CS 4173/5173
COMPUTER SECURITY

Authentication Protocols

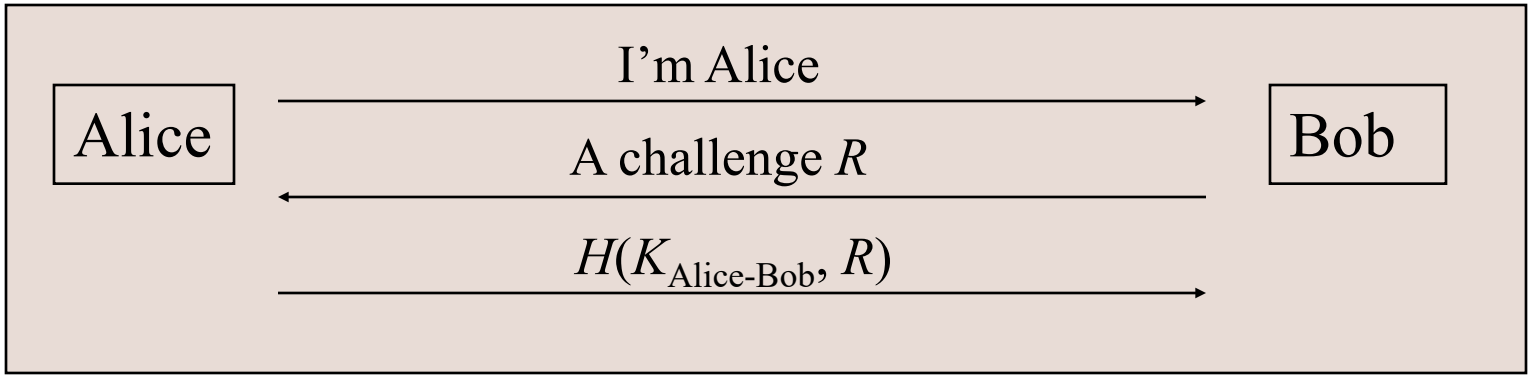
AUTHENTICATION HANDSHAKES

- Secure communication almost always includes an initial authentication handshake.
  - Authenticate each other
    - Based on cryptographic primitives
  - Establish session keys
  - *This process is not trivial; flaws in this process undermine secure communication*
    - Cryptographic primitives being secure is not equivalent to the design based on them being secure.

SECURITY ANALYSIS

- All communications are based on the public channel
- Ensure **MUTUAL** authentication
  - Alice proves to Bob that she is indeed Alice
    - If a third party Eve impersonates Alice, will she succeed?
  - and Bob proves to Alice that he is indeed Bob
    - If a third party Eve impersonates Bob, will she succeed?
- Ensure there is no other type of attacks
- Alice and Bob must share a common secret, but they need to prove to each other that they hold the same secret without disclosing it in the public channel

