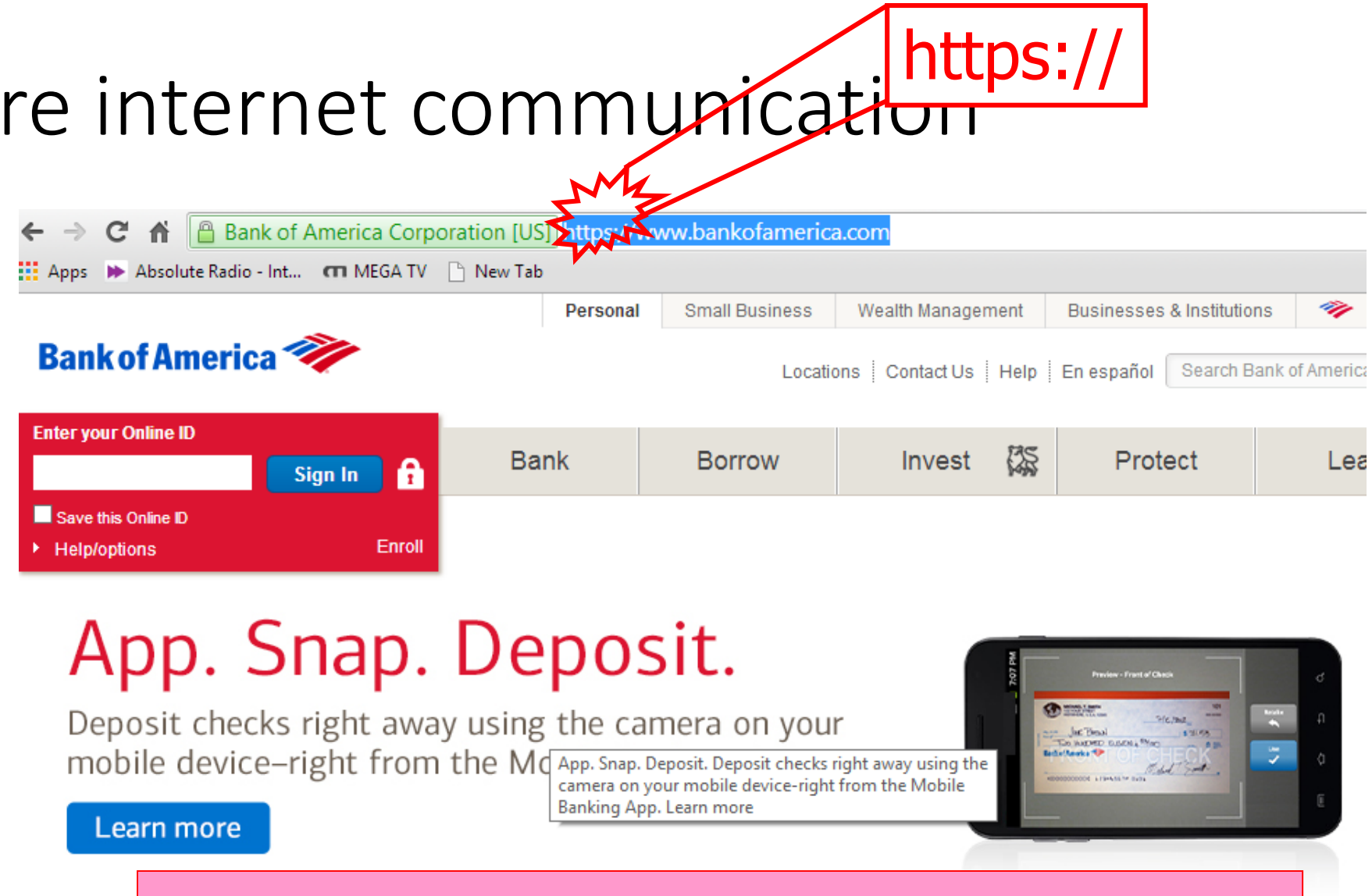
ENEE 457: Computer Systems Security

Lecture 7
Certificates and Web of Trust

Charalampos (Babis) Papamanthou

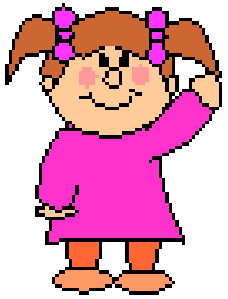
Department of Electrical and Computer Engineering
University of Maryland, College Park

Secure internet communication



What cryptographic keys are used to protect communication?

Public Keys and Trust



Public Key: P_A
Secret key: S_A

Public Key: P_B
Secret key: S_B

How are public keys stored?

How to obtain the public key?

How does Bob know or 'trusts' that P_A is Alice's public key?

