2550 Intro to cybersecurity

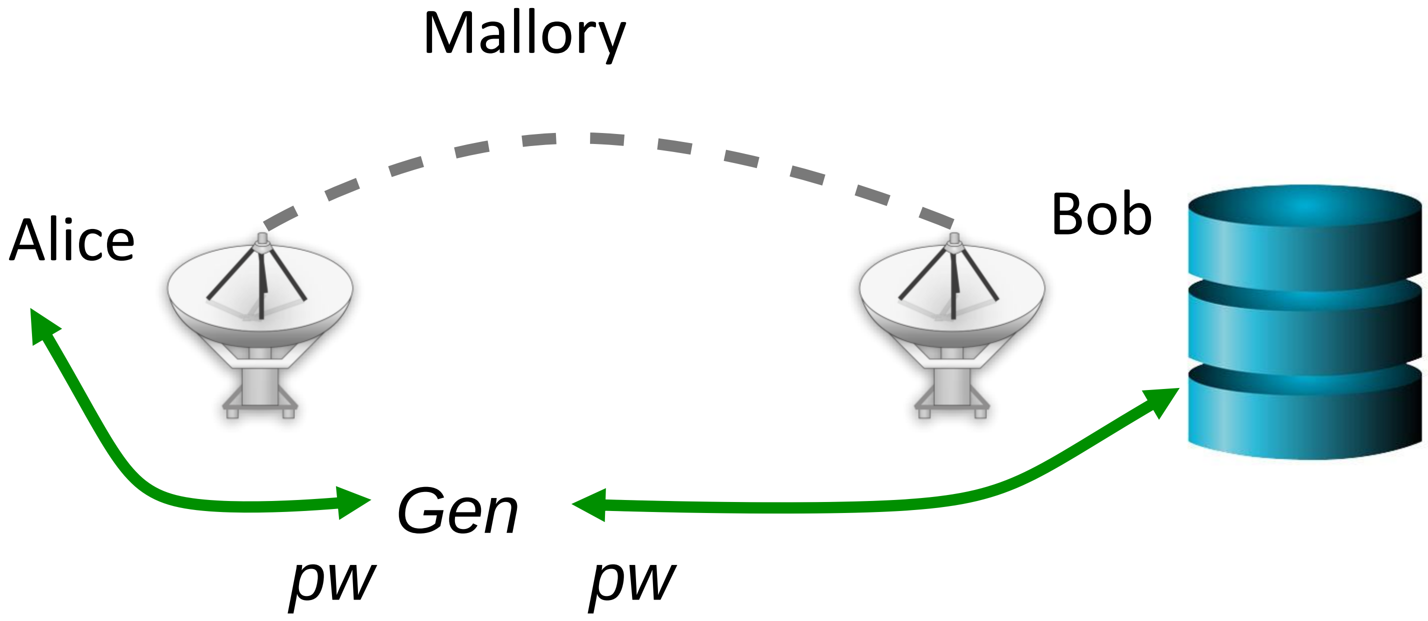
L5: Distributed Authentication

abhi shelat/Ran Cohen

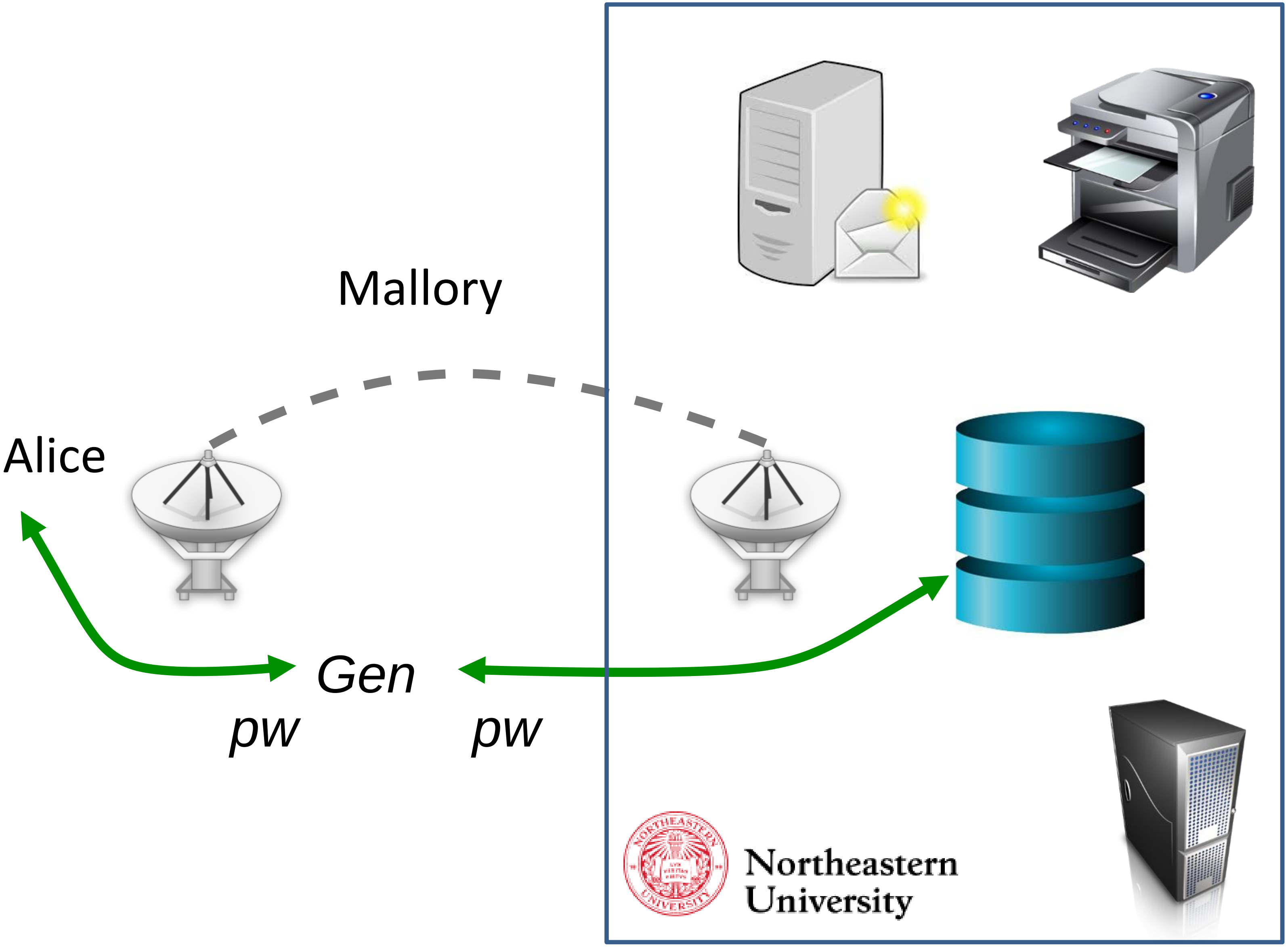
Agenda

- The problem of distributed authentication
- The Needham-Schroeder protocol
- Kerberos protocol
- Oauth

