Computer and Network Security: Mediated Authentication

Kameswari Chebrolu

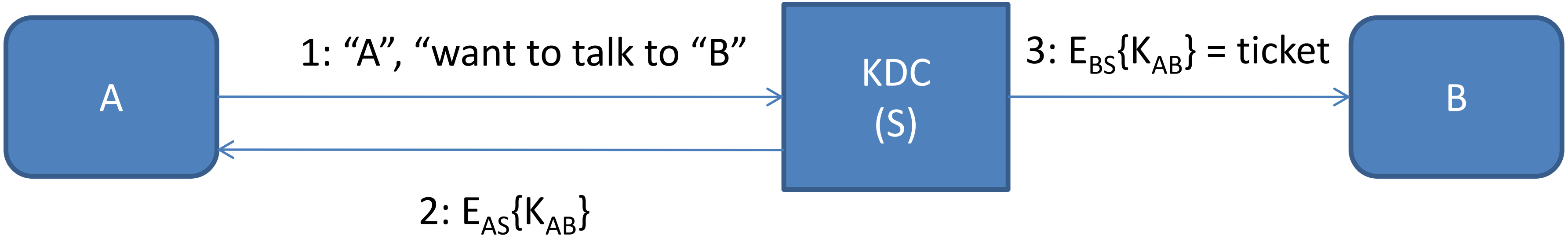
Outline

- Human Authentication
  - Focus: Password based systems
- **Cryptographic Authentication** (Human as well as computer): Prove identity by performing a cryptographic operation (hash, encryption etc)
  - ~~One way authentication (shared and public key)~~
  - ~~Mutual authentication (shared and public key)~~
  - **Mediated authentication (shared key)**
  - How to incorporate session key exchange?
  - How to follow it up to provide privacy and Integrity?

Mediated Authentication

- Long-term key in place between nodes and KDC
- KDC facilitates communication between nodes
  - Nodes do not share any shared key apriori between them
  - KDC helps nodes with a short term shared session key
- Need to ensure
  - Authentication: Am I talking with the right person
  - Short term session key establishment

Recap: KDC

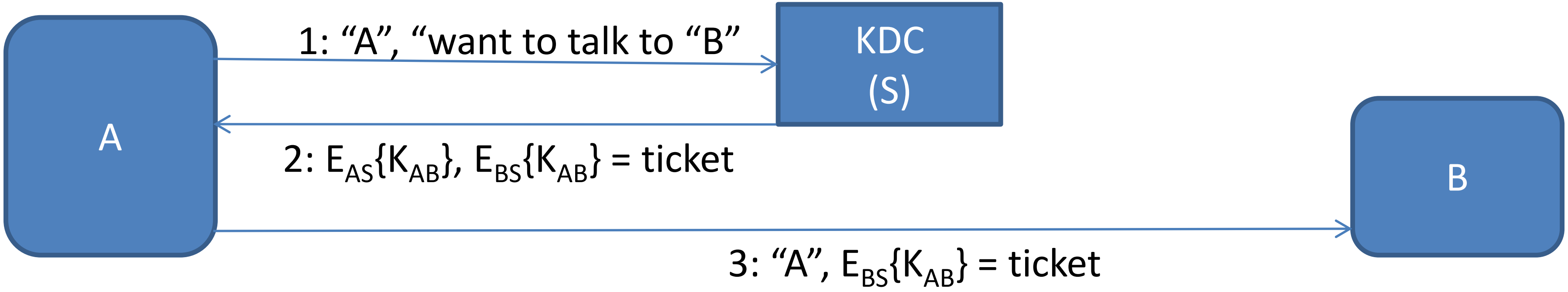


(a)

Ticket allows A to communicate with B

Better Solution

Ticket allows A to communicate with B



A

KDC (S)

B

1: “A”, “want to talk to “B”

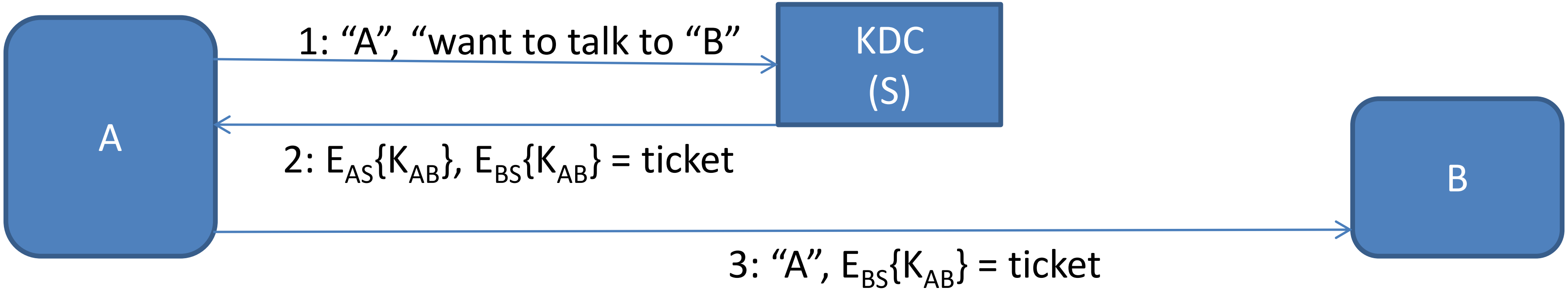
2: $E_{AS}\{K_{AB}\}$, $E_{BS}\{K_{AB}\}$ = ticket