Distribution of Public Keys

- **Public announcement**: users distribute public keys to recipients or broadcast to community at large

- **Publicly available directory**: can obtain greater security by registering keys with a public directory

- Both approaches have problems, and are vulnerable to forgeries

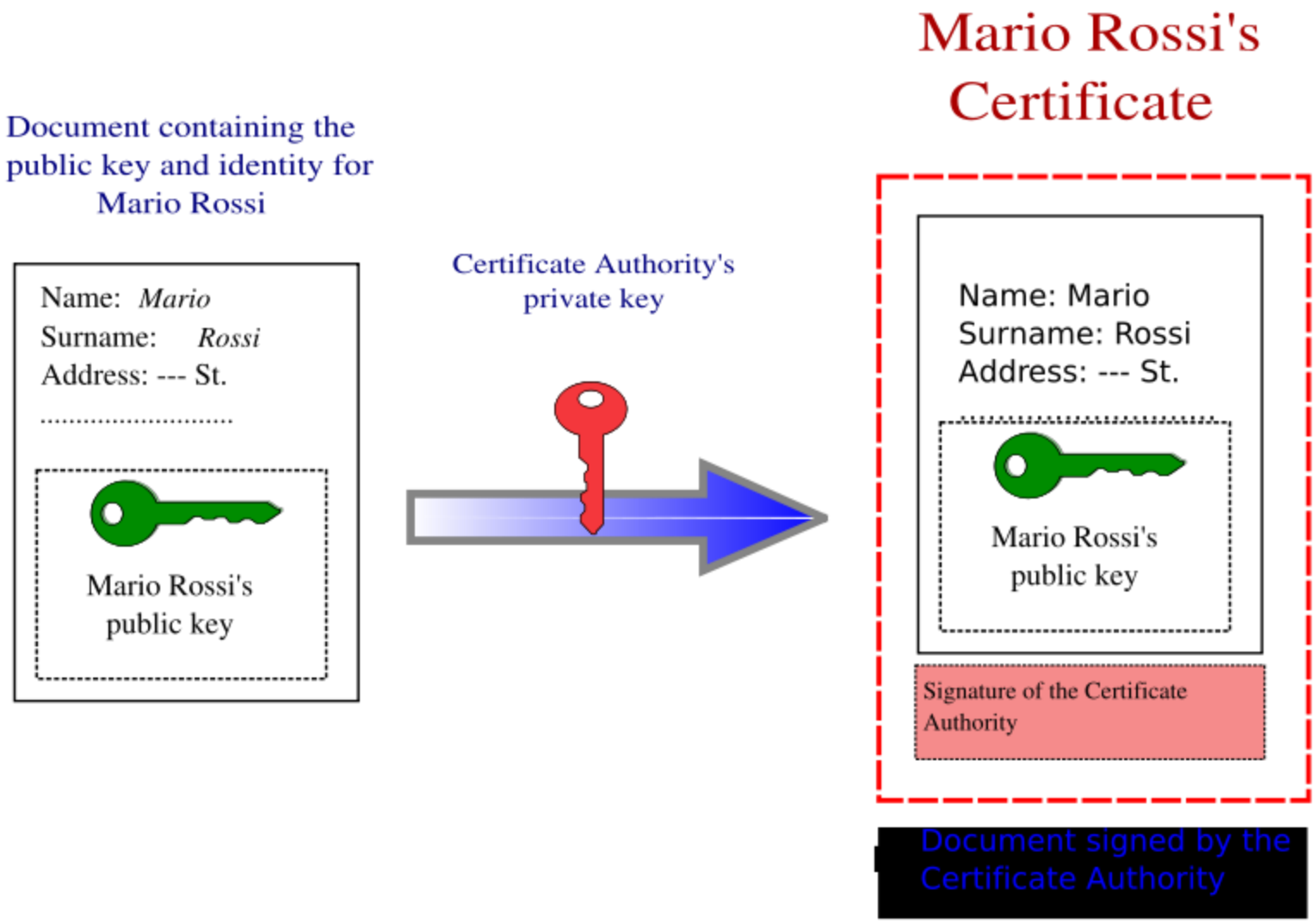
Public-Key Certificates

- A certificate binds identity (or other information) to public key
- It is a signature on a statement “Paul’s public key is 1032xD”

- Contents digitally signed by a trusted Public-Key or Certificate Authority (CA)
  - Can be verified by anyone who knows the authority’s public-key

- For Alice to send an encrypted message to Bob, obtains a certificate of Bob’s public key

