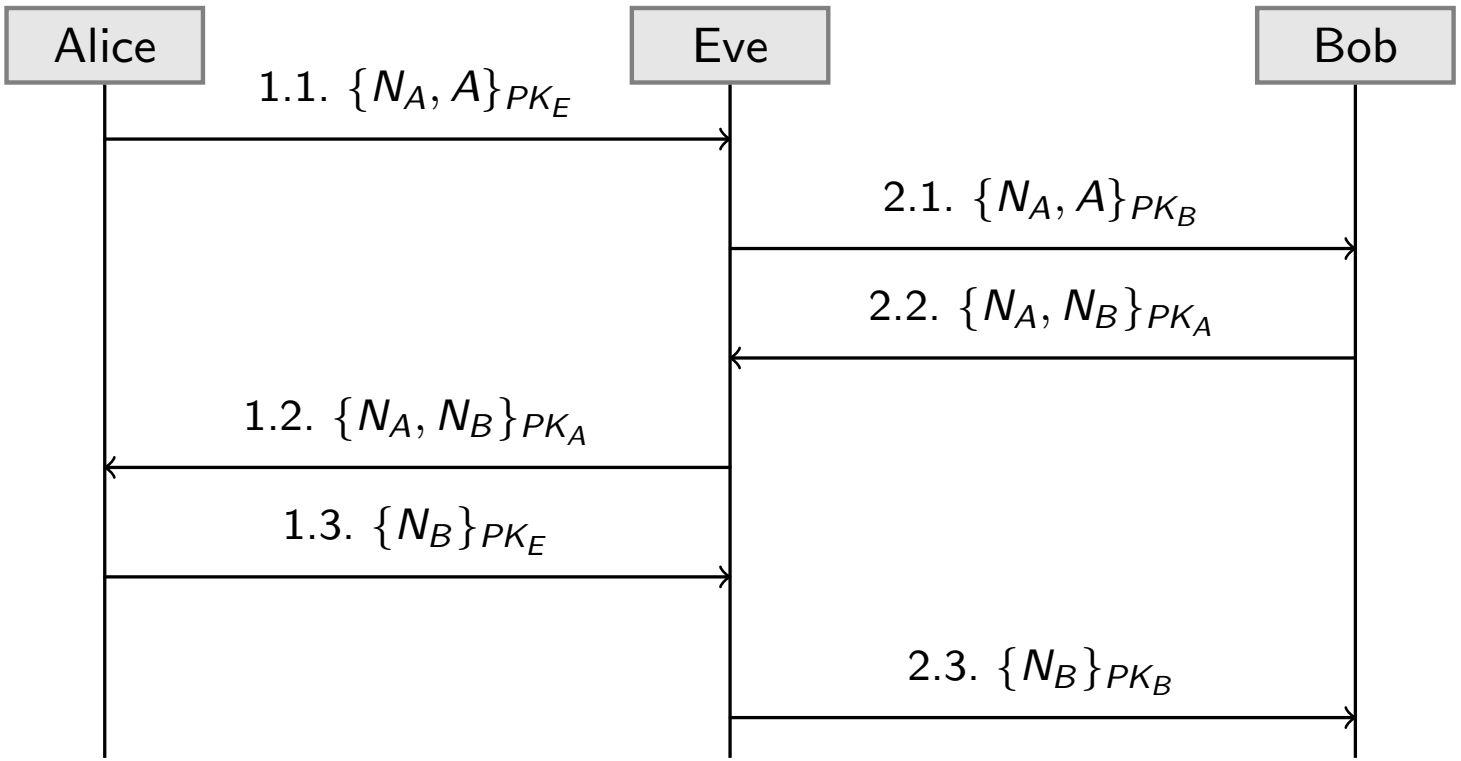
Attack for the Asymmetric Variant of the NSP