3: “A”, $E_{BS}\{K_{AB}\}$ = ticket

(b)

- Why like this?
  - A’s message may reach B before KDC could share ticket with B
  - A is anyway talking with B; Why let KDC open another?

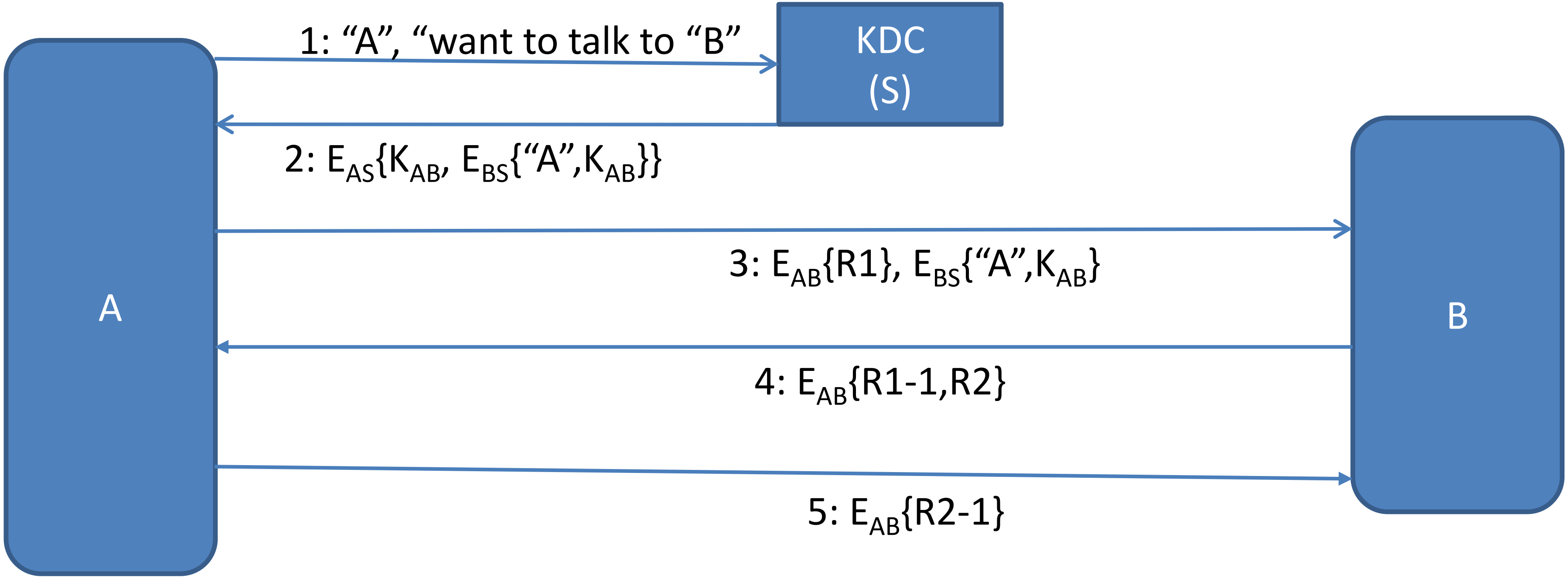
Ticket allows A to communicate with B


A
KDC
(S)
B
1: “A”, “want to talk to “B”
2: $E_{AS}\{K_{AB}\}$, $E_{BS}\{K_{AB}\}$ = ticket
3: “A”, $E_{BS}\{K_{AB}\}$ = ticket

(b)