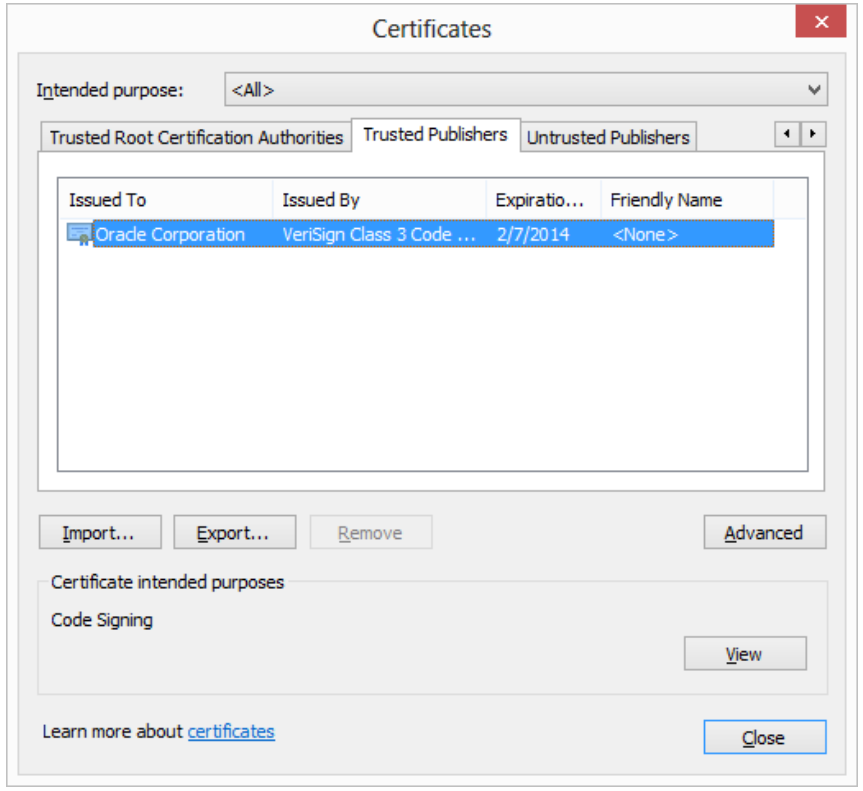
Details

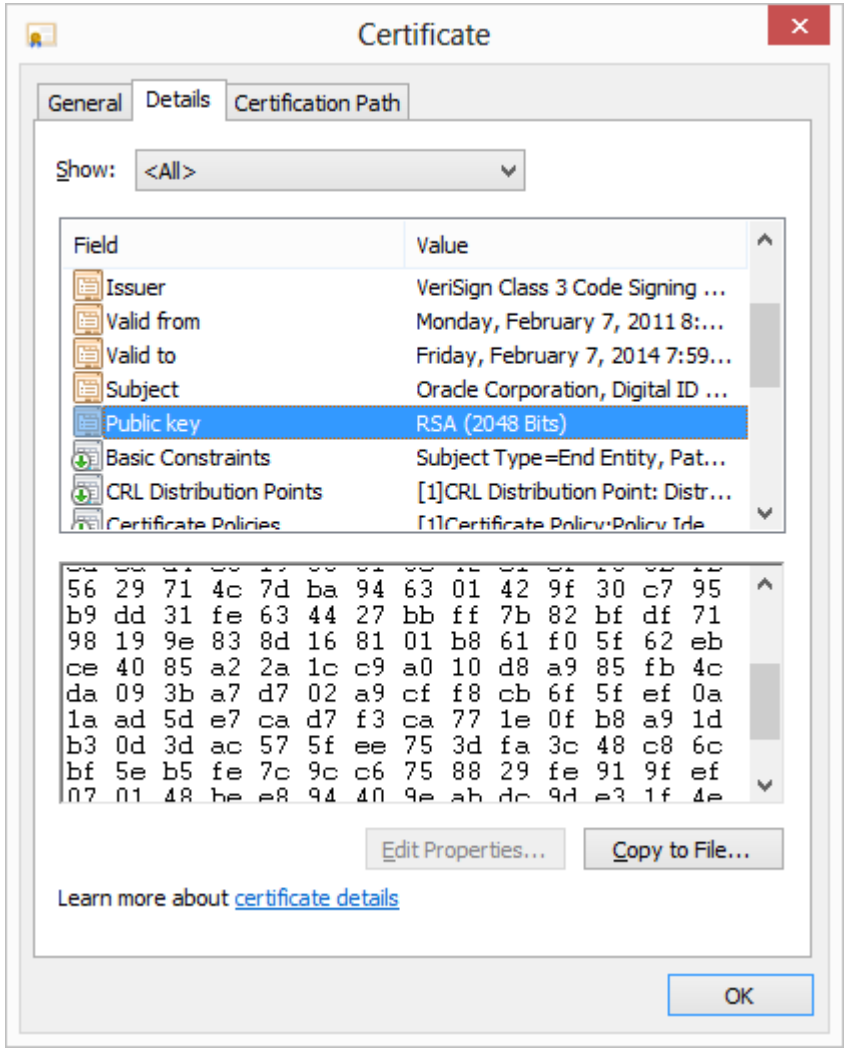

Document containing the public key and identity for Mario Rossi
Name: Mario
Surname: Rossi
Address: --- St.
Mario Rossi's public key
Certificate Authority's private key
Mario Rossi's Certificate
Name: Mario
Surname: Rossi
Address: --- St.
Mario Rossi's public key
Signature of the Certificate Authority
Document signed by the Certificate Authority

X.509 Certificates

- Defines framework for authentication services:
  - Defines that public keys stored as **certificates** in a public directory.
  - Certificates are **issued and signed** by an entity called **certification authority (CA)**
- Used by numerous applications: SSL
- Example: see certificates accepted by your browser