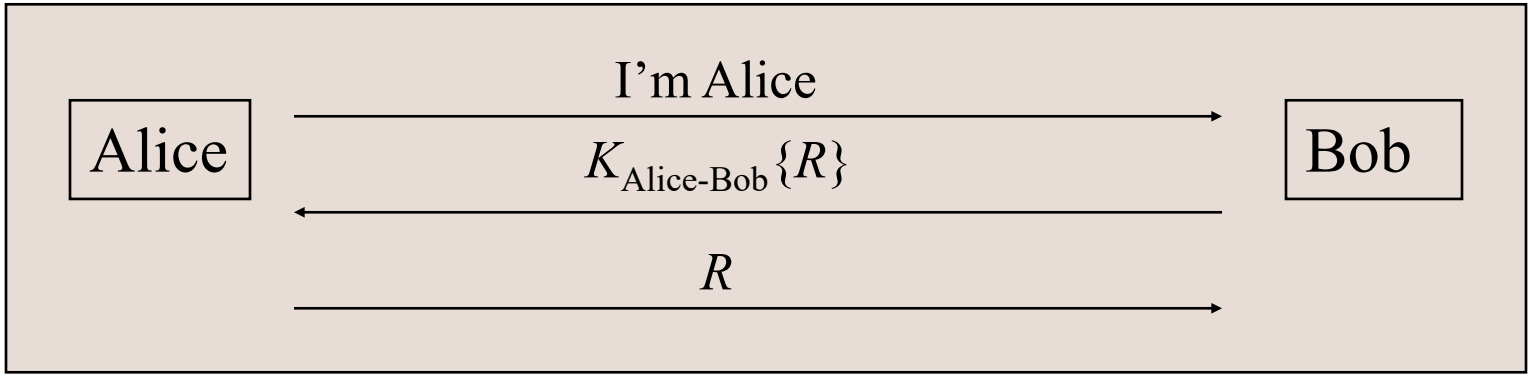
AUTHENTICATION WITH SHARED SECRET



- Assumptions:
  - Shared key: $K_{\text{Alice-Bob}}$
  - R is a random number

- Questions:
  - Can Alice prove to Bob that she is indeed Alice?
  - Can Bob prove that he is indeed Bob?

AUTHENTICATION WITH SHARED SECRET (CONT'D)



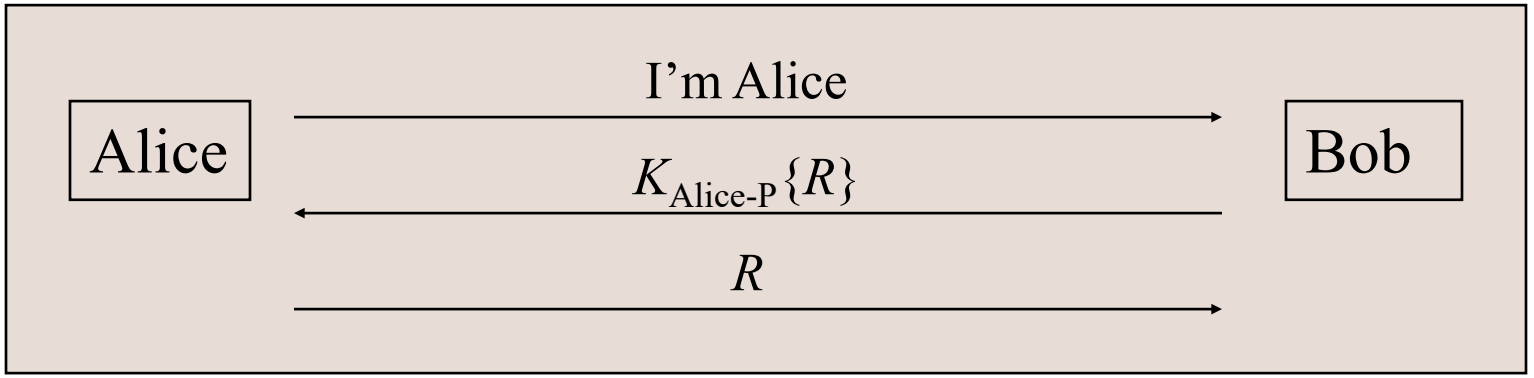
- Questions:
  - Can Alice prove to Bob that she is indeed Alice?
  - Can Bob prove that he is indeed Bob?