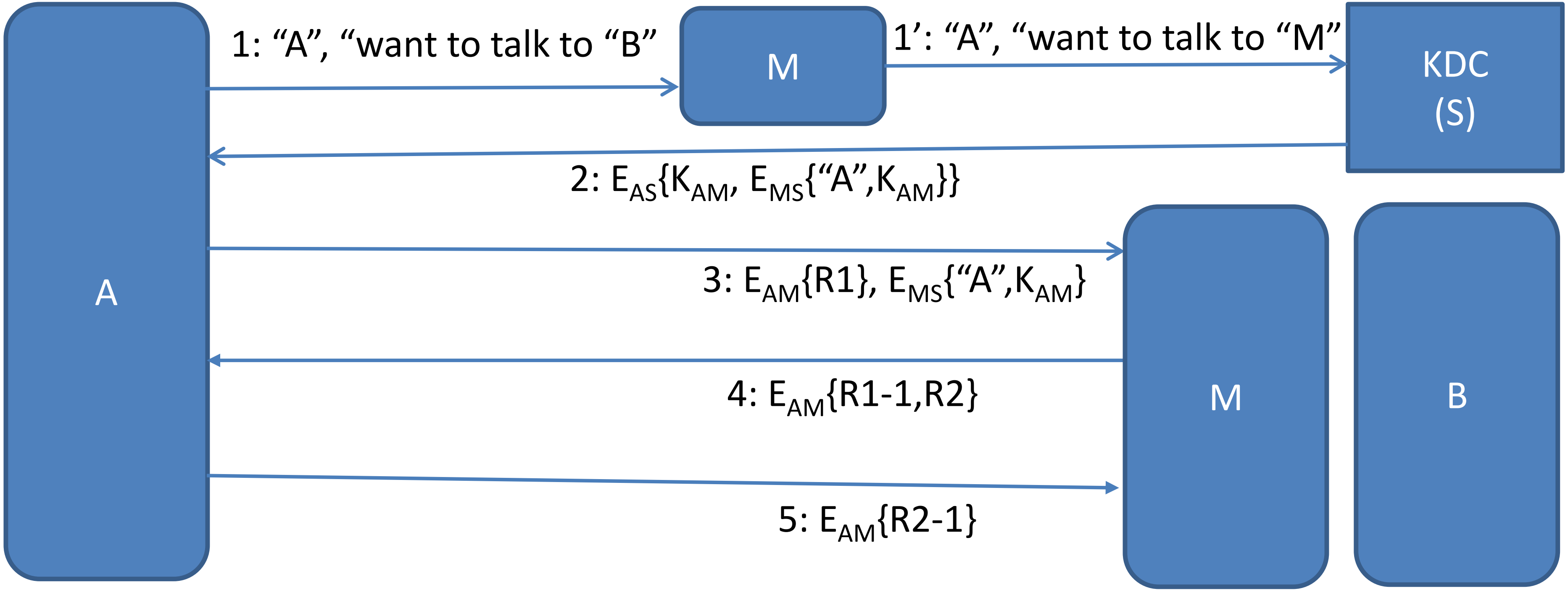
- No authentication or freshness check
- Subject to replay and man-in-the-middle attacks

Version-1


1: “A”, “want to talk to “B”
KDC
(S)
2: $E_{AS}\{K_{AB}, E_{BS}\{“A”,K_{AB}\}\}$
A
3: $E_{AB}\{R1\}, E_{BS}\{“A”,K_{AB}\}$
B
4: $E_{AB}\{R1-1,R2\}$
5: $E_{AB}\{R2-1\}$