Example: Oracle's certificate

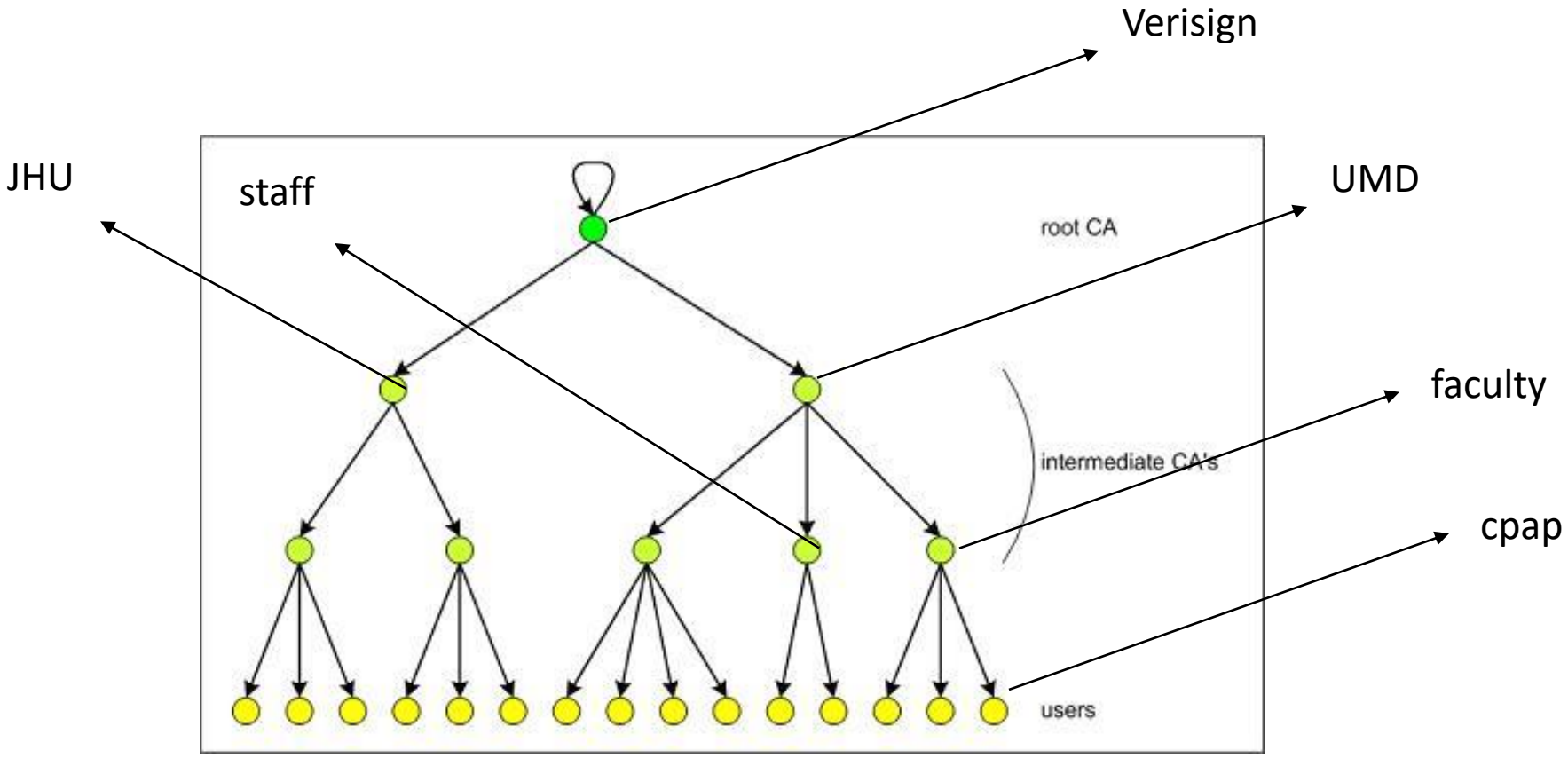




Method 1: Certificate Hierarchy

- Single CA certifying every public key is impractical
- Instead, use trusted root authorities
- Root CA signs certificates for intermediate CAs, they sign certificates for lower-level CAs, etc.
  - Certificate "chain of trust"
    - $sig_{Verisign}$("UMD", PK_{UMD})
    - sig_{UMD}("faculty", $PK_{faculty}$)
    - $sig_{faculty}$("cpap", PK_{cpap})

Example



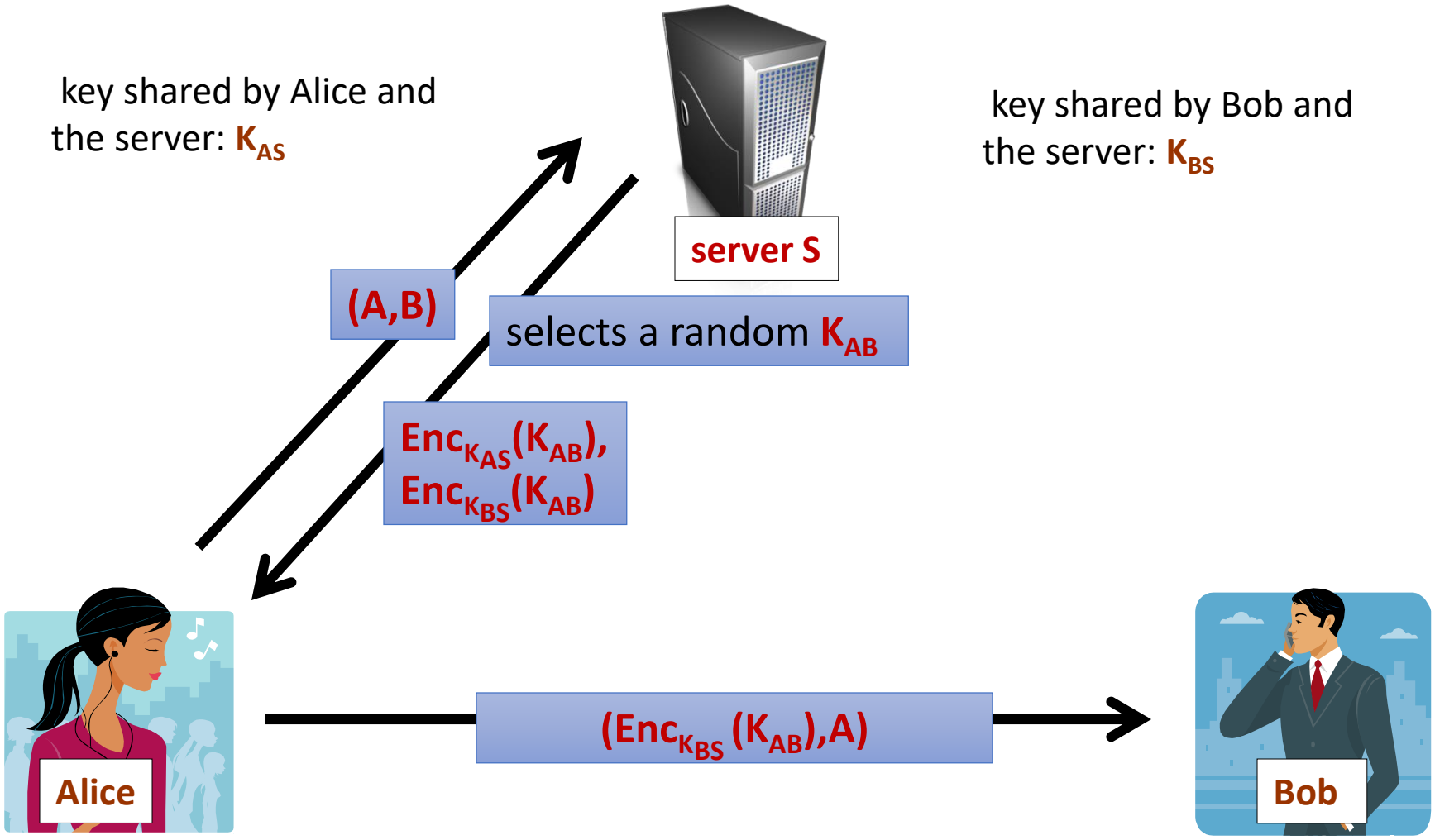
What bad things can happen if the root CA system is compromised?