AUTHENTICATION WITH PUBLIC KEY

- Alice and Bob know each other's public key

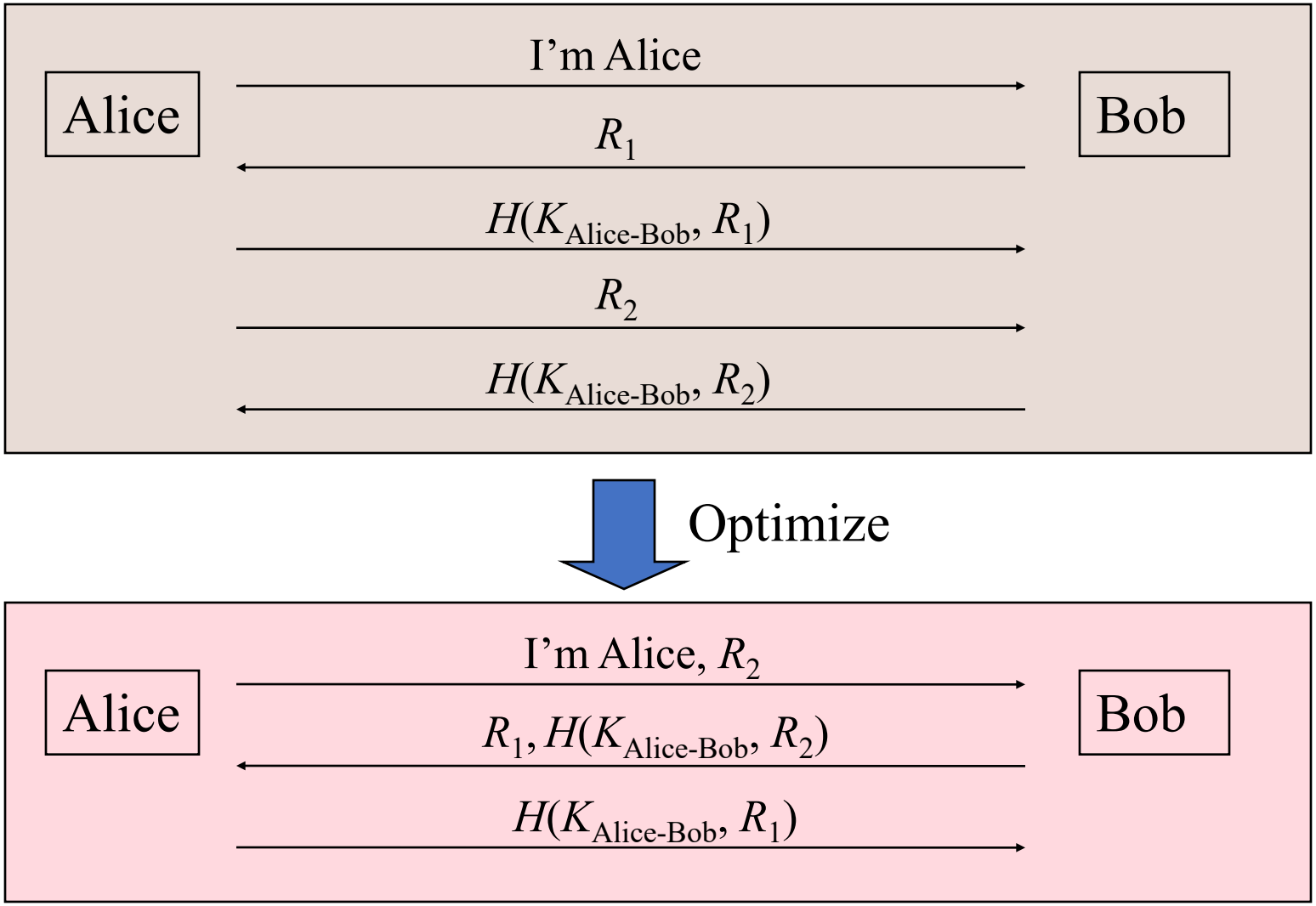
- Questions:
  - Can Alice prove to Bob that she is indeed Alice?
  - Can Bob prove that he is indeed Bob?

AUTHENTICATION WITH PUBLIC KEY (CONT'D)



- Questions:
  - Can Alice prove to Bob that she is indeed Alice?
  - Can Bob prove that he is indeed Bob?