Attack for the Asymmetric Variant of the NSP

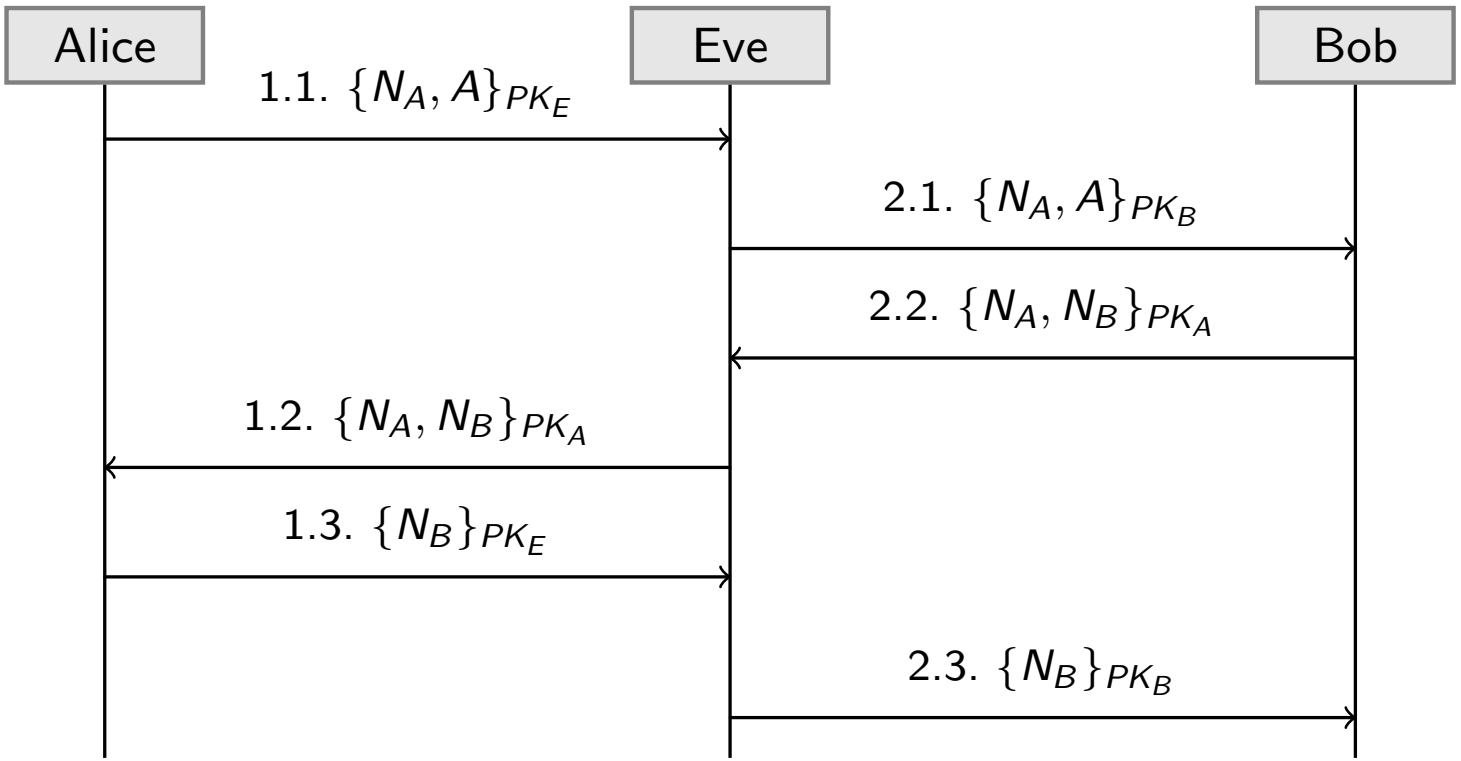


Attack for the Asymmetric Variant of the NSP



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

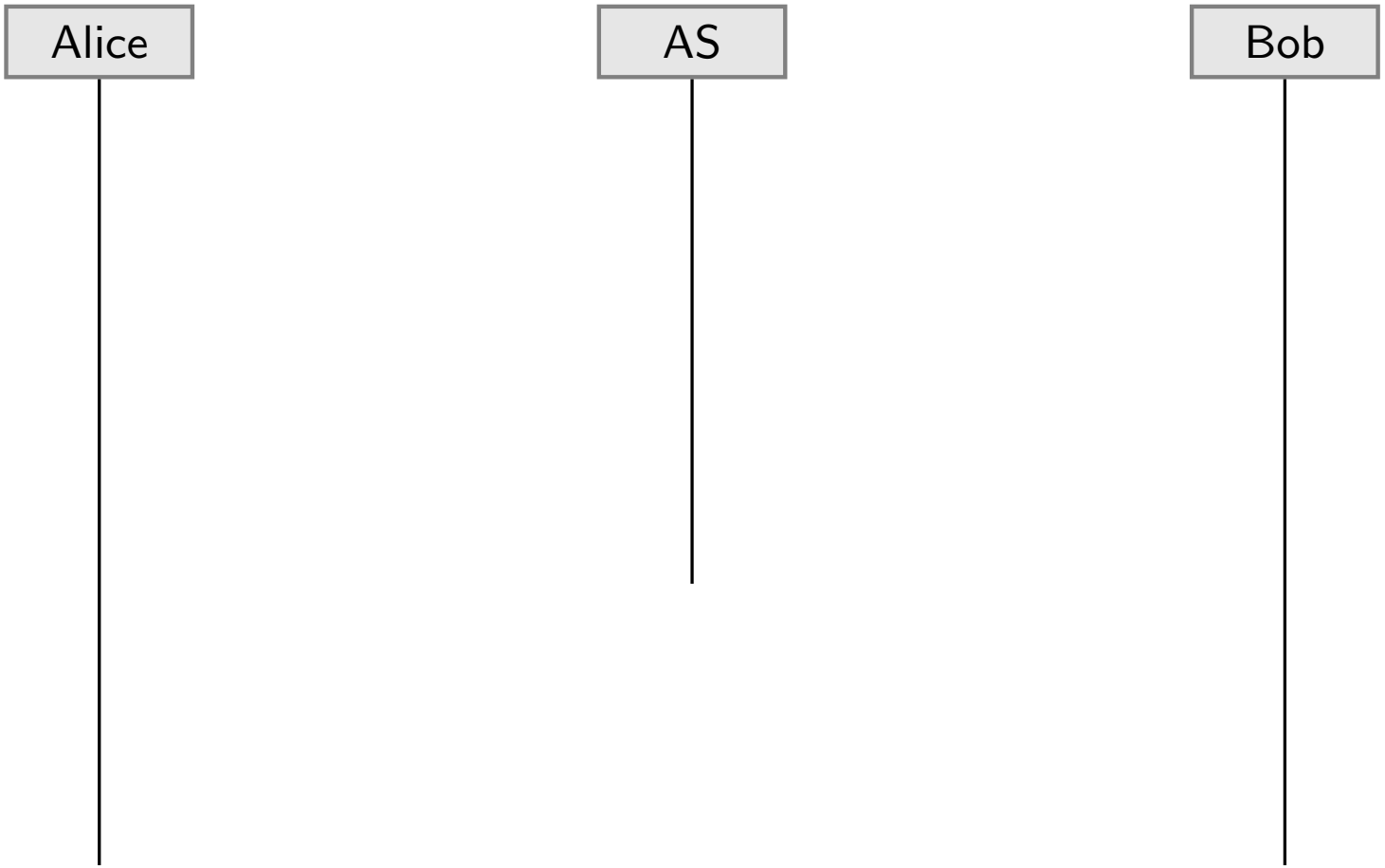
Attack for the Asymmetric Variant of the NSP