Method 2: Web of Trust

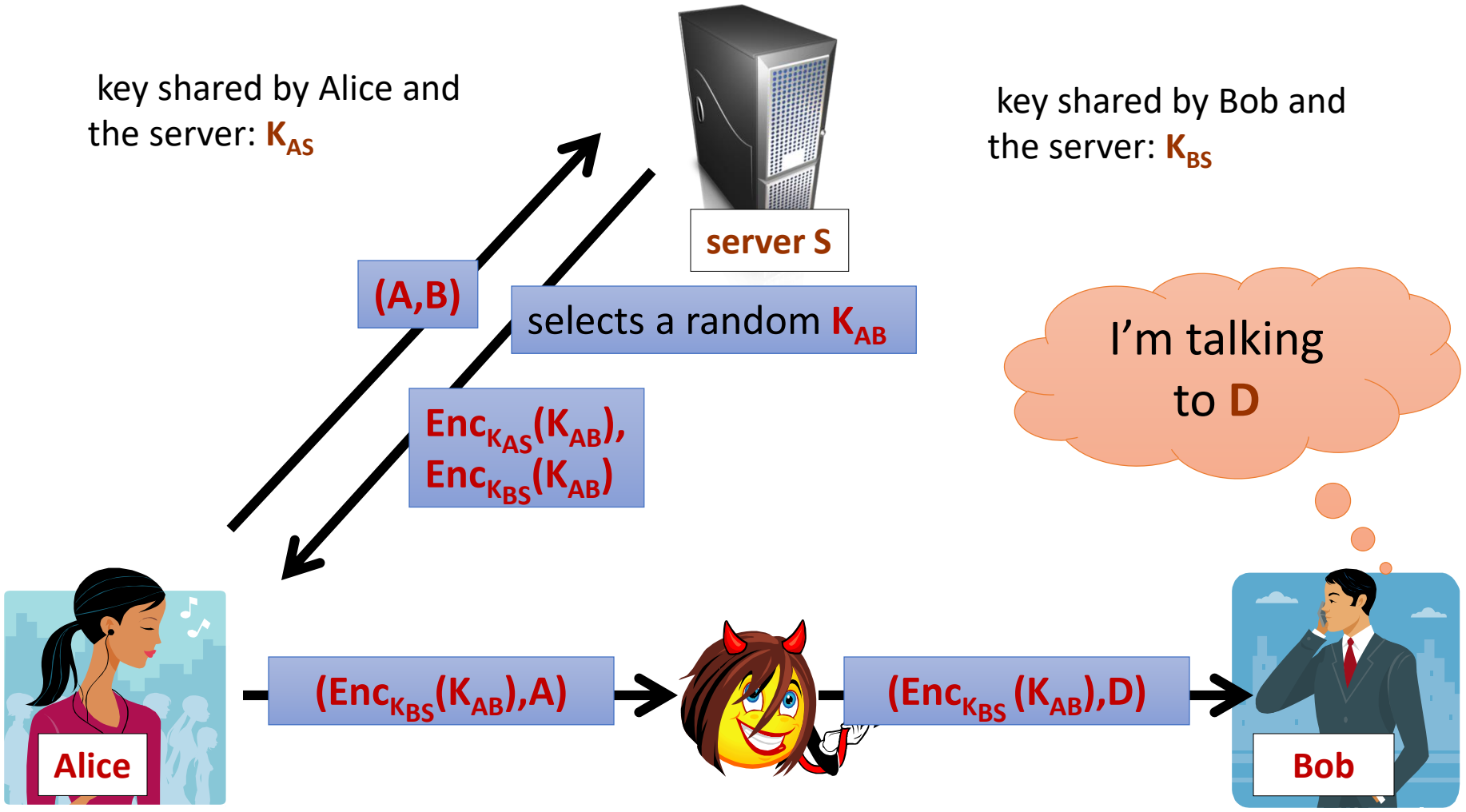
- No centralized authority
- Individuals sign one another's public keys, these "certificates" are stored along with keys in key rings.
- PGP computes a *trust level* for each public key in key ring.
- Users interpret trust level for themselves.
- Original intention was that all e-mail users would contribute to web of trust.
- Reality is that this web is sparsely populated.
- How should security-unaware users assign and interpret trust levels