So far: authenticating to a server



Authenticating to an organization



Mallory

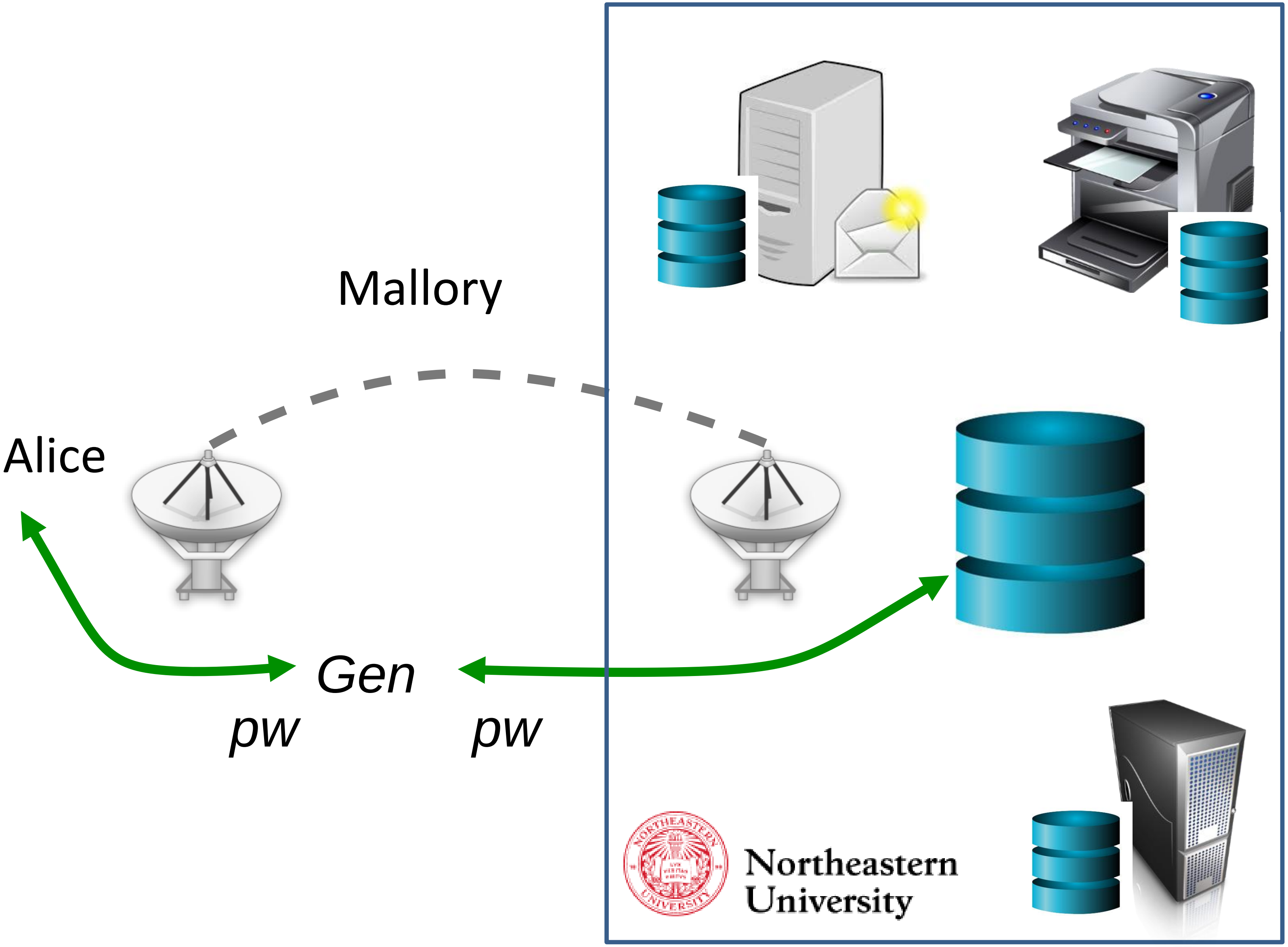
Alice

Gen

pw

pw

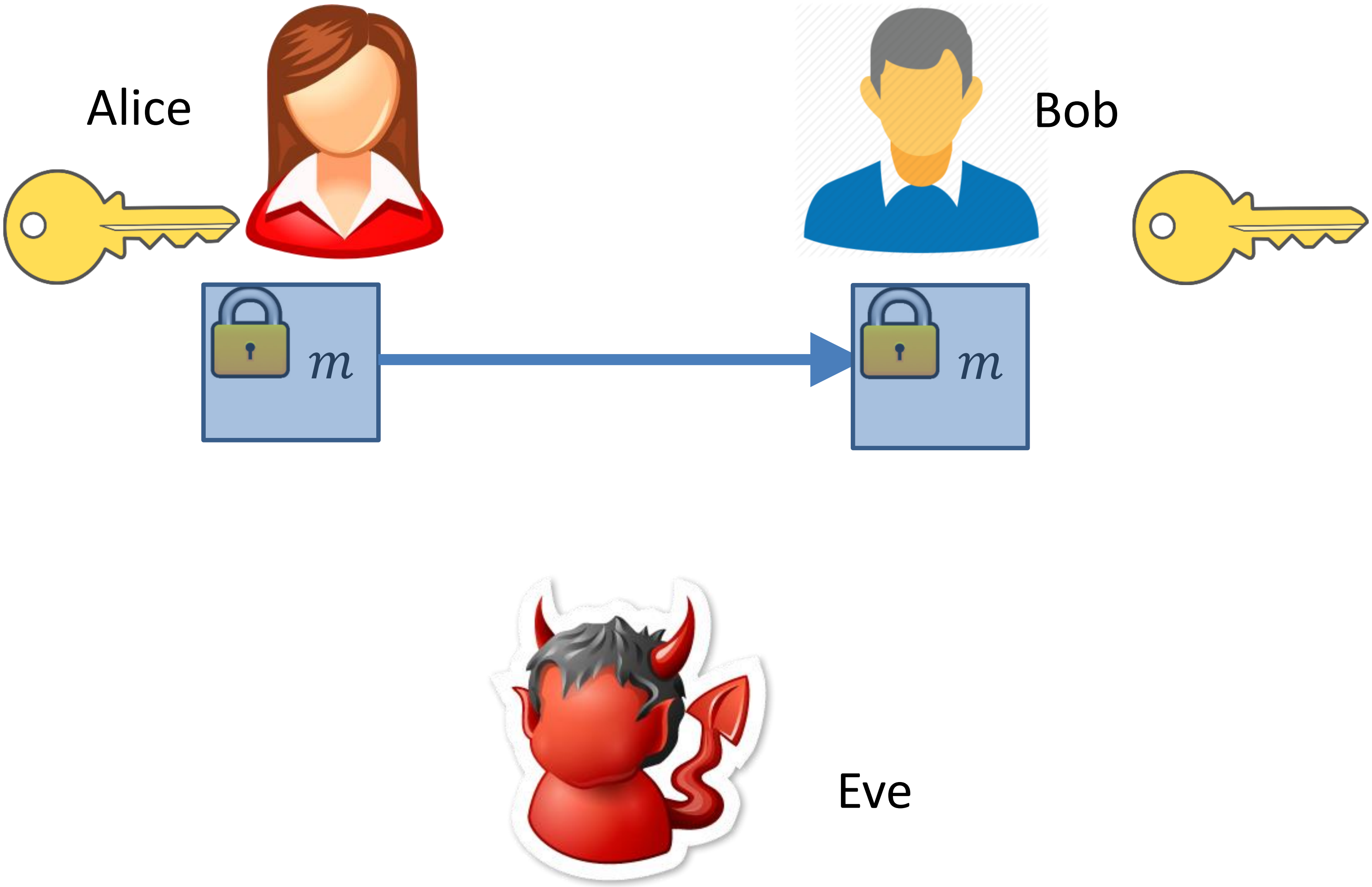
Authenticating to an organization



Distributed authentication

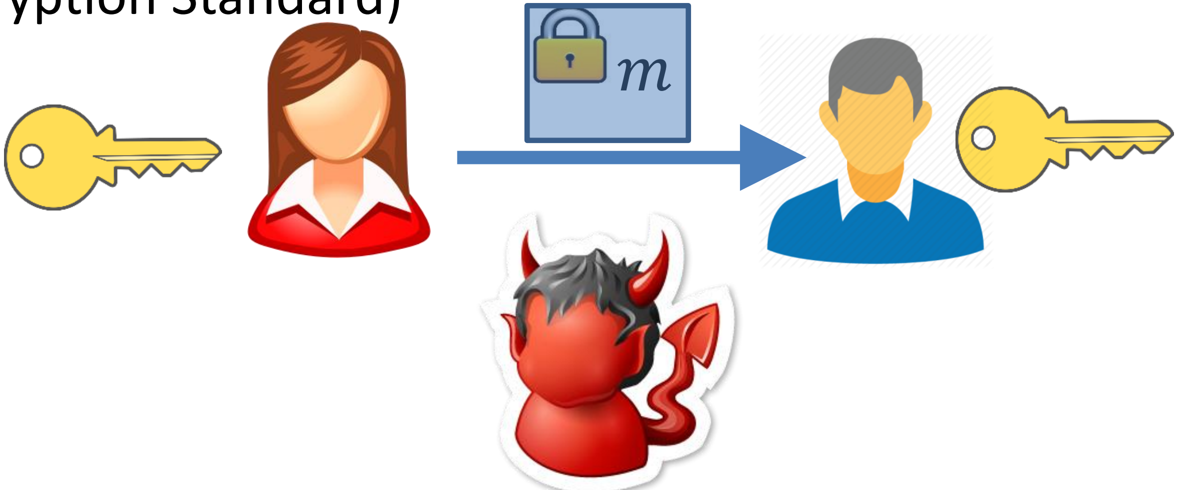
- Organizations have many entities (users/services)
- Secure communication over insecure channels
- Password-based authentication
- Passwords are never transmitted (except for the setup phase)
- Enable mutual authentication

Basic tool: symmetric encryption



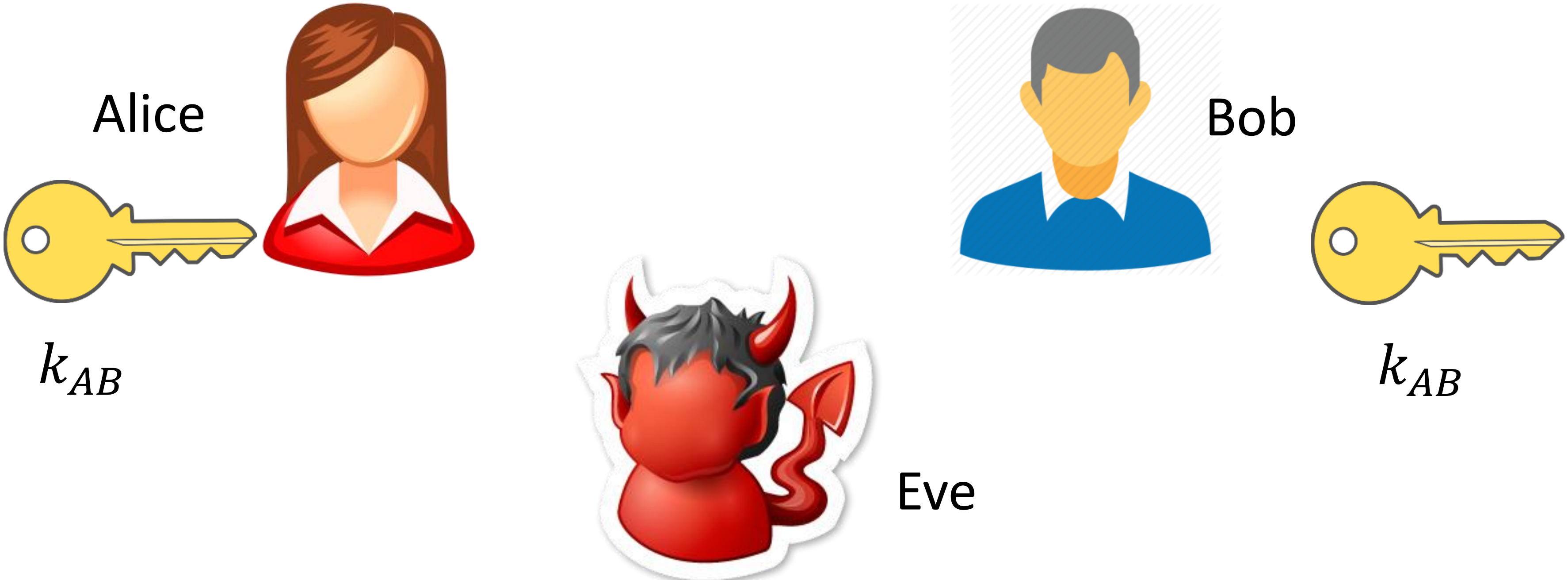
Basic tool: symmetric encryption

- Gen: generates secret key k
- Enc: given k and m output a ciphertext c
  Denote $Enc_k(m)$, $E_k(m)$, $\{m\}_k$
- Dec: given k and c output a message m
- Security (informal):
  Whatever Eve can learn on m given c can be learned without c
- Examples:
  - DES (Data Encryption Standard)
  - AES (Advanced Encryption Standard)