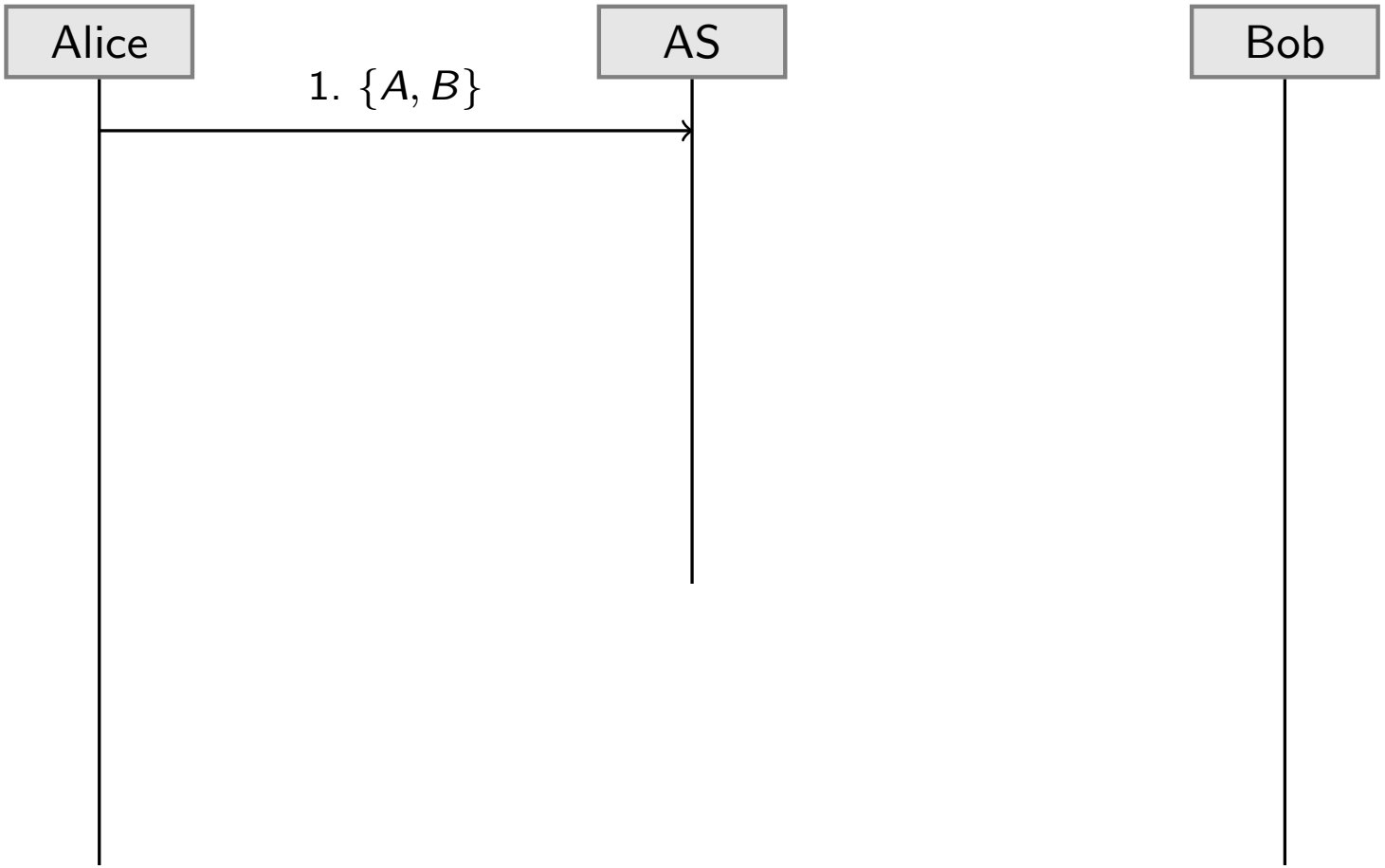
Eve cheats on Bob. She pretends to be Alice in reality.