MUTUAL AUTHENTICATION



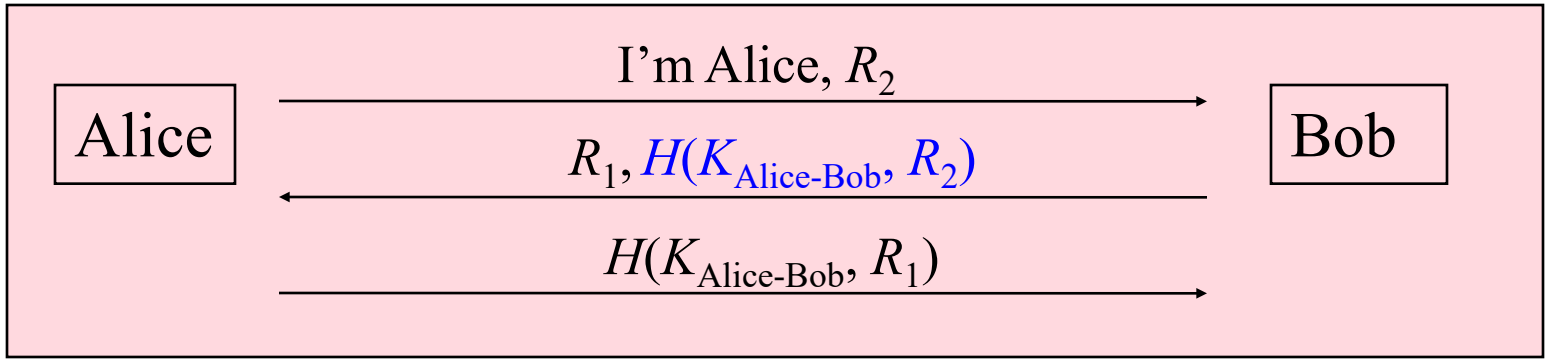
MUTUAL AUTHENTICATION (CONT'D)

- Reflection attack

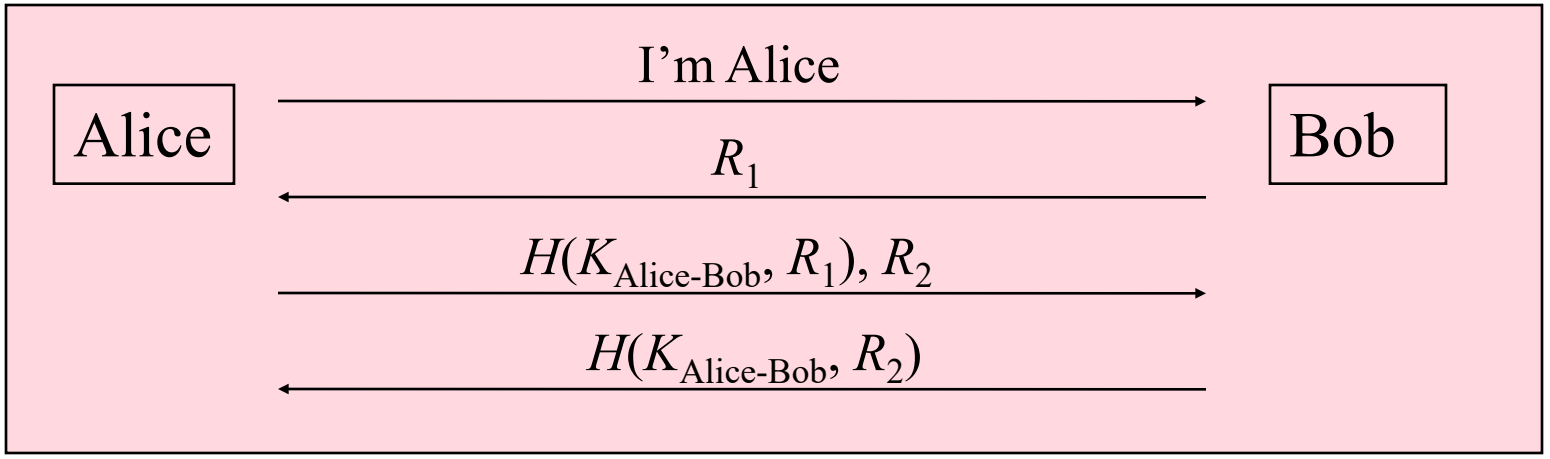
REFLECTION ATTACKS (CON'TD)

- Lesson: Don't have Alice and Bob do exactly the same thing
  - Different keys
    - Totally different keys
    - $K_{Alice-Bob} = K_{Bob-Alice} + 1$
  - Different Challenges: Alice and Bob's challenges cannot be the same
  - The initiator should be the first to prove its identity
    - Assumption: initiator is more likely to be the bad guy

MUTUAL AUTHENTICATION (CONT'D)