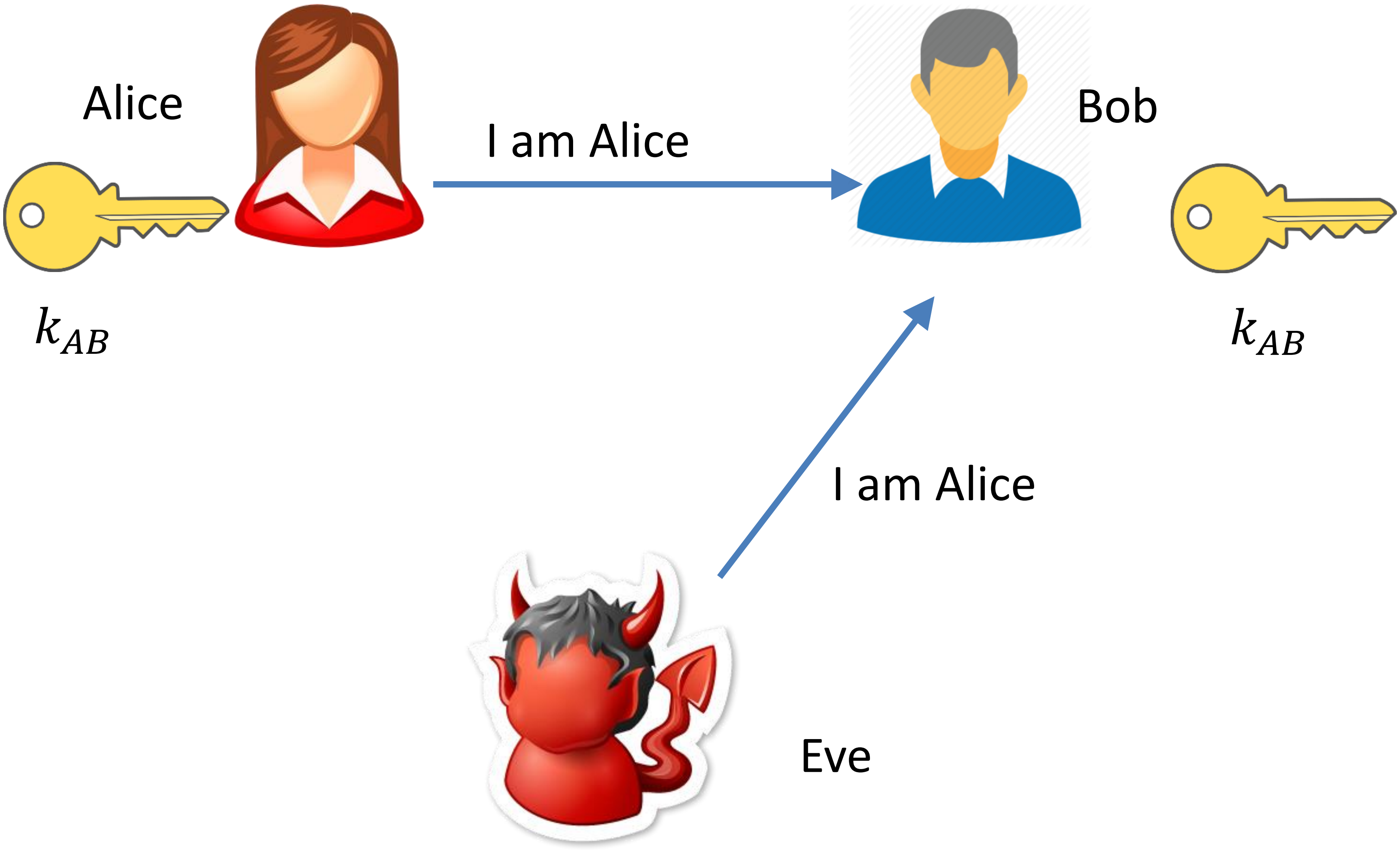


Authentication from Encryption

- Alice and Bob share a key
- They communicate over an insecure channel
- Alice wants to prove her identity to Bob
- Eve's goal: impersonate Alice