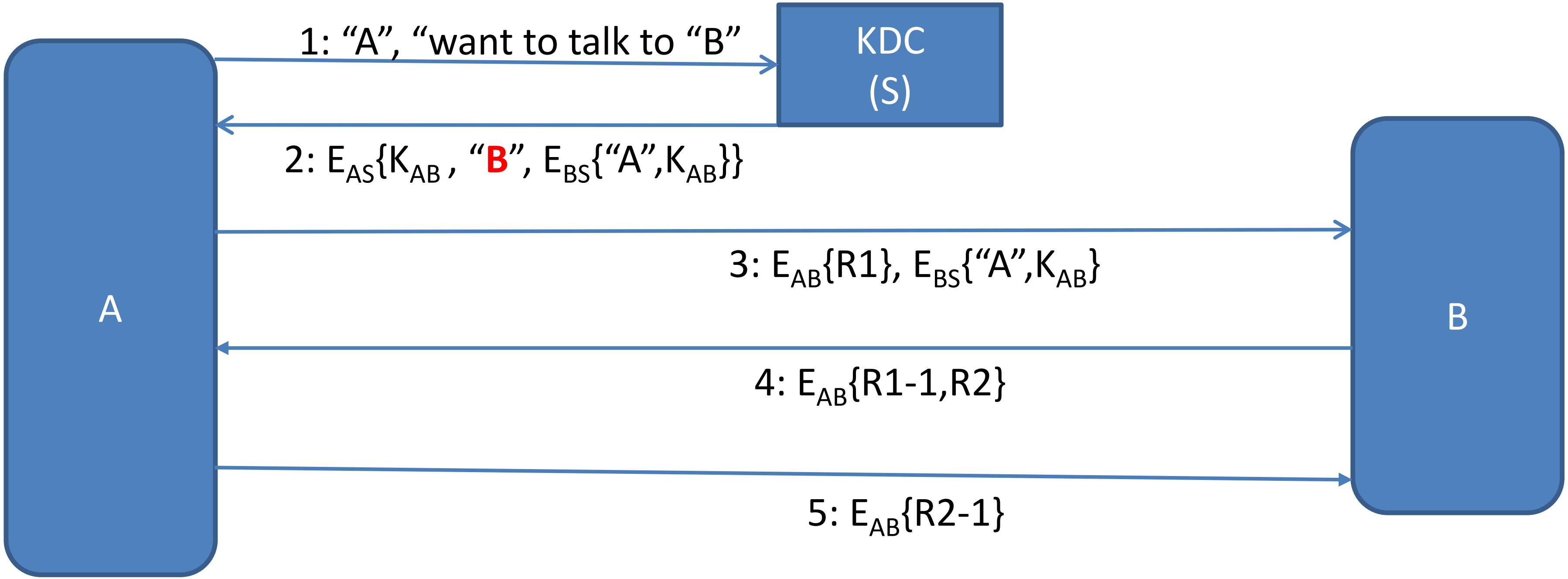
- Messages 3,4,5: challenge-response

MITM Attack on V1: M impersonates B


A
1: "A", "want to talk to "B"
M
1': "A", "want to talk to "M"
KDC
(S)
2: $E_{AS}\{K_{AM}, E_{MS}\{\text{"A"},K_{AM}\}\}$
3: $E_{AM}\{R1\}, E_{MS}\{\text{"A"},K_{AM}\}$
4: $E_{AM}\{R1\text{-}1,R2\}$
5: $E_{AM}\{R2\text{-}1\}$
M
B

- M impersonates B to A
- Message 2 is the problem. Include destination identity in it

Version-2



1: “A”, “want to talk to “B”