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

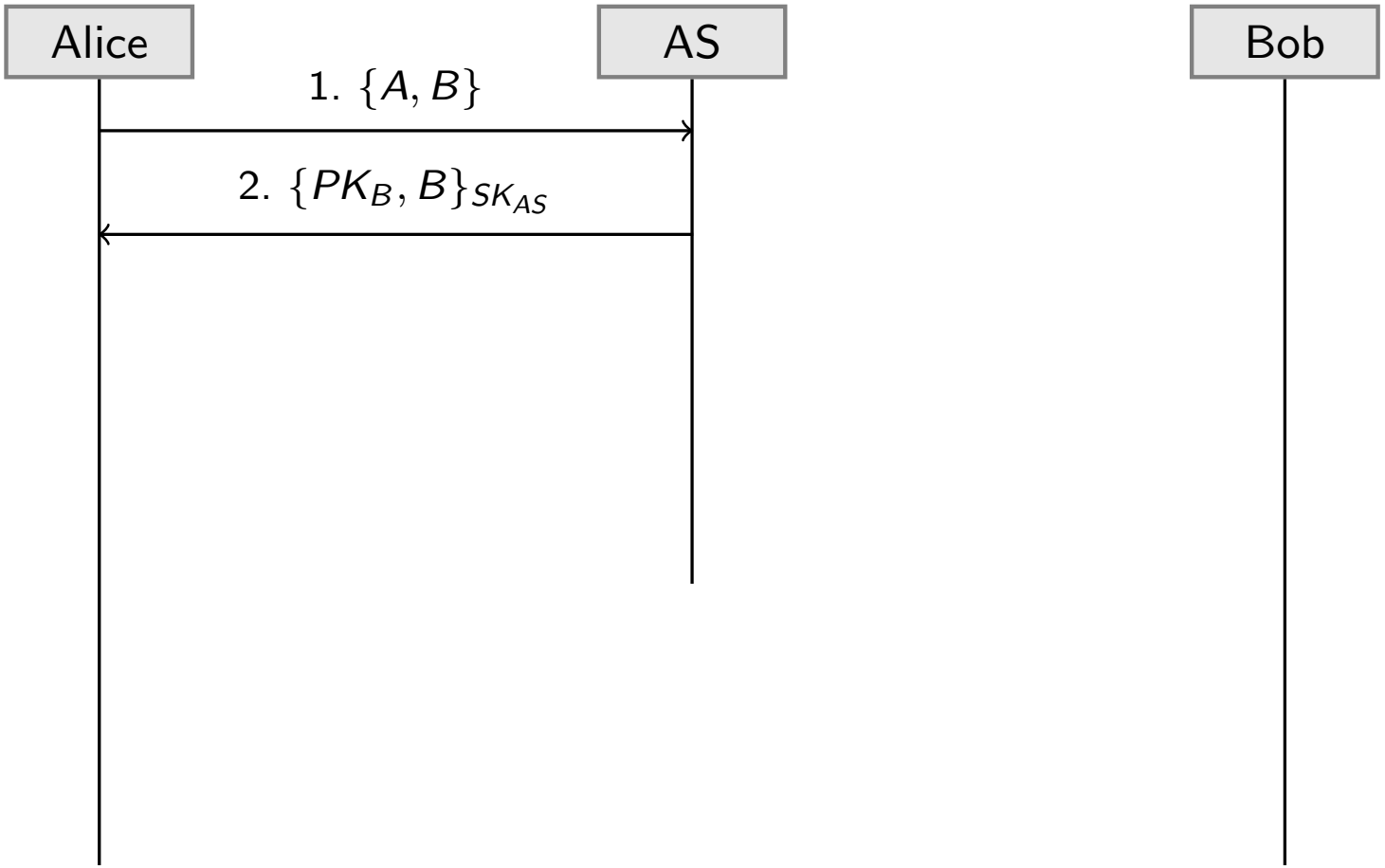
Corrected Variant of the Asymmetric NSP