Attempt #1



Alice

k_{AB}

I am Alice

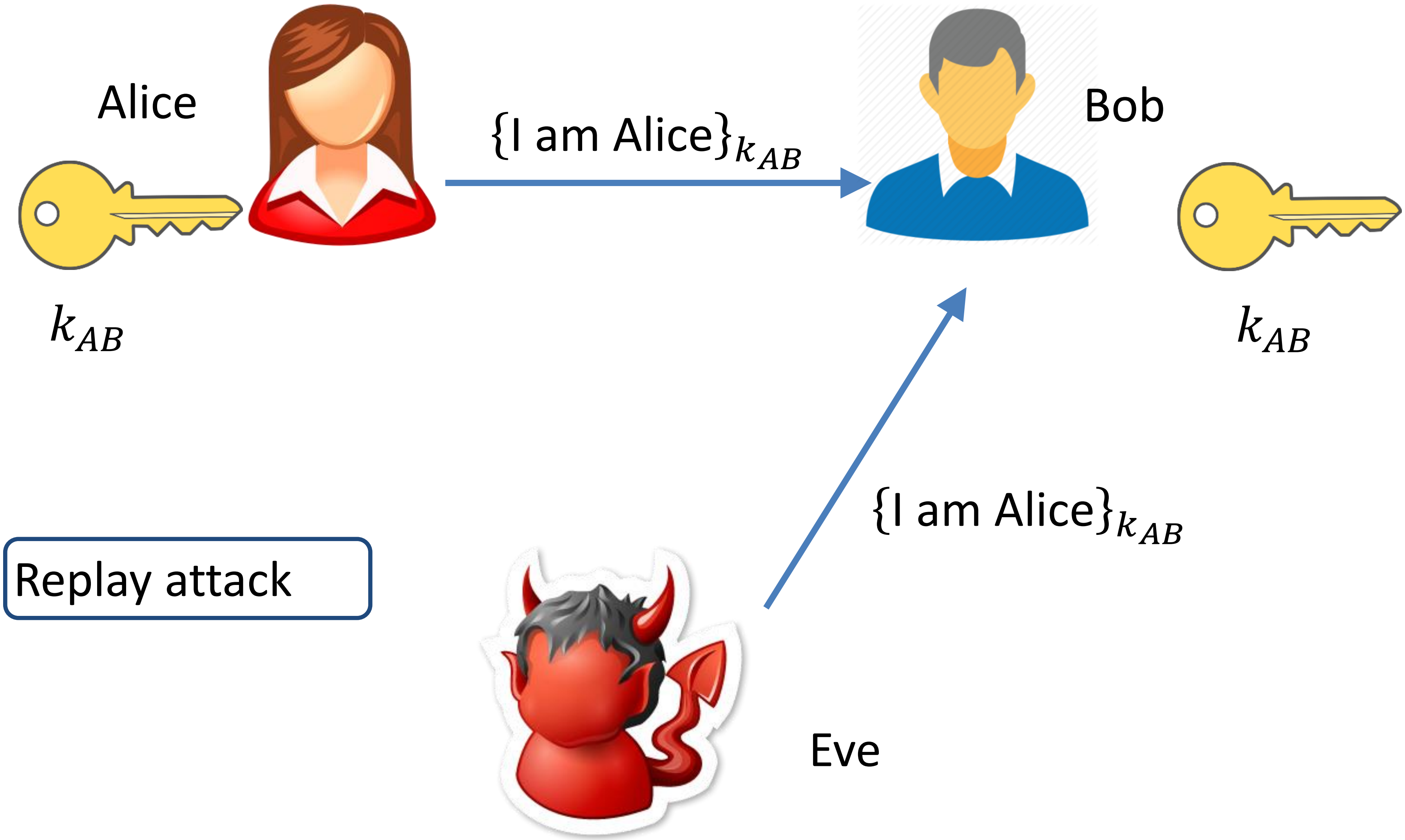
Bob

k_{AB}

I am Alice

Eve

Attempt #2: use the key



Alice

{I am Alice}$_{k_{AB}}$

Bob

k_{AB}

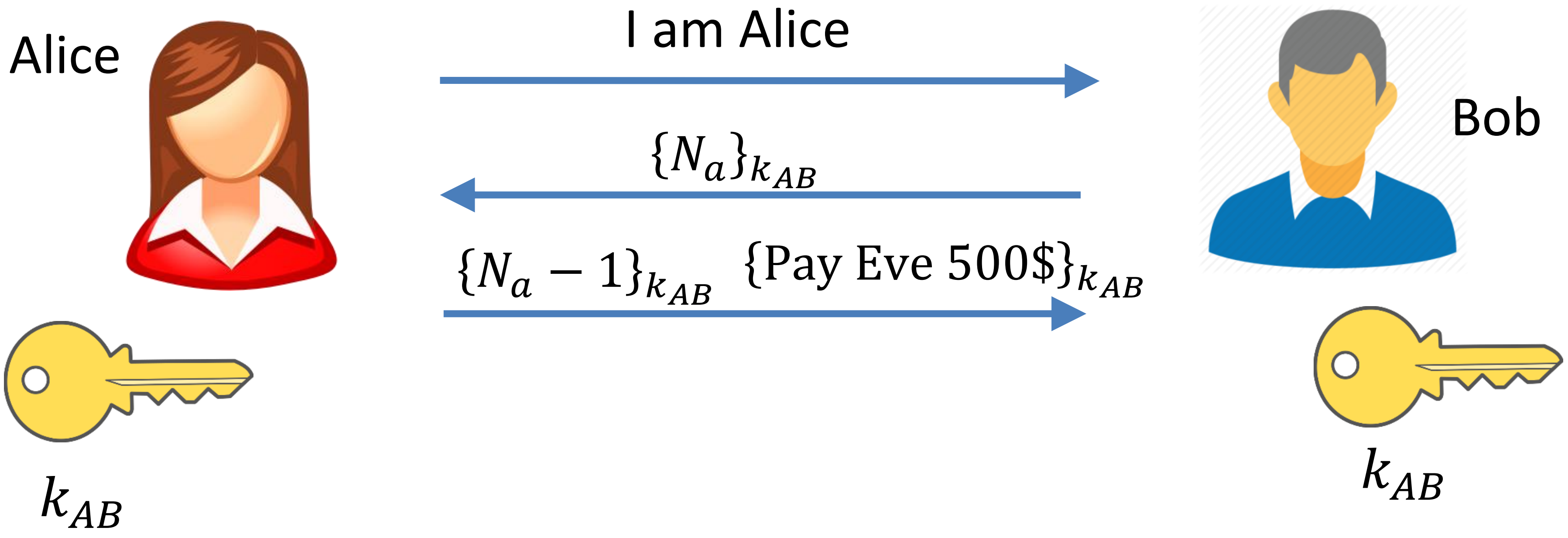
k_{AB}

{I am Alice}$_{k_{AB}}$

Eve

Replay attack

Attempt #3: use nonce



Alice

Bob

I am Alice

$\{N_a\}_{k_{AB}}$

$\{N_a - 1\}_{k_{AB}}$ $\{\text{Pay Eve 500\$}\}_{k_{AB}}$

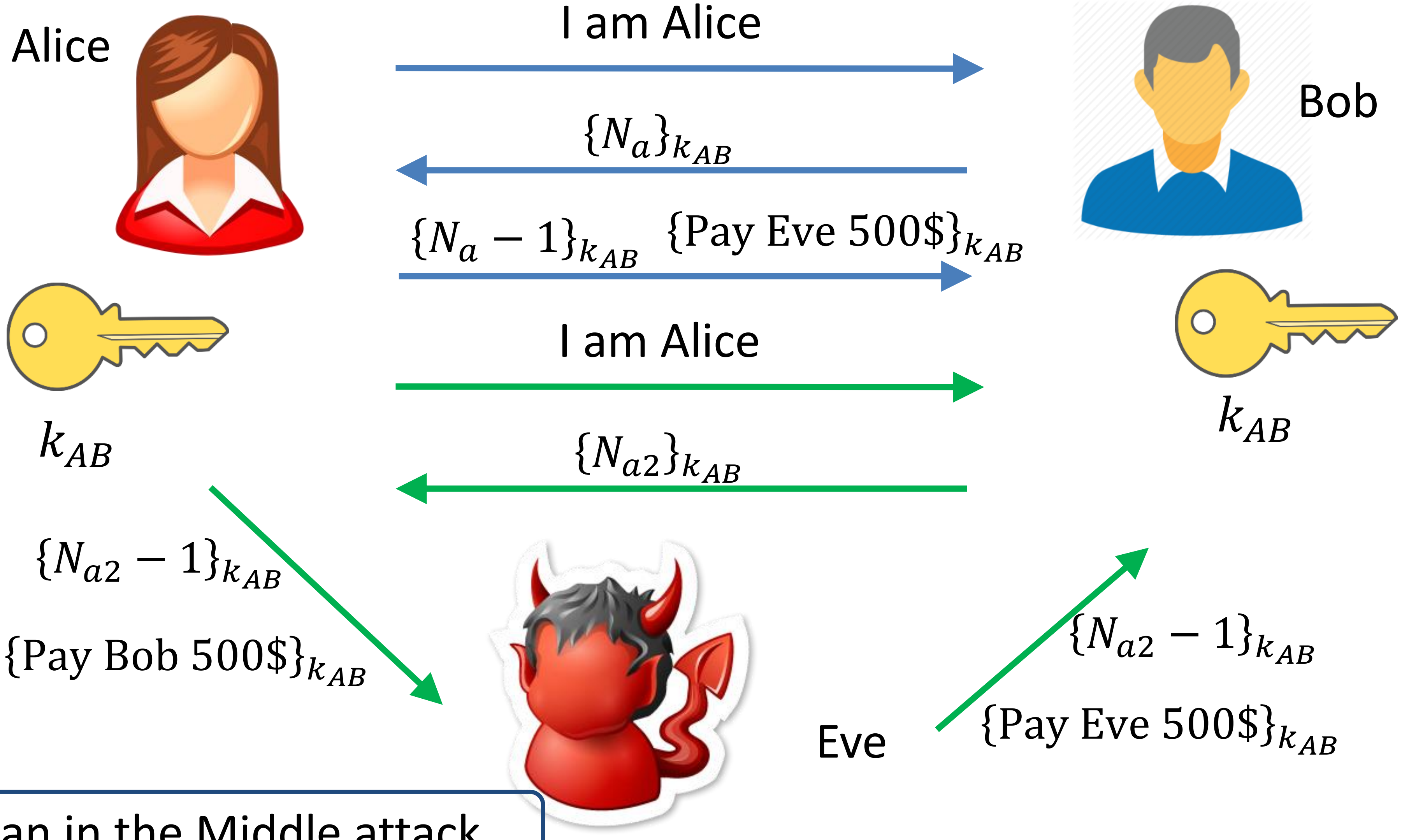
k_{AB}

k_{AB}

Nonce:
a random number
for a one-time use

Eve

Attempt #3: use nonce



Alice

I am Alice

Bob

$\{N_a\}_{k_{AB}}$

$\{N_a - 1\}_{k_{AB}}$ $\{\text{Pay Eve } 500\$\}_{k_{AB}}$

k_{AB}

I am Alice

k_{AB}

$\{N_{a2}\}_{k_{AB}}$

$\{N_{a2} - 1\}_{k_{AB}}$

$\{\text{Pay Bob } 500\$\}_{k_{AB}}$

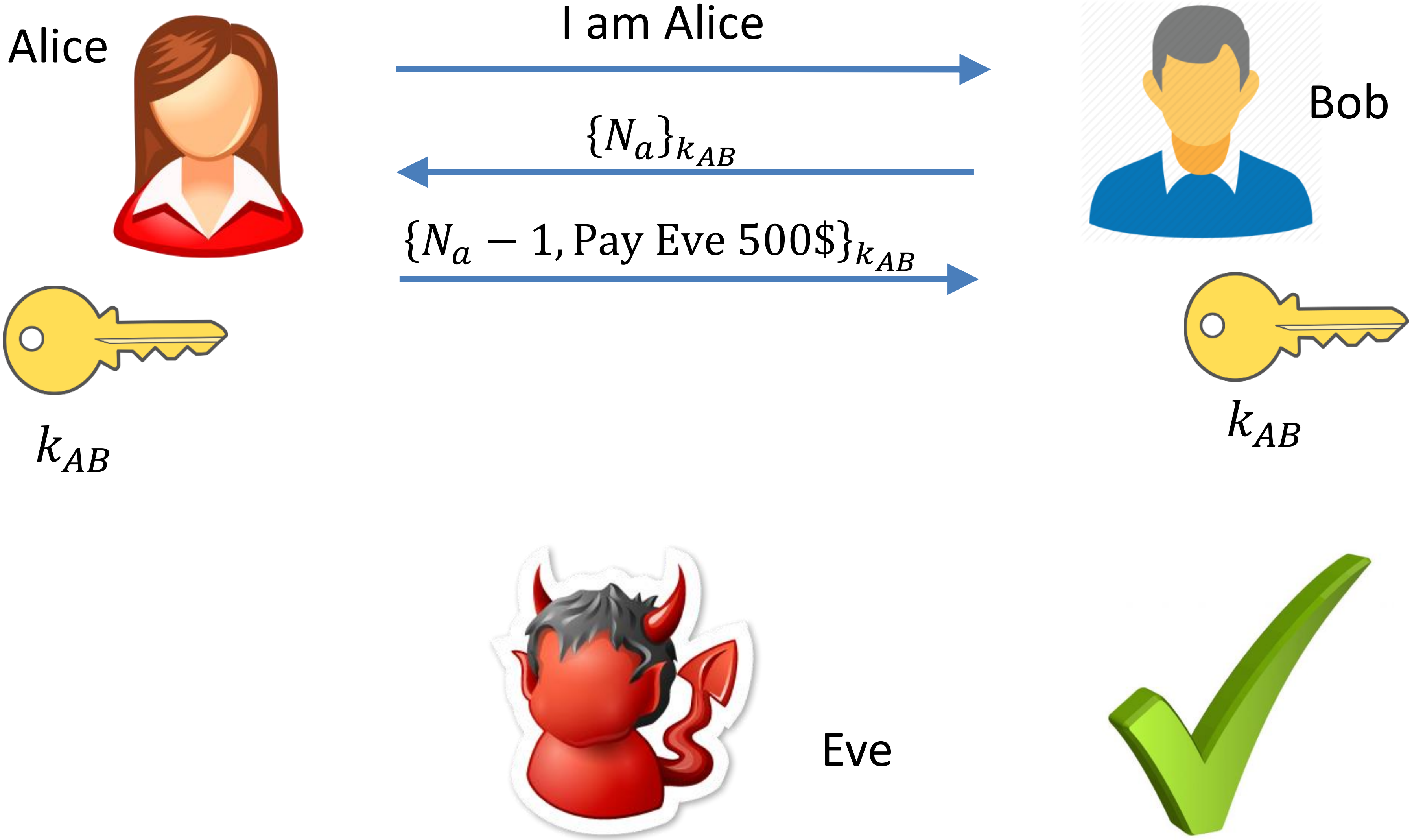
$\{N_{a2} - 1\}_{k_{AB}}$

Eve

$\{\text{Pay Eve } 500\$\}_{k_{AB}}$

Man in the Middle attack

Attempt #4



Alice

I am Alice

$\{N_a\}_{k_{AB}}$

$\{N_a - 1, \text{Pay Eve } 500\$\}_{k_{AB}}$

Bob

k_{AB}

k_{AB}

Eve

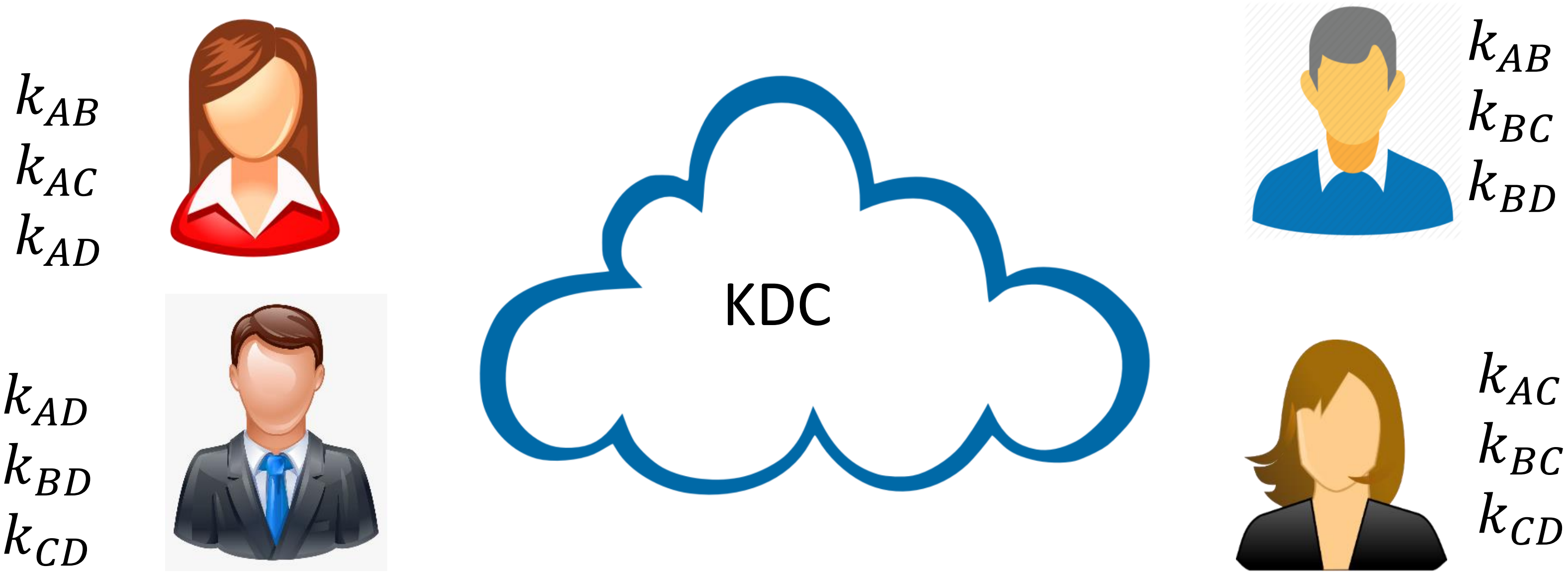
Key establishment

- The protocol worked because Alice and Bob shared a key
- How do parties agree on a key?