2: E_{AS}{K_{AB} , “**B**”, E_{BS}{“A”,K_{AB}}}

3: E_{AB}{R1}, E_{BS}{“A”,K_{AB}}

4: E_{AB}{R1-1,R2}

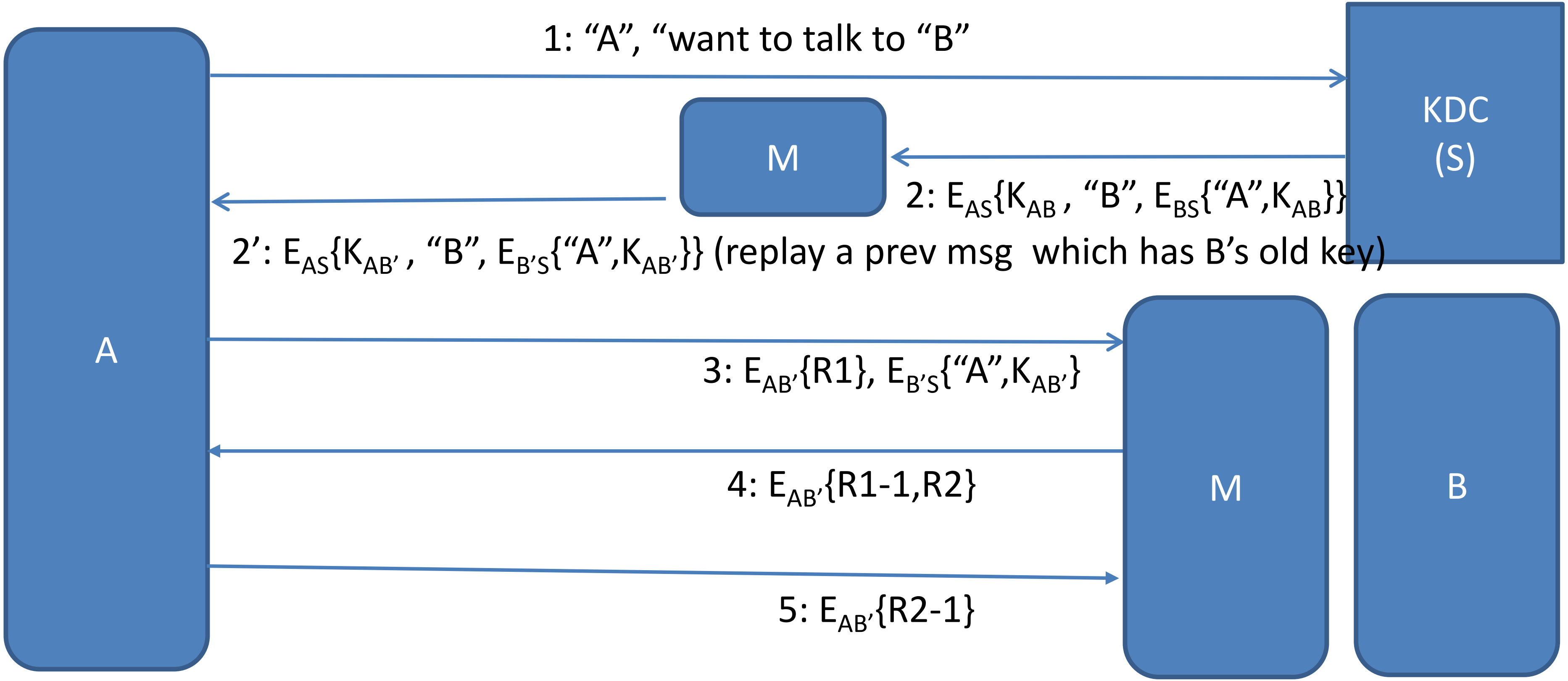
5: E_{AB}{R2-1}

- Done?

Lost/compromised keys

- User’s key compromised
- User gets a new key

MITM attack on V2: M impersonates B



1: “A”, “want to talk to “B”

2: $E_{AS}\{K_{AB}$, “B”, $E_{BS}\{$“A”,$K_{AB}\}\}$

2’: $E_{AS}\{K_{AB'}$, “B”, $E_{B'S}\{$“A”,$K_{AB'}\}\}$ (replay a prev msg which has B’s old key)

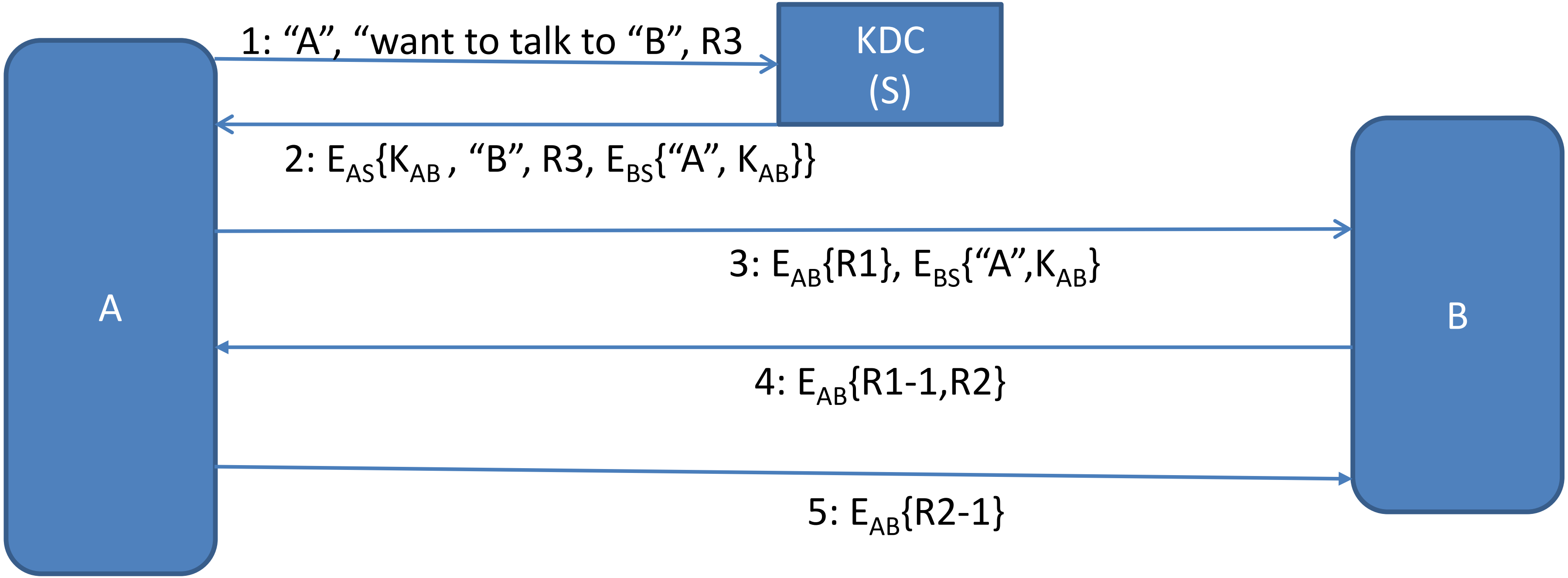
3: $E_{AB'}\{R1\}$, $E_{B'S}\{$“A”,$K_{AB'}\}$

4: $E_{AB'}\{R1\text{-}1,R2\}$

5: $E_{AB'}\{R2\text{-}1\}$

- M cracked B’s old key (represented by B’); B is using a new key (represented by B)
- Prevent M from replaying message 2 → message 2 needs to be fresh