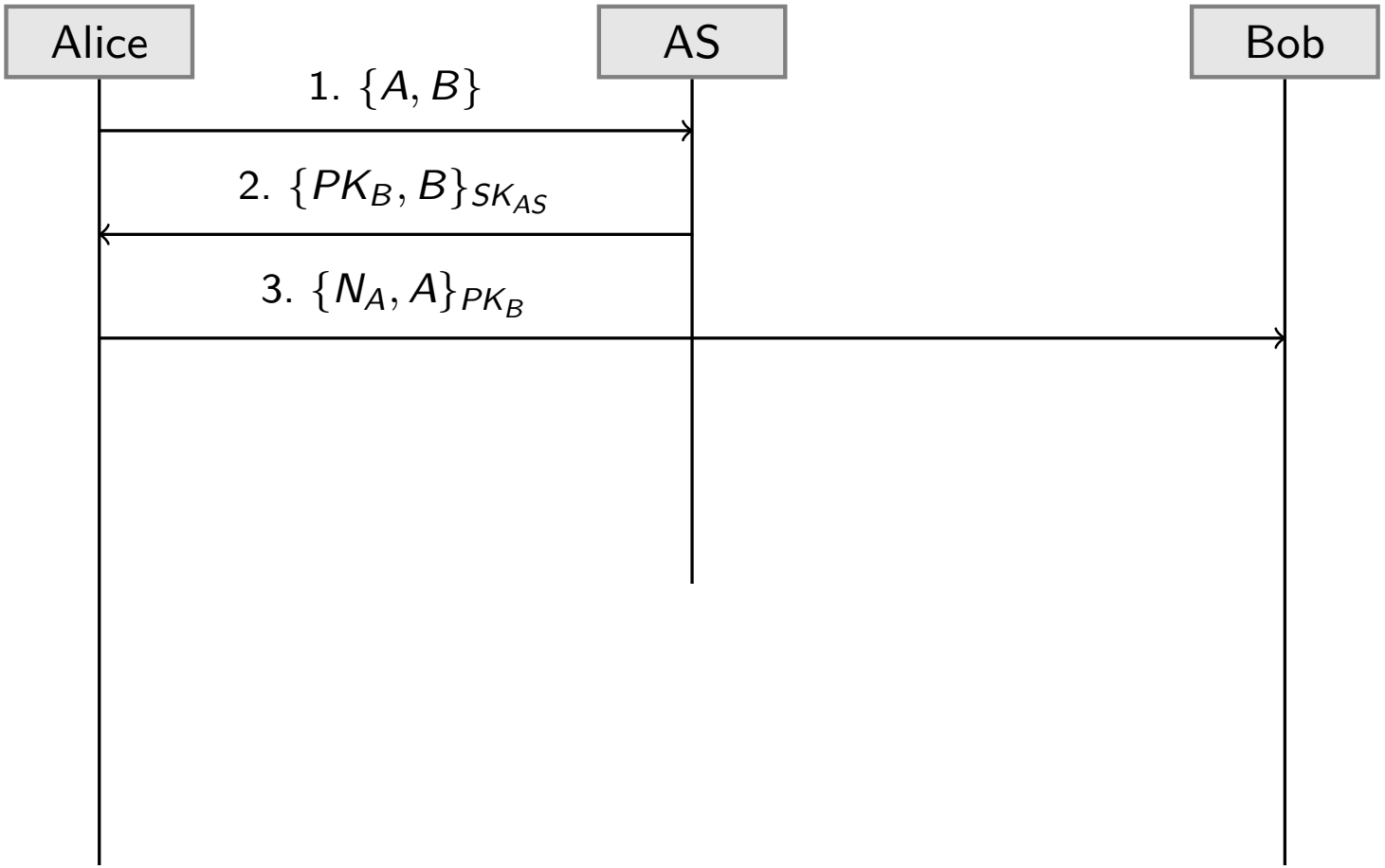
Corrected Variant of the Asymmetric NSP



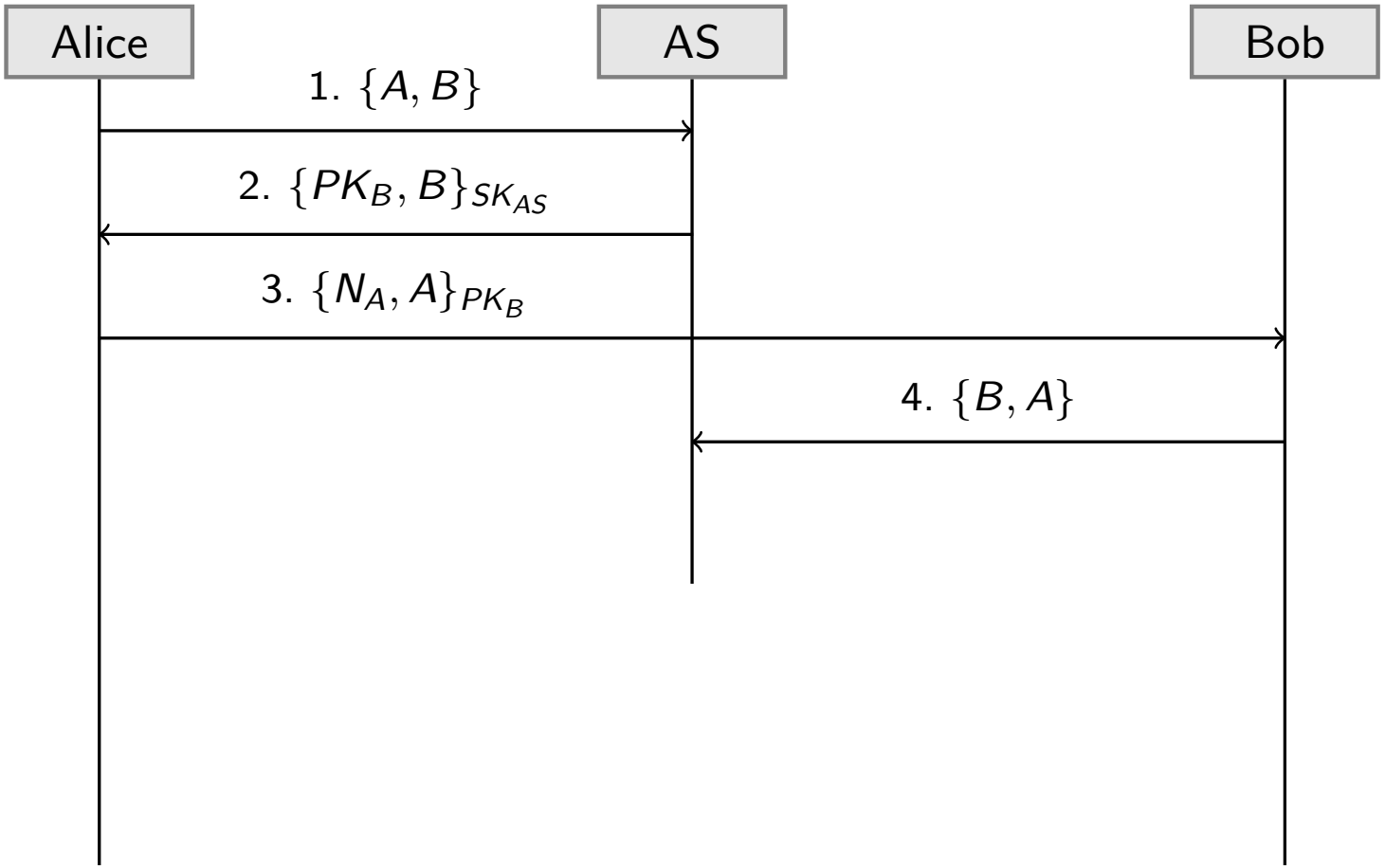
Corrected Variant of the Asymmetric NSP