  - Run a key agreement protocol (later in the semester)
  - Use a trusted third party (this lecture)

- Key distribution center (KDC):
  - Shares a key with each entity
  - Single point of failure
  - Reasonable assumption for organizations
  - Not useful for open environments (e.g. the Internet)

Naïve solution

- KDC generates a key for each pair
- Number of keys $n(n-1)$, number of key pairs $\frac{n(n-1)}{2} = \binom{n}{2}$
- Drawbacks:
  - Quadratic number of keys
  - Adding new users is complex
- May be useful for static small networks



k_{AB}
k_{AC}
k_{AD}

k_{AB}
k_{BC}
k_{BD}

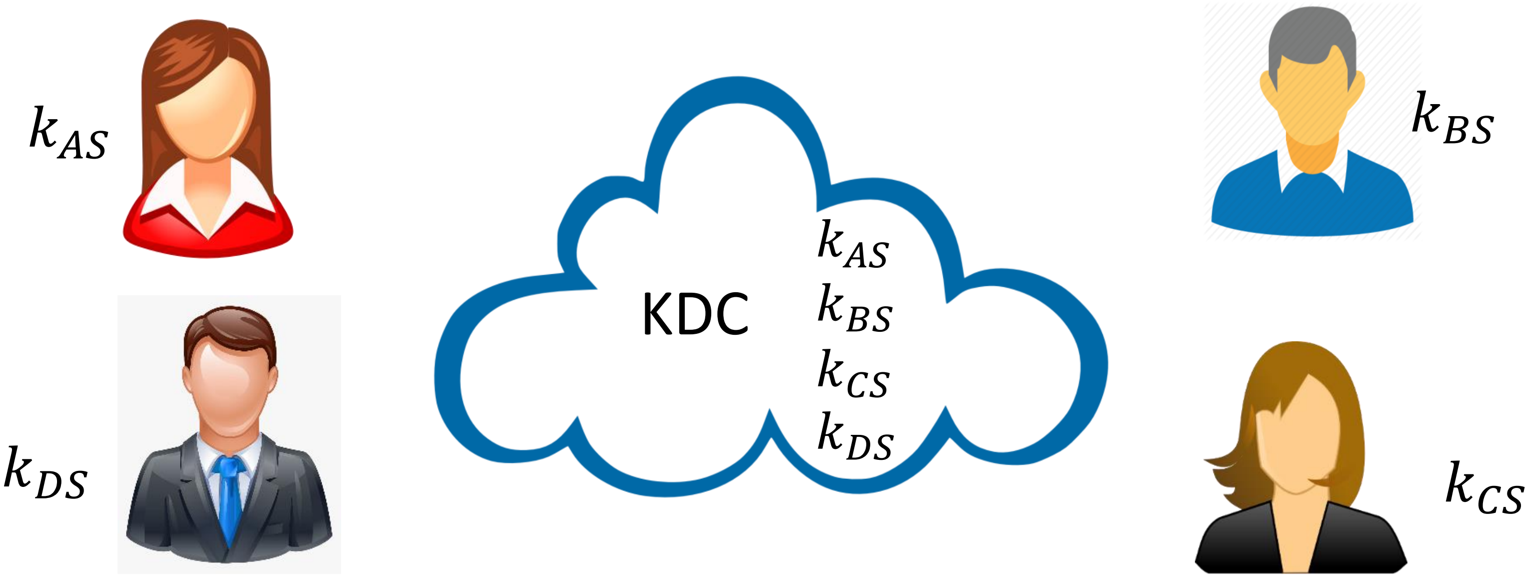
KDC

k_{AD}
k_{BD}
k_{CD}

k_{AC}
k_{BC}
k_{CD}

Desire: solution with linear keys

- KDC shares a key with each user
- Number of keys $2n$
- Number of key pairs n
- These are long-term keys
- Alice and Bob establish a fresh session key



k_{AS}

k_{BS}

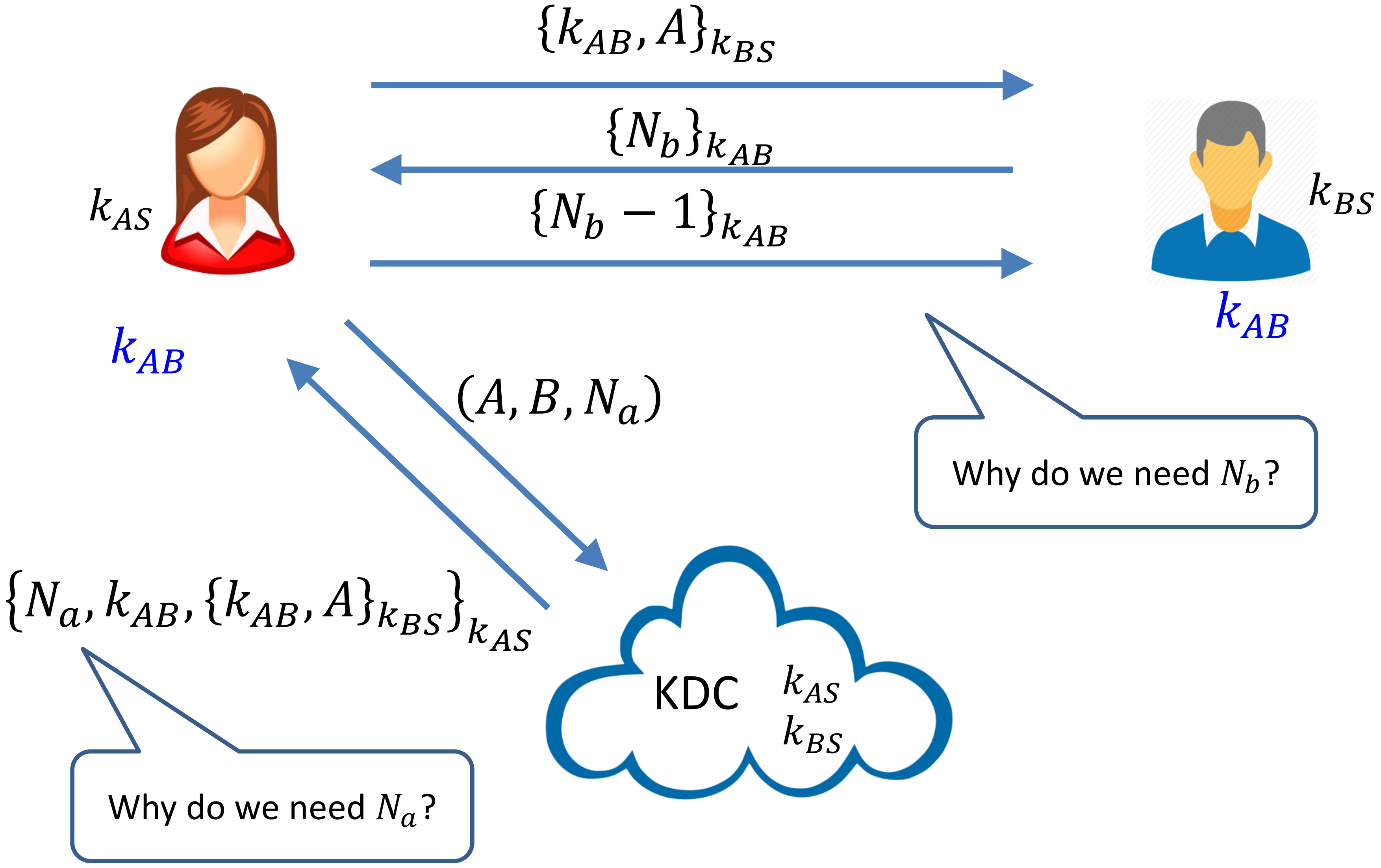
KDC

k_{AS}
k_{BS}
k_{CS}
k_{DS}

k_{DS}

k_{CS}

Needham-Schroeder Protocol (1978)



$\{k_{AB}, A\}_{k_{BS}}$

$\{N_b\}_{k_{AB}}$

$\{N_b - 1\}_{k_{AB}}$

k_{AS}

k_{BS}

k_{AB}

k_{AB}

(A, B, N_a)

Why do we need N_b?