What if we do not want to use PKI to share a key?

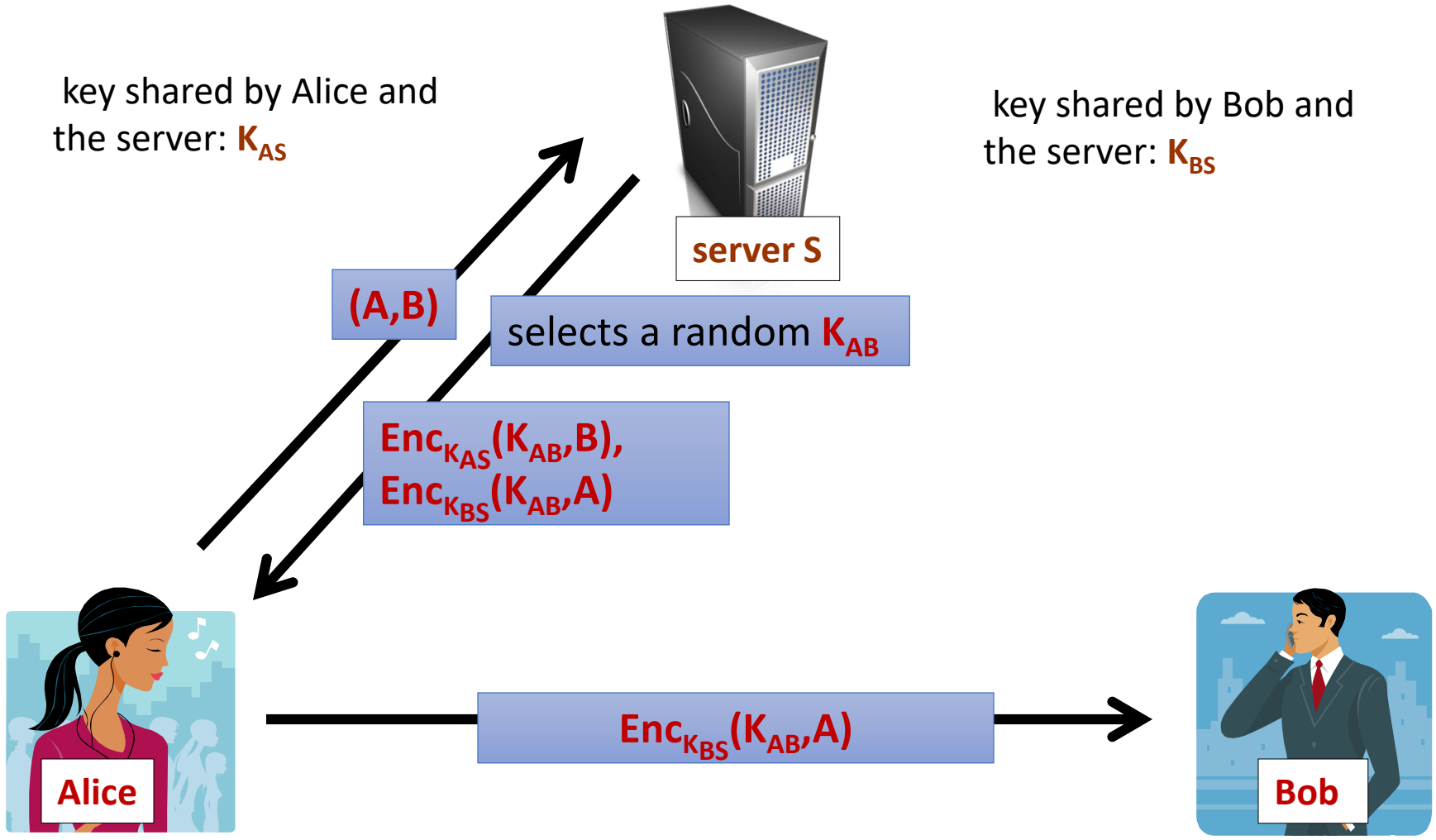
An idea (1)


key shared by Alice and the server: K_{AS}
key shared by Bob and the server: K_{BS}
server S
(A,B)
selects a random K_{AB}
$Enc_{K_{AS}}(K_{AB})$, $Enc_{K_{BS}}(K_{AB})$
Alice
$(Enc_{K_{BS}}(K_{AB}),A)$
Bob