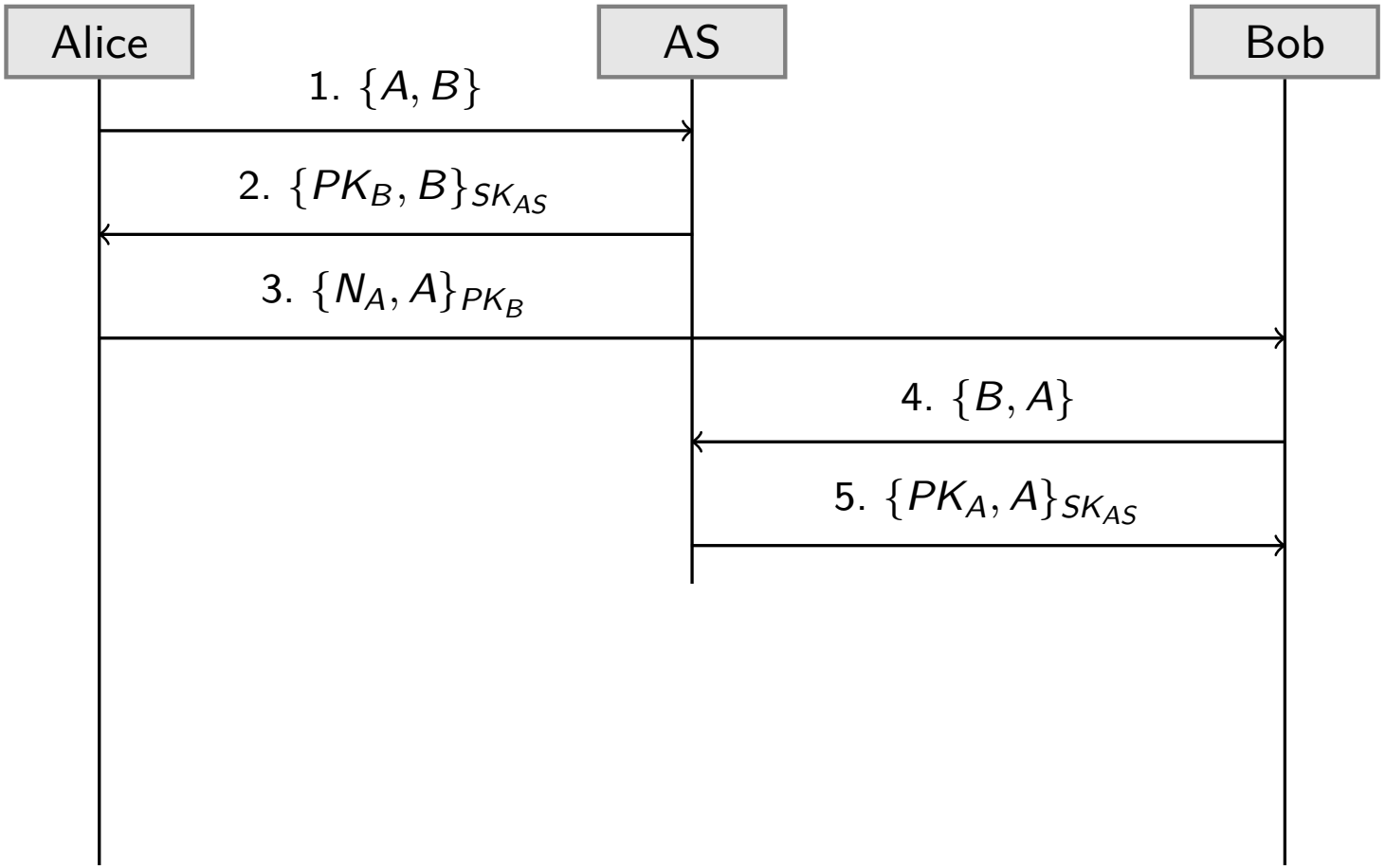
Corrected Variant of the Asymmetric NSP



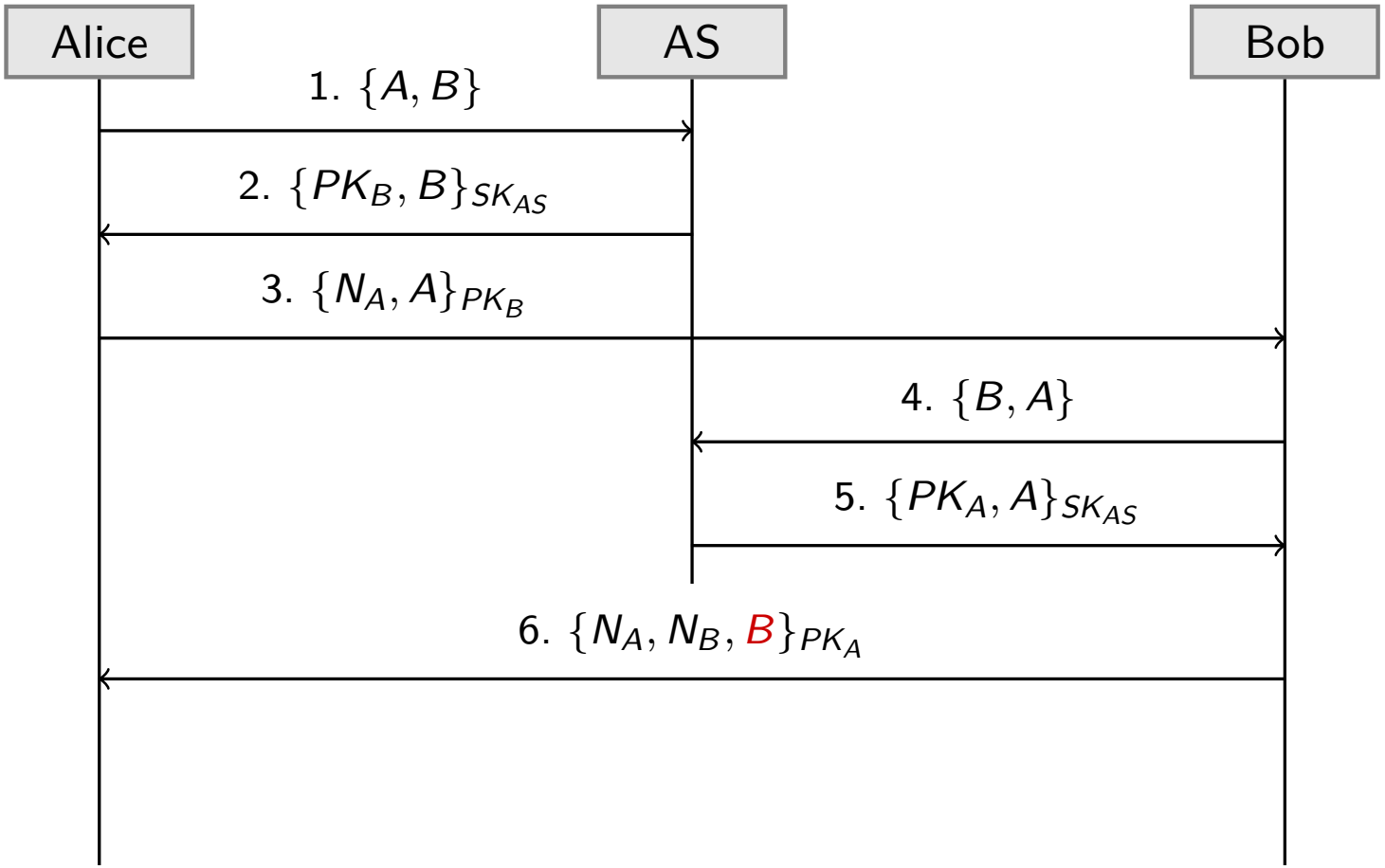
Corrected Variant of the Asymmetric NSP