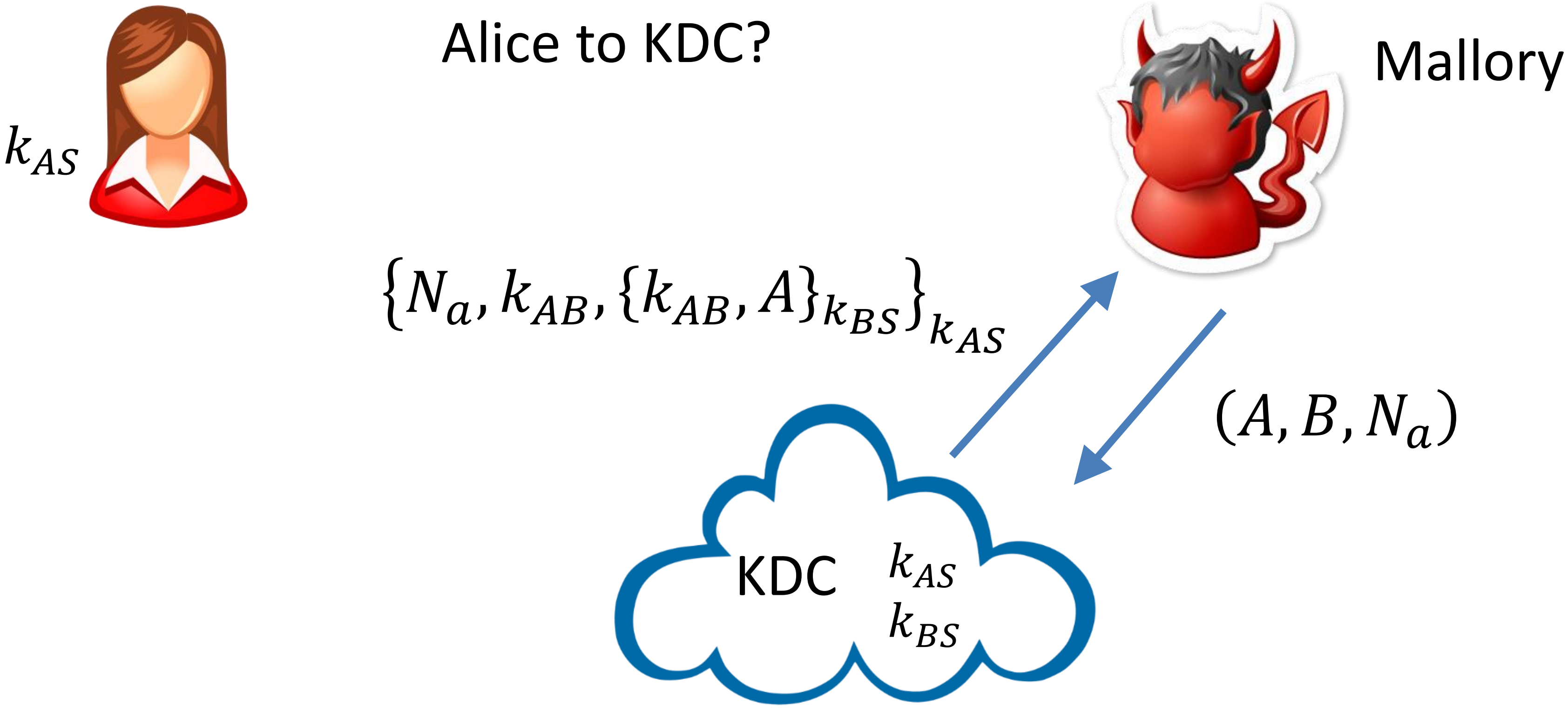
$\{N_a, k_{AB}, \{k_{AB}, A\}_{k_{BS}}\}_{k_{AS}}$

KDC k_{AS} k_{BS}

Why do we need N_a?

Is Needham-Schroeder secure?

Can Mallory impersonate Alice to KDC?



k_{AS}

Mallory

$\{N_a, k_{AB}, \{k_{AB}, A\}_{k_{BS}}\}_{k_{AS}}$

(A, B, N_a)