Version-3



KDC

(S)

A

B

1: “A”, “want to talk to “B”, R3

2: E_{AS}{K_{AB} , “B”, R3, E_{BS}{“A”, K_{AB}}}

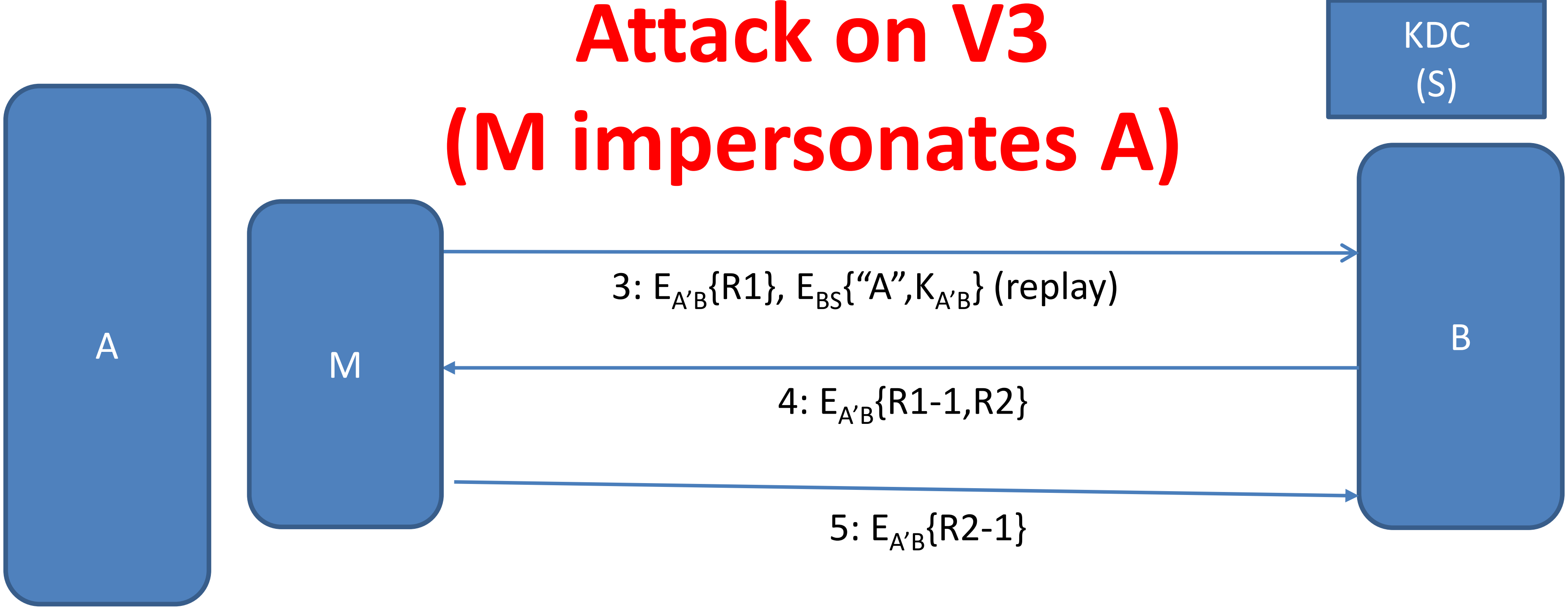
3: E_{AB}{R1}, E_{BS}{“A”,K_{AB}}

4: E_{AB}{R1-1,R2}

5: E_{AB}{R2-1}

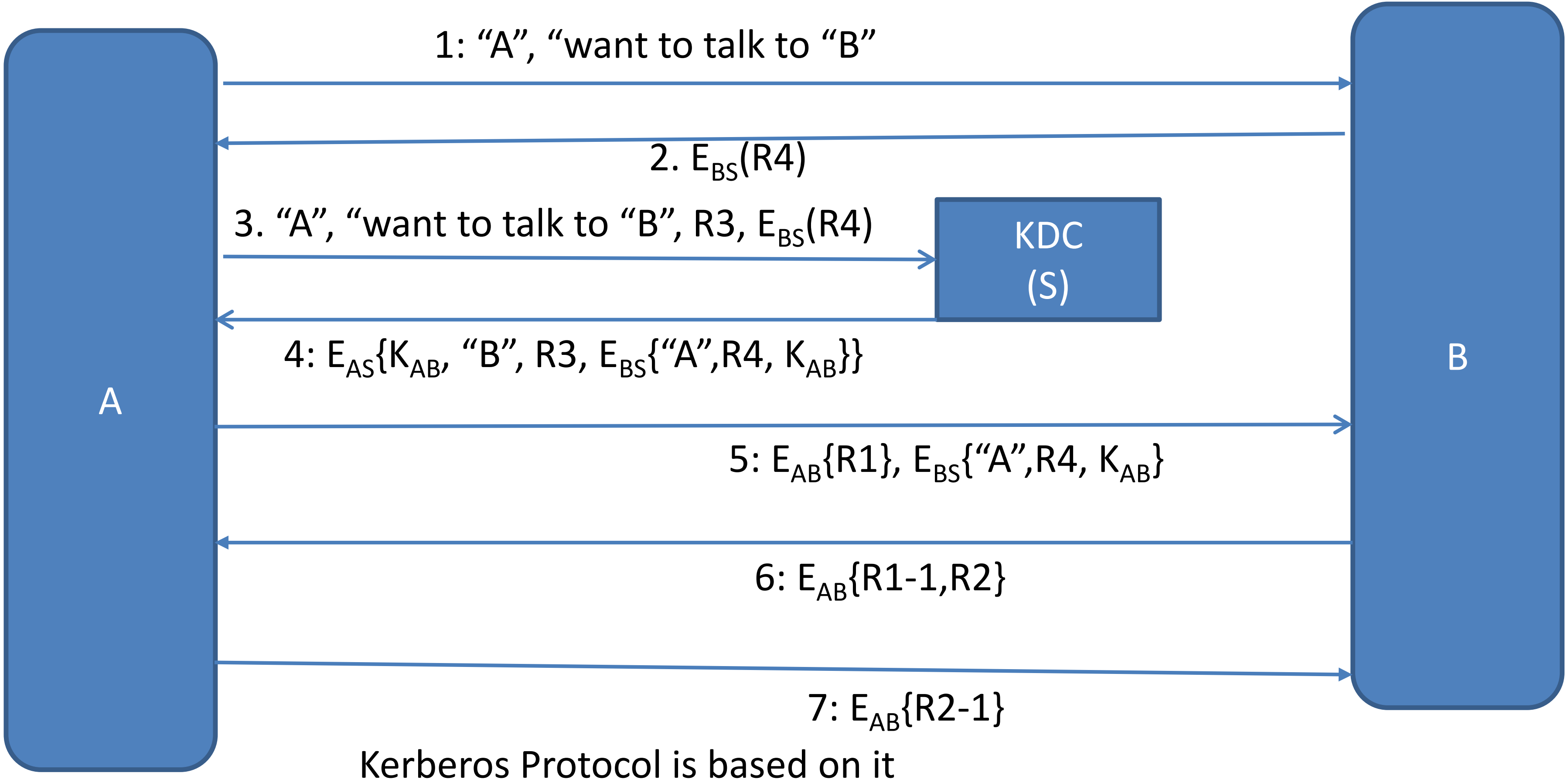
- Done?

Attack on V3
(M impersonates A)



- M cracked A's old key (represented by A')
- See version-3: M decrypted earlier message (2) from KDC using A's old key
  - Obtains key $K_{A'B}$ and E_{BS}\{"A",$K_{A'B}$\}
  - Can send message 3 above with nonce R1 encrypted with old key $K_{A'B}$
- Problem: Replay of old ticket

Needham-Schroeder Protocol



A

B

KDC
(S)

1: “A”, “want to talk to “B”

2. E_{BS}(R4)

3. “A”, “want to talk to “B”, R3, E_{BS}(R4)