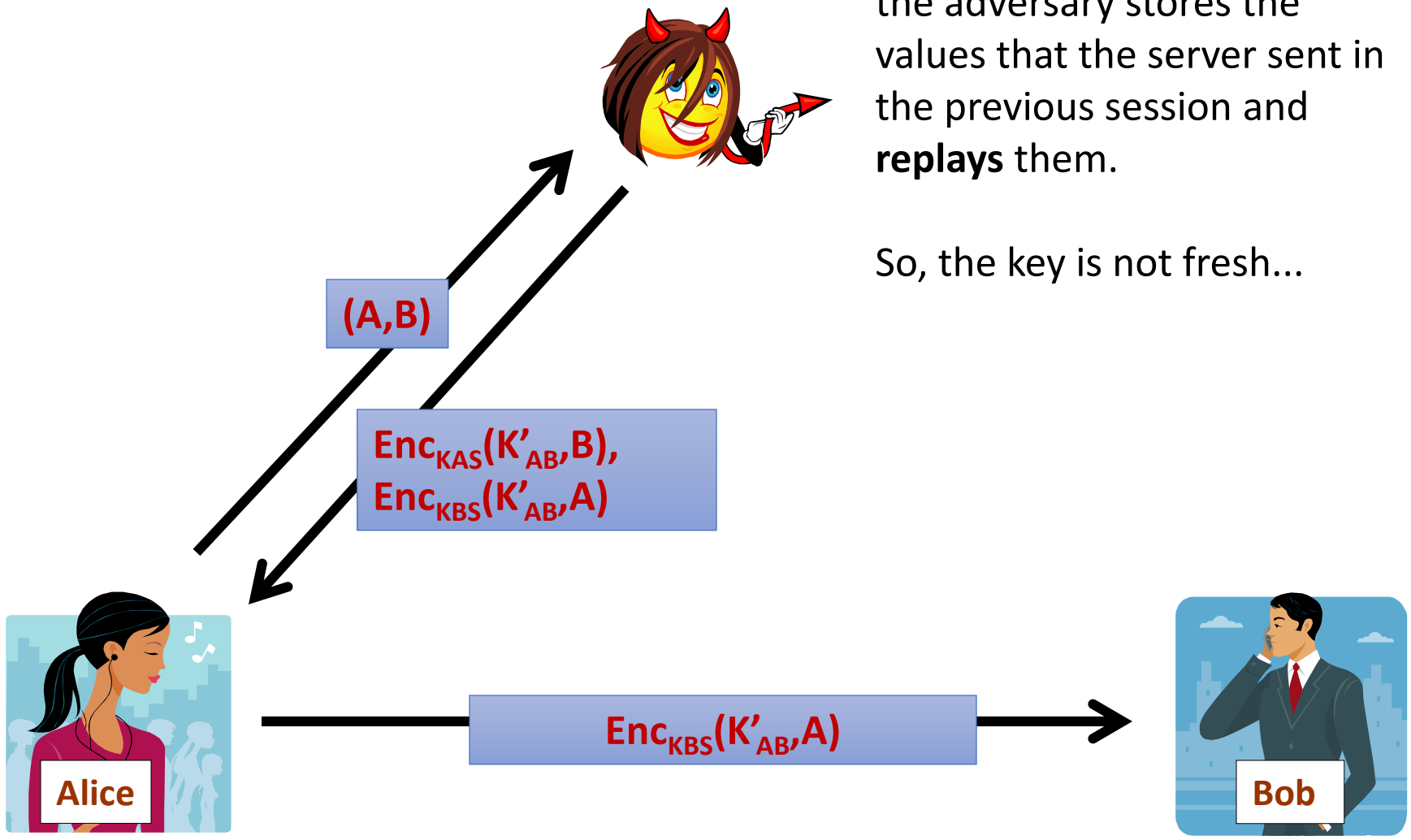
An attack


key shared by Alice and the server: K_{AS}
key shared by Bob and the server: K_{BS}
server S
(A,B)
selects a random K_{AB}
$Enc_{K_{AS}}(K_{AB})$, $Enc_{K_{BS}}(K_{AB})$
I'm talking to D
Alice
$(Enc_{K_{BS}}(K_{AB}),A)$
$(Enc_{K_{BS}}(K_{AB}),D)$
Bob

An idea (2)


key shared by Alice and the server: K_{AS}
key shared by Bob and the server: K_{BS}
server S
(A,B)
selects a random K_{AB}
$Enc_{K_{AS}}(K_{AB},B)$, $Enc_{K_{BS}}(K_{AB},A)$
Alice
$Enc_{K_{BS}}(K_{AB},A)$
Bob

A replay attack



the adversary stores the values that the server sent in the previous session and **replays** them.