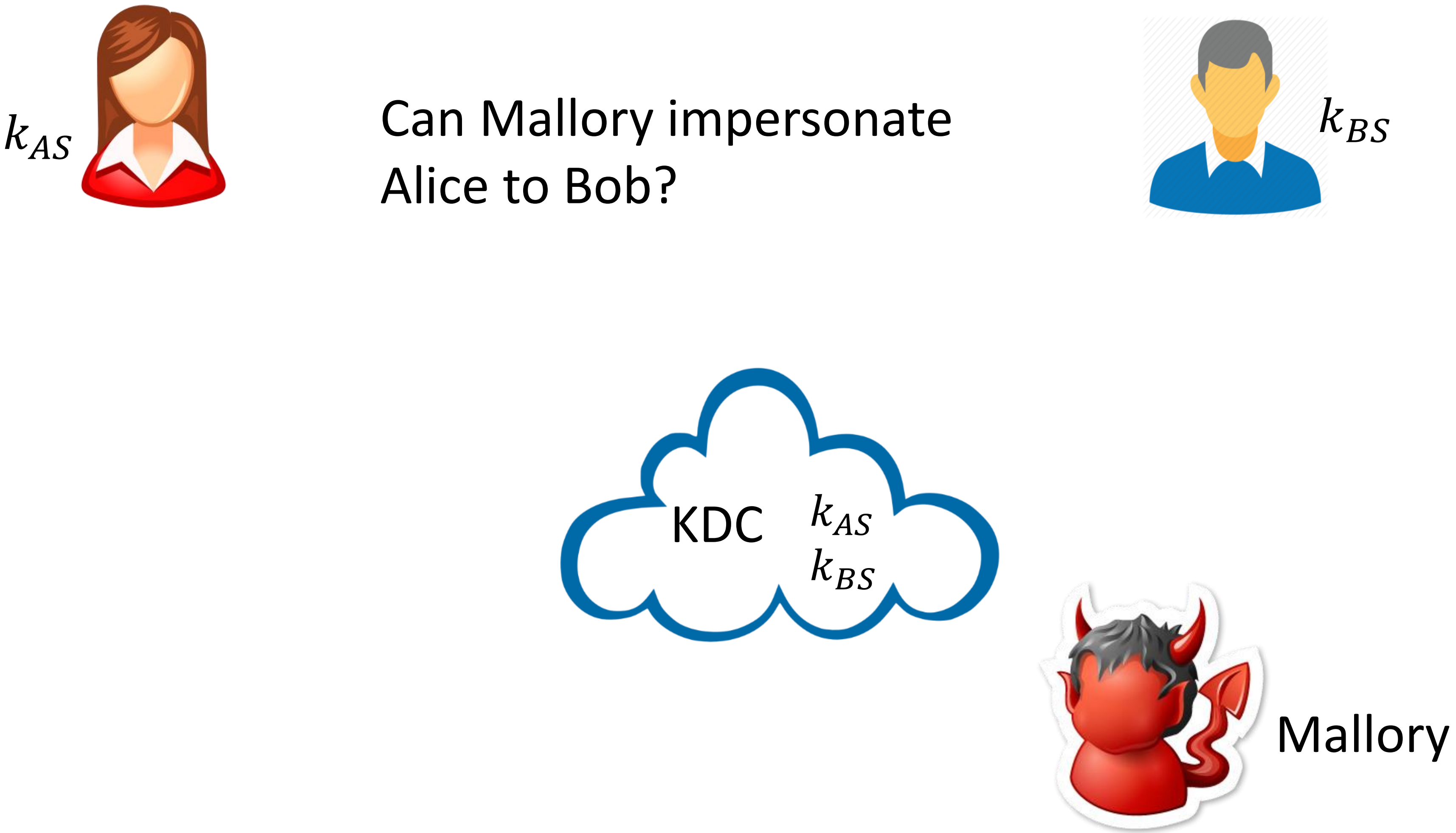
KDC

k_{AS}

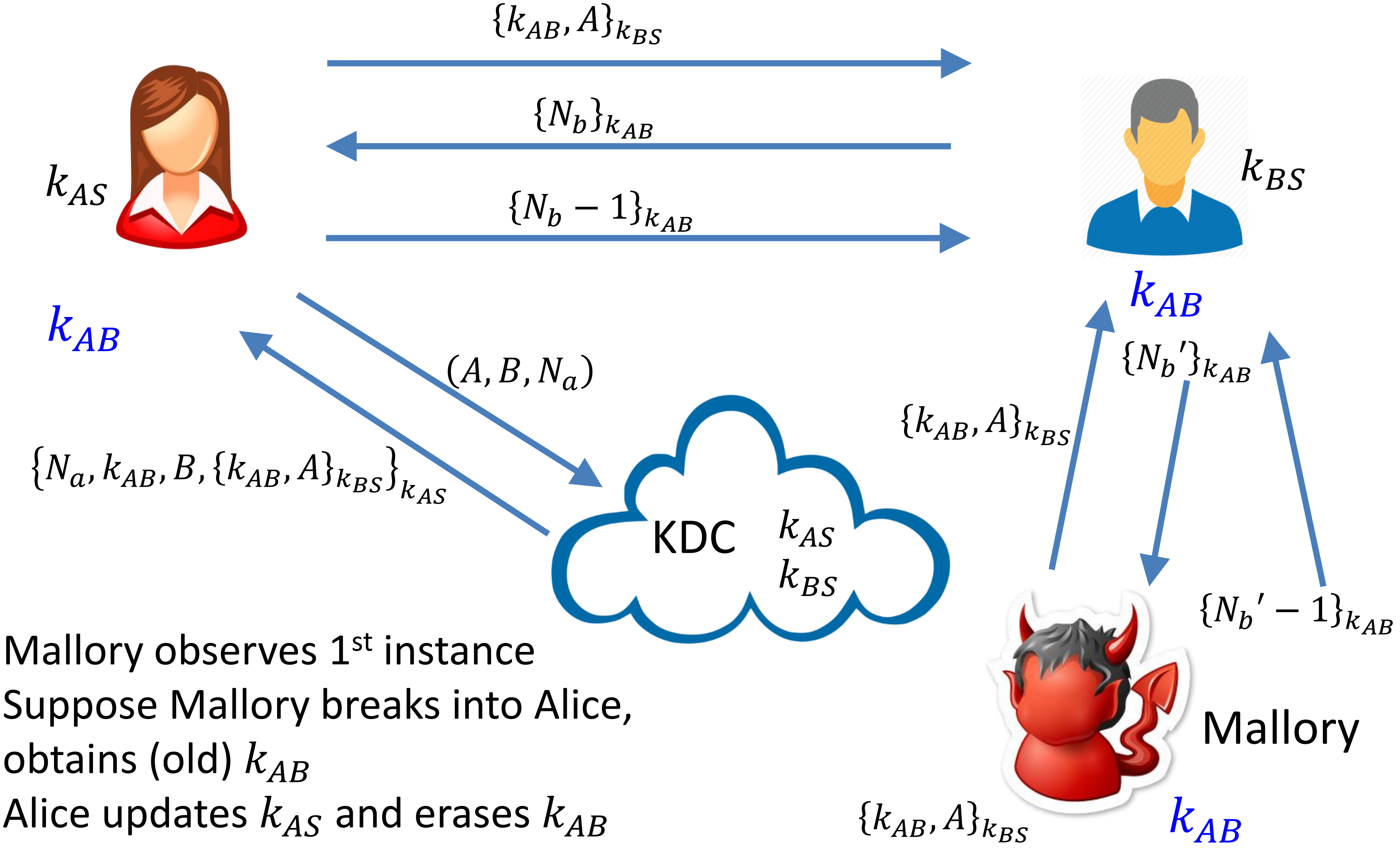
k_{BS}

Is Needham-Schroeder secure?

Can Mallory impersonate Alice to Bob?

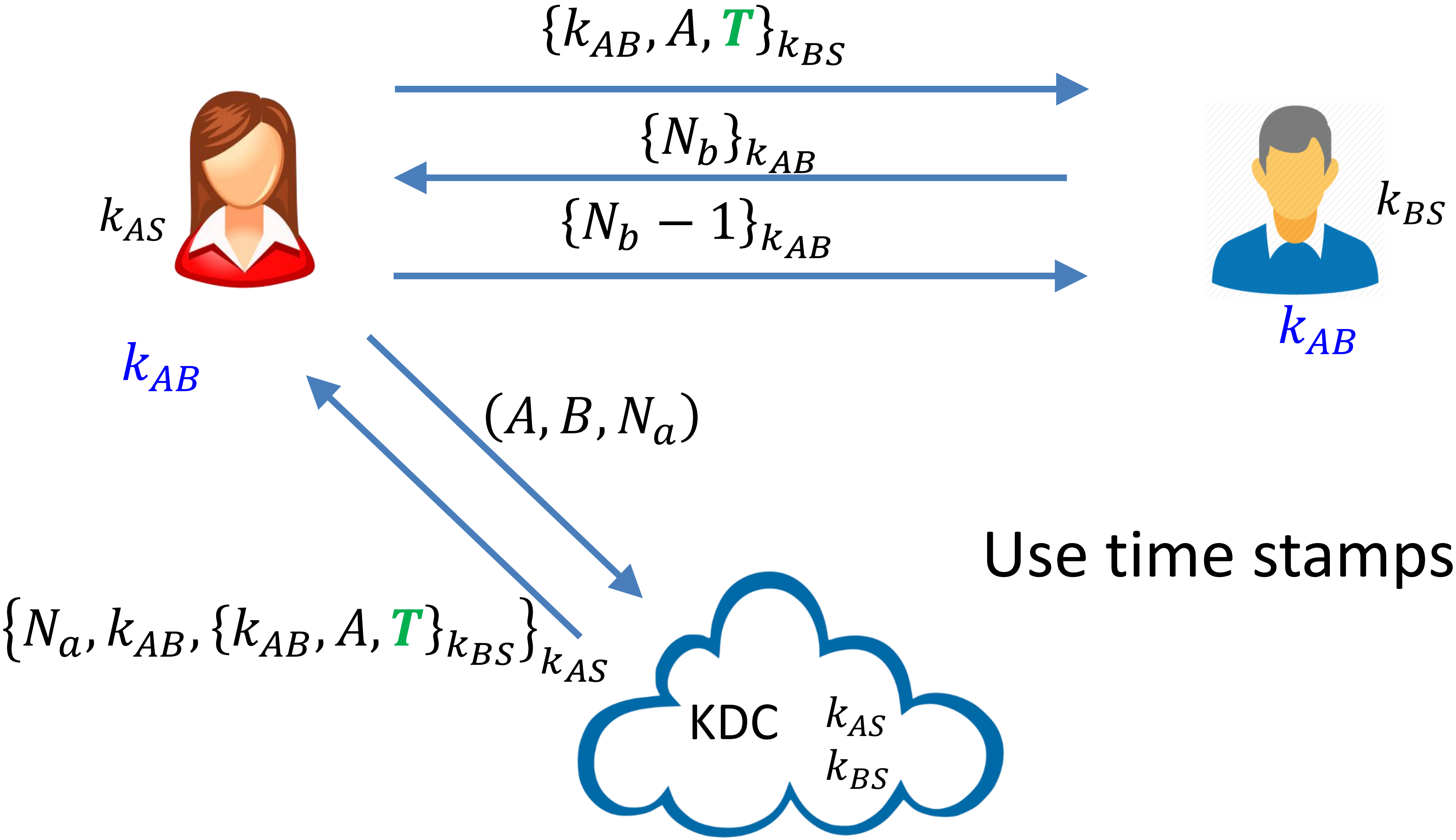


Needham-Schroeder replay attack



1) Mallory observes 1st instance
2) Suppose Mallory breaks into Alice, obtains (old) k_{AB}
3) Alice updates k_{AS} and erases k_{AB}

Fixed Needham-Schroeder



$\{k_{AB}, A, \mathbf{T}\}_{k_{BS}}$

$\{N_b\}_{k_{AB}}$

$\{N_b - 1\}_{k_{AB}}$

k_{AS}

k_{AB}

k_{BS}

k_{AB}

(A, B, N_a)

$\{N_a, k_{AB}, \{k_{AB}, A, \mathbf{T}\}_{k_{BS}}\}_{k_{AS}}$

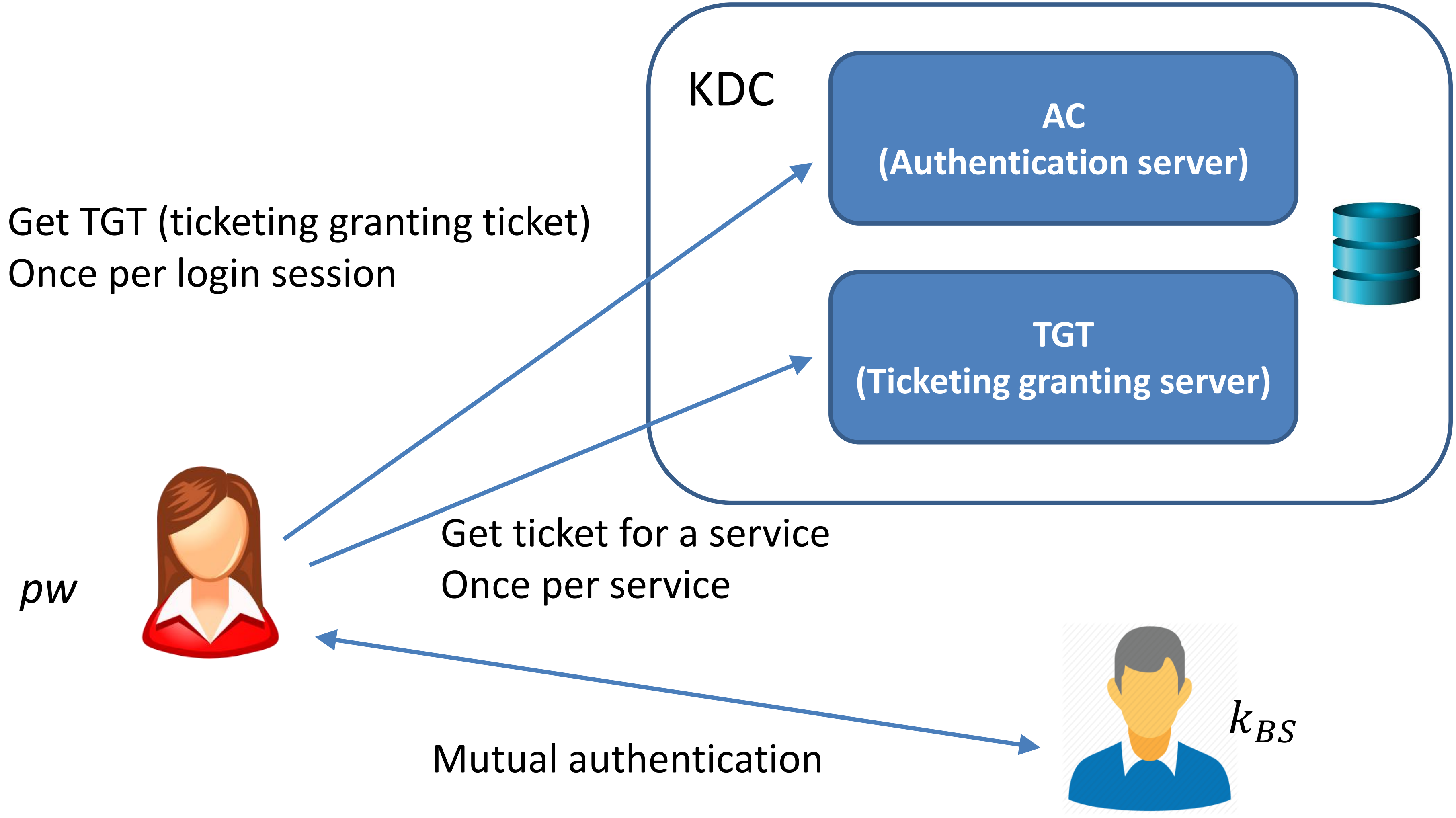
KDC k_{AS} k_{BS}

Use time stamps

Kerberos

- Developed in MIT in the ‘80s
- Based on Needham-Schroeder
  - Versions 1-3 not published
  - Version 4 not secure
  - Version 5 published in 1993
- Widely used nowadays:
  - The basis of Microsoft’s active directory
  - Many Unix versions