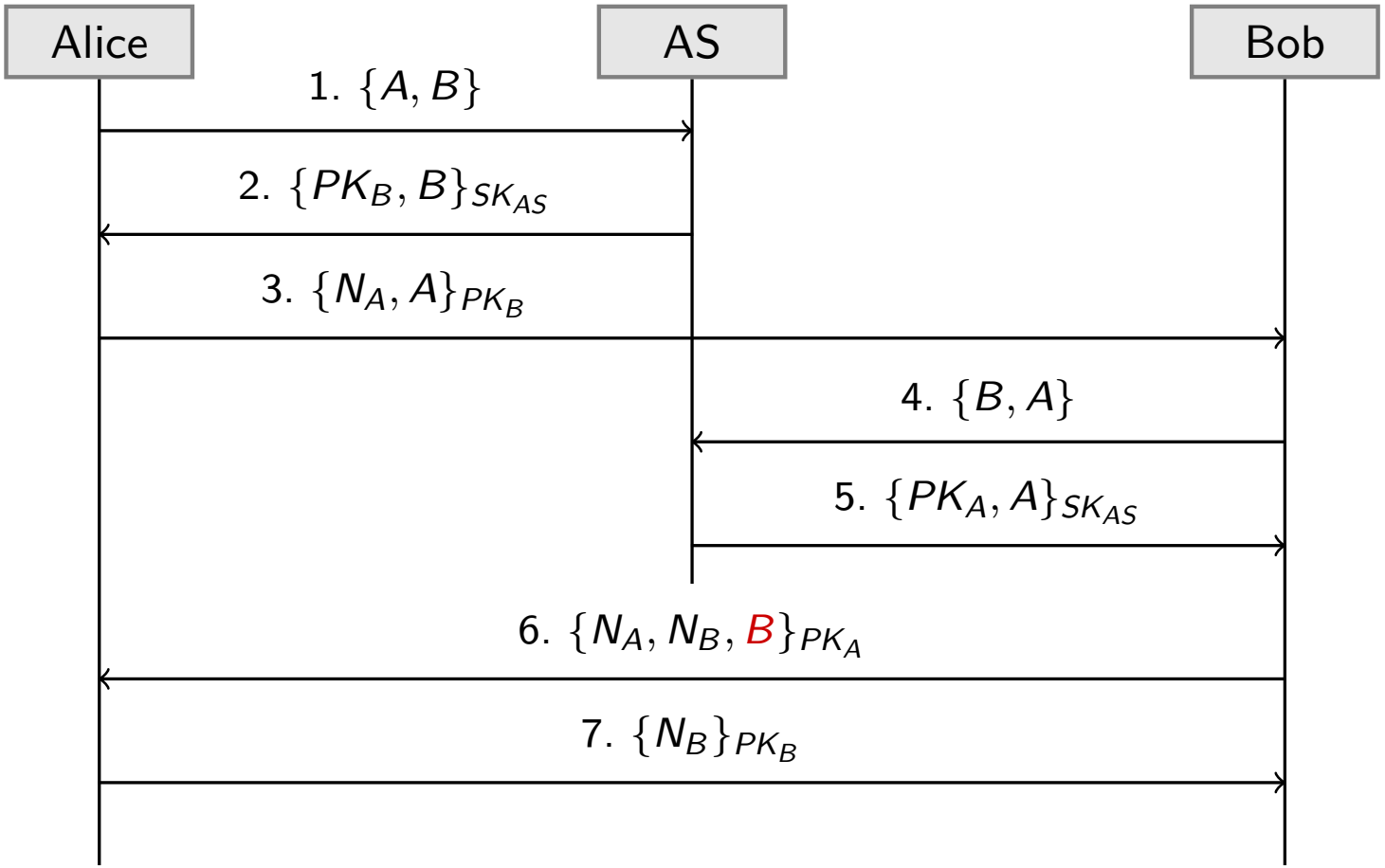
Corrected Variant of the Asymmetric NSP



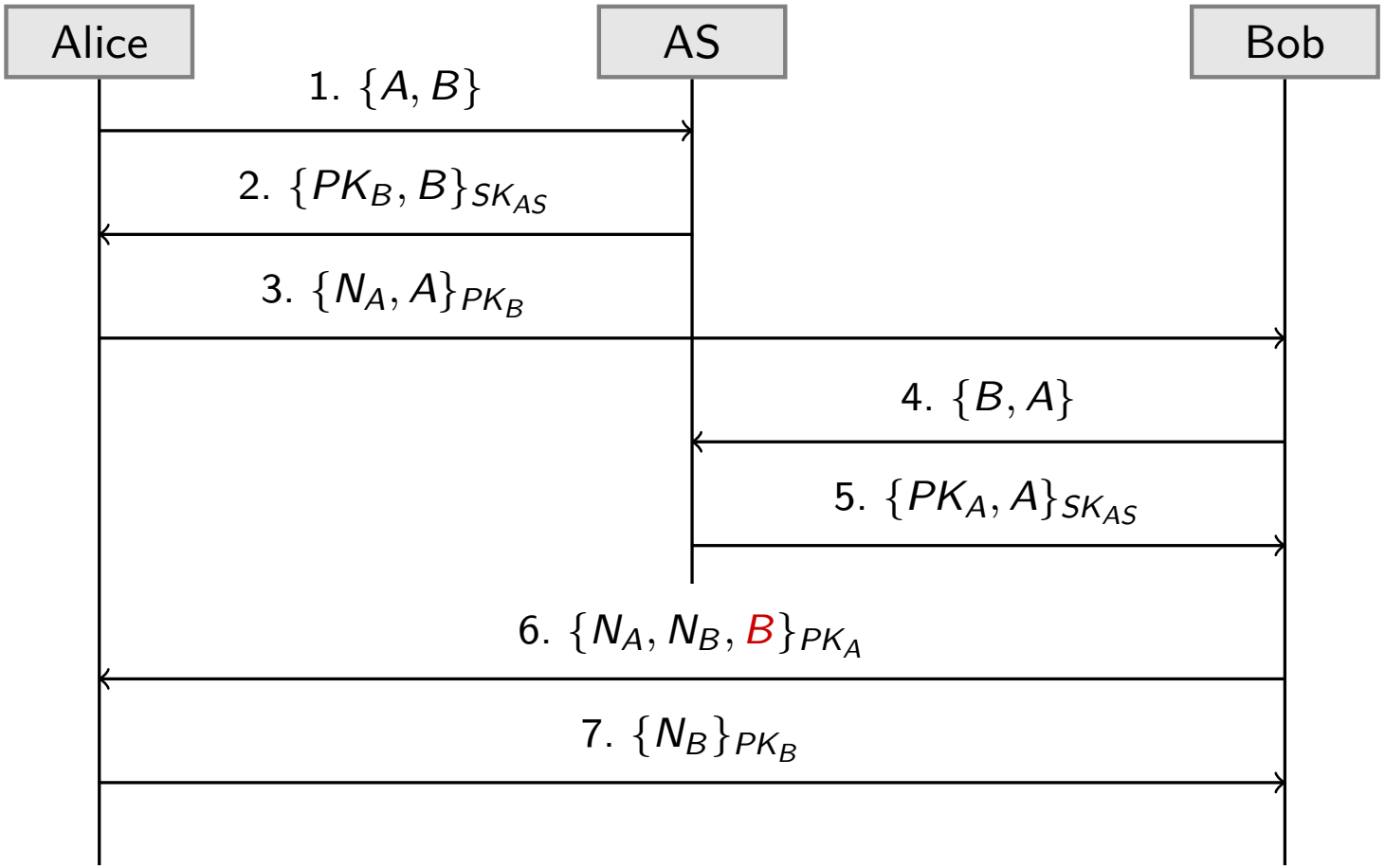
Corrected Variant of the Asymmetric NSP



Corrected Variant of the Asymmetric NSP



Corrected Variant of the Asymmetric NSP



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