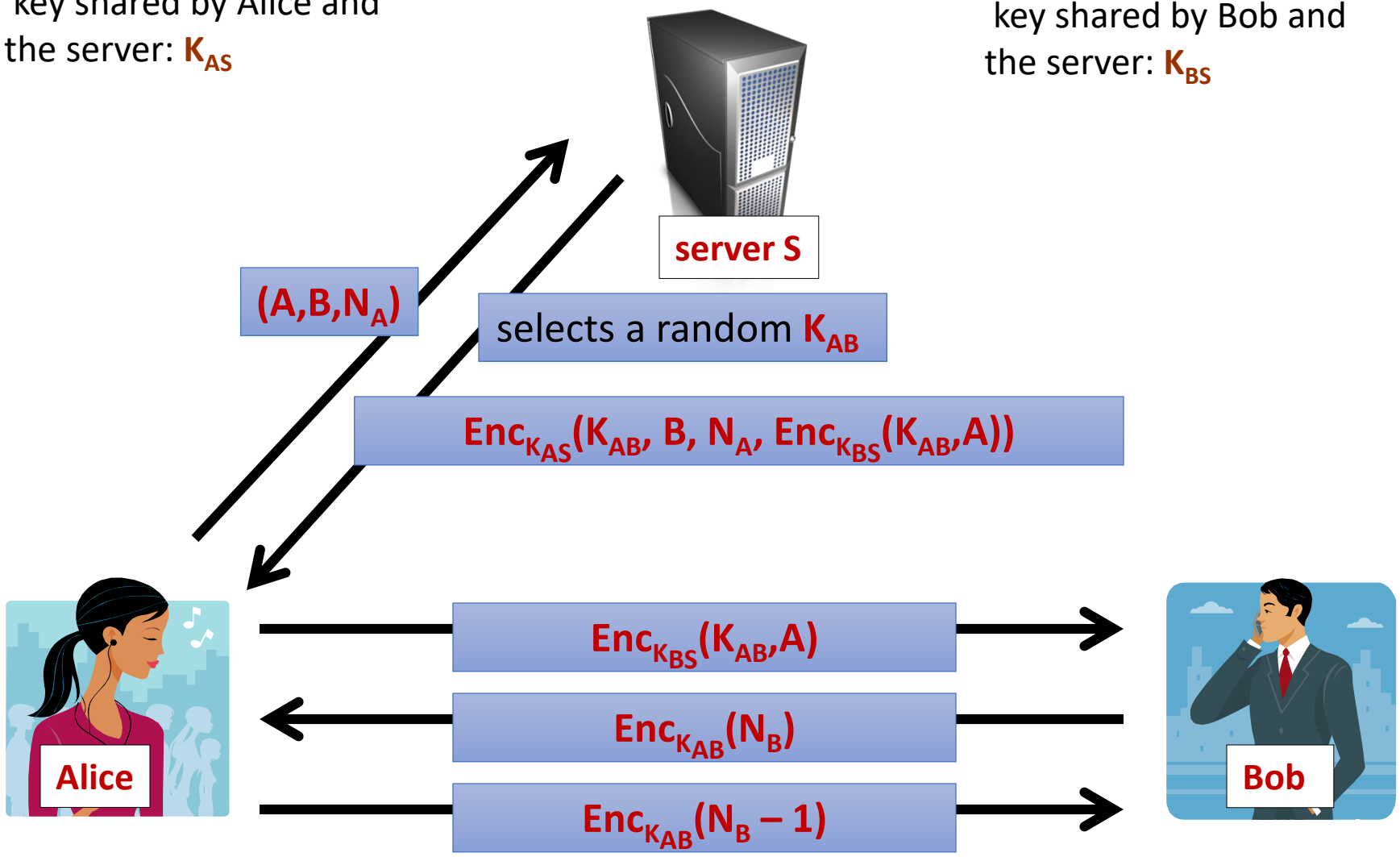
So, the key is not fresh...

How to protect against the replay attacks?

Nonce – “number used once”.

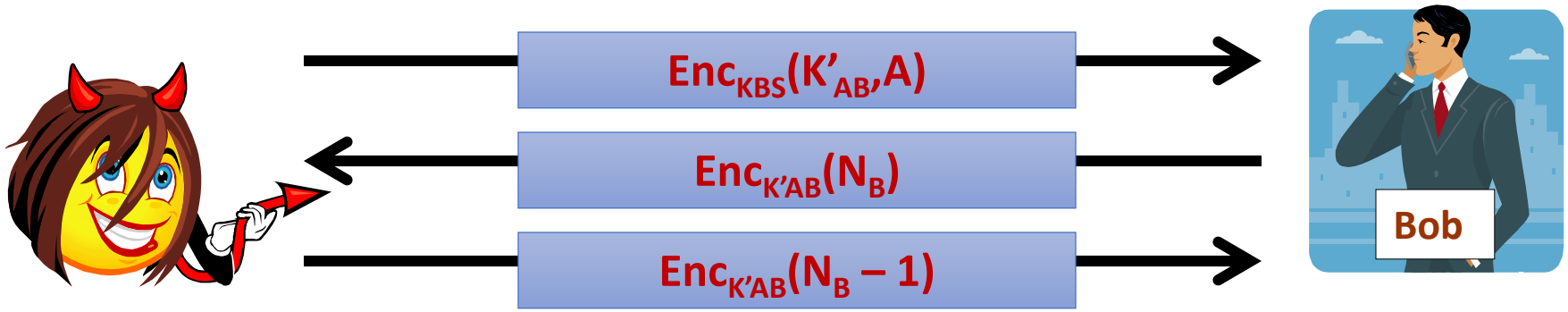
Nonce is a random number generated by one party and returned to that party to show that a message is newly generated.

An idea (3): Needham Schreoder 1972.



An attack on Needham Schroeder

Assume that an old session key K'_{AB} is known to the adversary.



The final solution

key shared by Alice and the server: K_{AS}

key shared by Bob and the server: K_{BS}

server S

(A,B,N_A,N_B)

selects a random K_{AB}

$Enc_{K_{AS}}(K_{AB}, B, N_A)$
$Enc_{K_{BS}}(K_{AB}, A, N_B)$

Alice

(B,N_B)

$Enc_{K_{BS}}(K_{AB}, A, N_B)$

Bob