Kerberos



KDC

AC
(Authentication server)

TGT
(Ticketing granting server)

Get TGT (ticketing granting ticket)
Once per login session

Get ticket for a service
Once per service

pw

Mutual authentication

k_{BS}

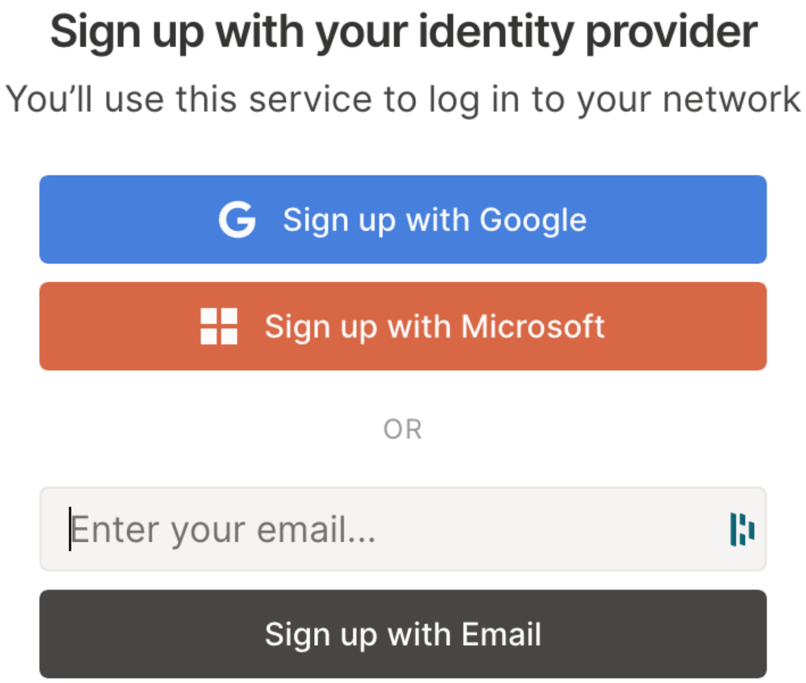
Kerberos

- Passwords are not sent over the network
- Alice's key k_{AS} is a hash of her password

- Kerberos weaknesses:
  - KDC is a single point of failure
  - DoS the KDC and the network ceases to function
  - Compromise the KDC leads to network-wide compromise
  - Time synchronization is a very hard problem

Access delegation (valet key)

“Single Sign on”

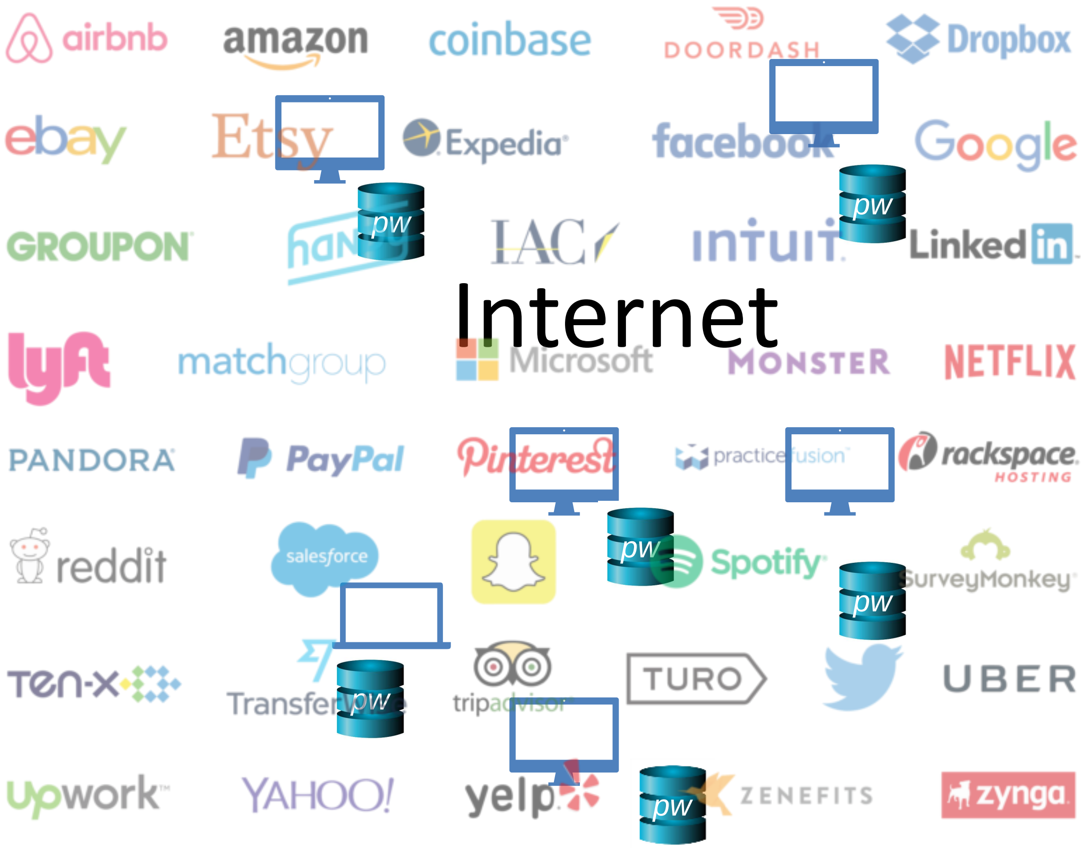


Sign up with your identity provider

You’ll use this service to log in to your network

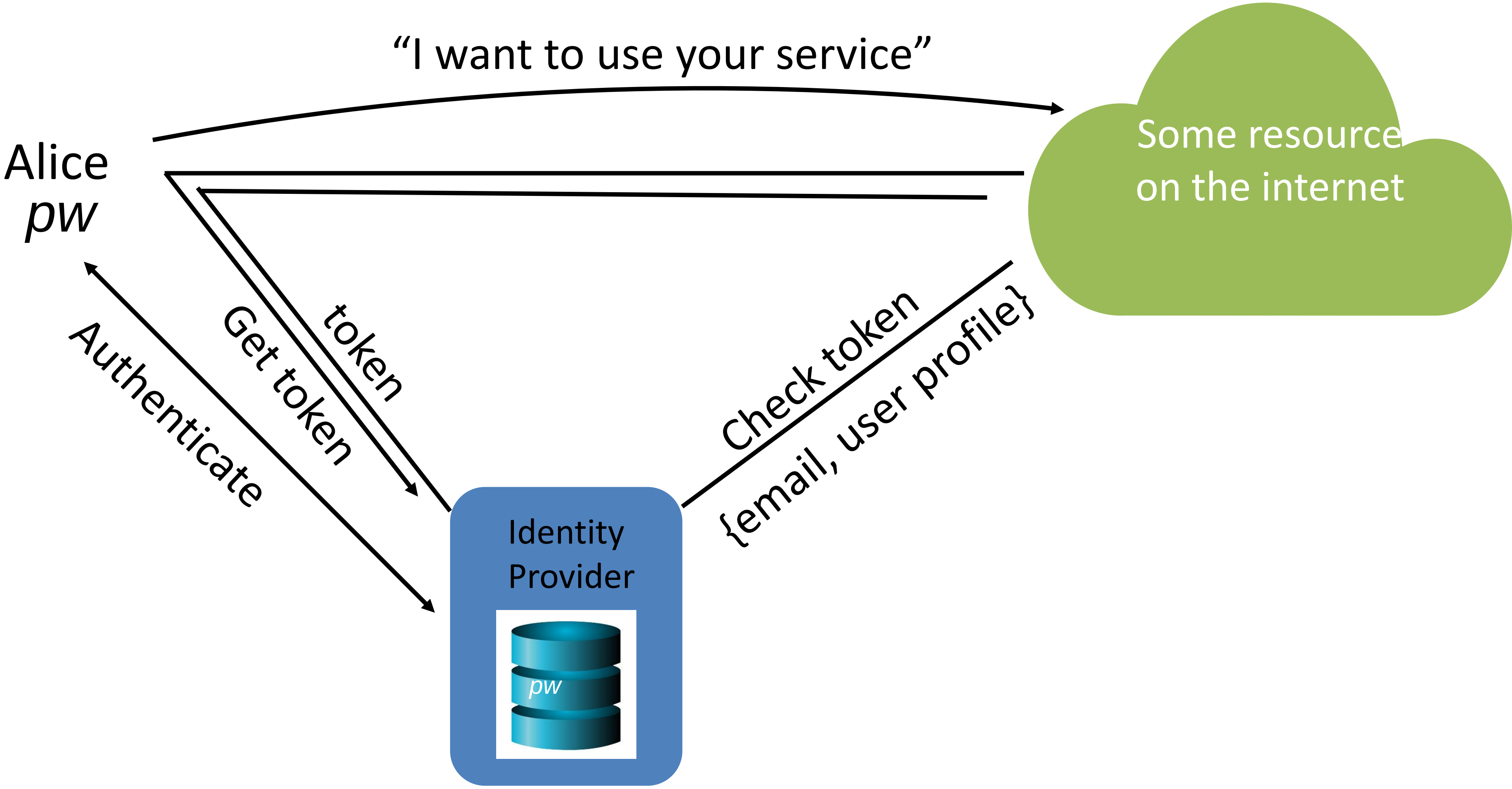
Same problem as before

Alice
pw



Internet

OAuth



"I want to use your service"

Alice

pw

Some resource on the internet

Authenticate

Get token

token

Check token

{email, user profile}

Identity Provider

Recap

- Distributed authentication
- The Needham-Schroeder protocol
- Kerberos protocol
- Oauth