4: E_{AS}{K_{AB}, “B”, R3, E_{BS}{“A”,R4, K_{AB}}}

5: E_{AB}{R1}, E_{BS}{“A”,R4, K_{AB}}

6: E_{AB}{R1-1,R2}

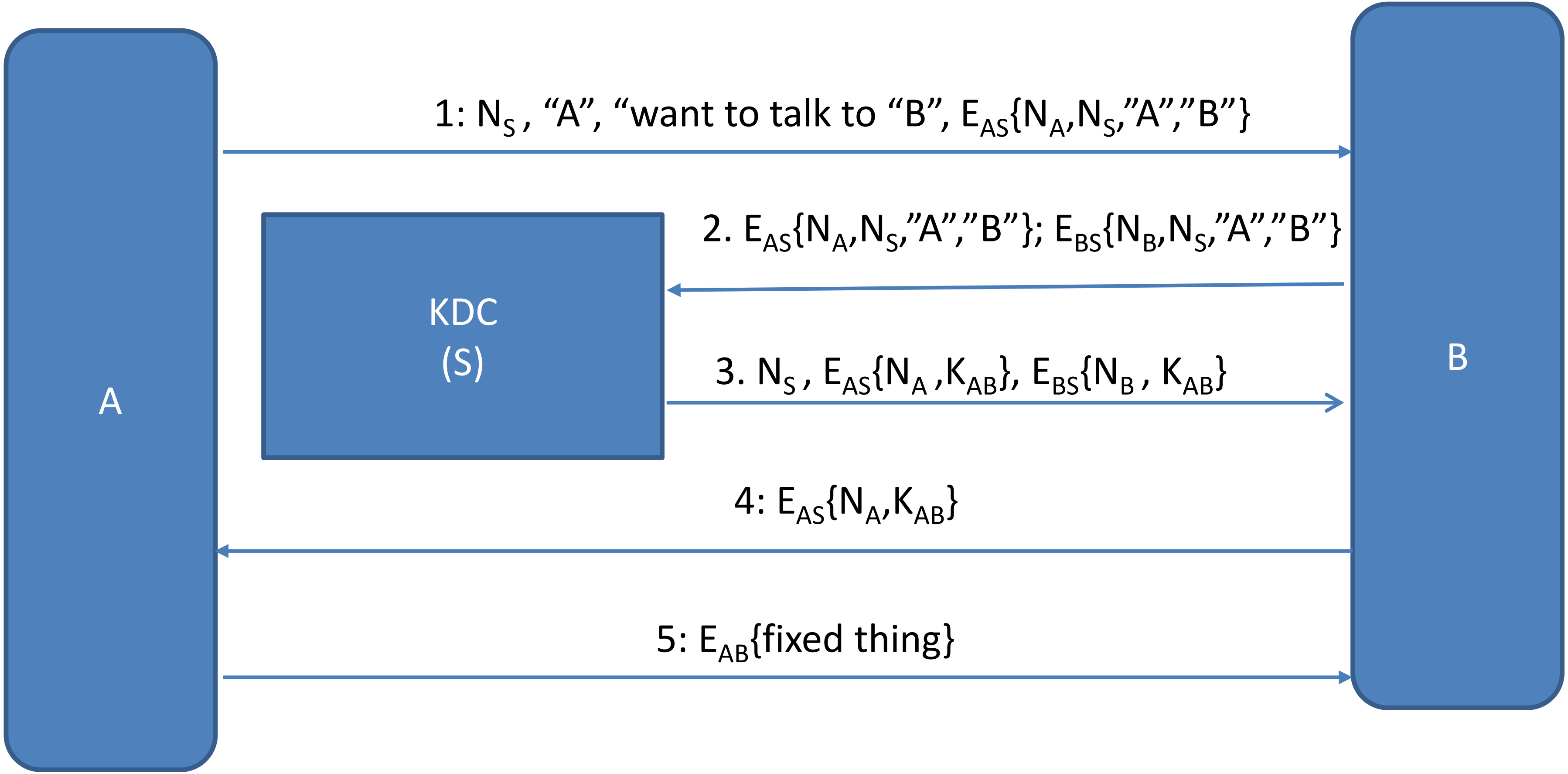
7: E_{AB}{R2-1}

Kerberos Protocol is based on it

Why does this work?

- B is challenging KDC indirectly
  - A will send the encrypted nonce from B to the KDC
  - KDC will package it in the ticket to B
  - When B get it, it knows that the ticket is fresh and has come from KDC
- If Alice changes her key,
  - M will not be able to talk to the KDC using A's old key i.e. M will not be able to get the ticket out from message 4

Otway-Rees



1: N_S , “A”, “want to talk to “B”, $E_{AS}\{N_A,N_S,”A”,”B”\}$

2. $E_{AS}\{N_A,N_S,”A”,”B”\}$; $E_{BS}\{N_B,N_S,”A”,”B”\}$

3. N_S , $E_{AS}\{N_A ,K_{AB}\}$, $E_{BS}\{N_B , K_{AB}\}$

4: $E_{AS}\{N_A,K_{AB}\}$

5: E_{AB}\{fixed thing\}

Details

- Message2: KDC authenticates Bob
  - KDC compares the N_s in both messages
  - Same means Bob is really Bob since he knows K_{BS}
- Message 3: Bob authenticates KDC (N_B in the message)
- Message 4: A authenticates Bob and KDC
  - A knows it is KDC because of NA in the message
  - A knows it is B because KDC continued the protocol
- Message5: B authenticates A
  - A shows B it knows the secret key

Verification (will not be covered)

- Protocol Correctness Verification: an active area of research
  - Belief Logic (e.g. BAN logic): based on postulates and definitions to check correctness
  - State exploration: finite state machine and exhaustive search if all reachable states are safe
  - Theorem proving: Use induction over trace of protocol execution

Summary

- Cryptographic protocols achieve security related functions based on cryptography
- Looked at key distribution as well as human, one-way, mutual and mediated authentication
- Authentication protocols notoriously hard to get right
  - Flaws often discovered many years later
  - Best to leave design to greats in the field