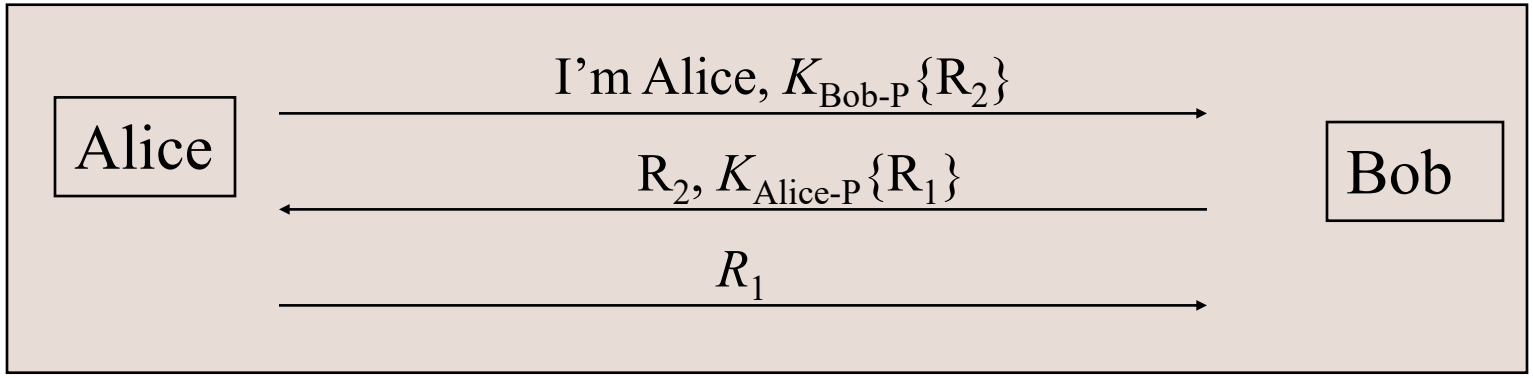


Countermeasure: Alice proves herself first



MUTUAL AUTHENTICATION (CONT'D)

- Public keys
  - Authentication of public keys is a critical issue



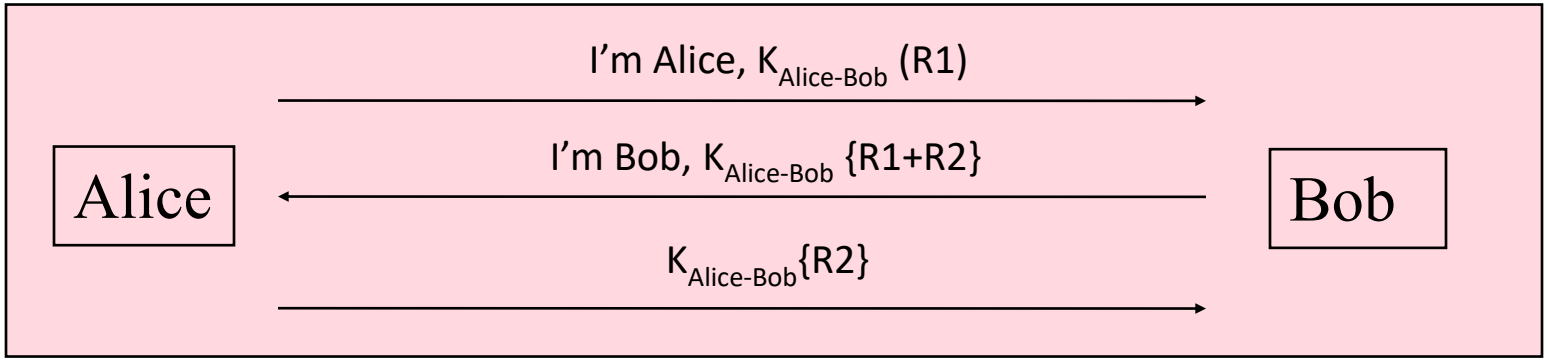
- Question:
  - Vulnerability to reflection?
  - Any bad design?

BETTER DESIGN

- Provide mutual authentication
- Make two parties do different things
- Challenge the initiators first
- Avoid reflection attacks
- Avoid message decryption

EXERCISE

- In a three-message authentication protocol, Alice initiates contact with Bob. Assume that Alice and Bob share a key $K_{Alice-Bob}$. The protocol works as follows, where R1 and R2 are random numbers generated by Alice and Bob, respectively. Is it mutual authentication?

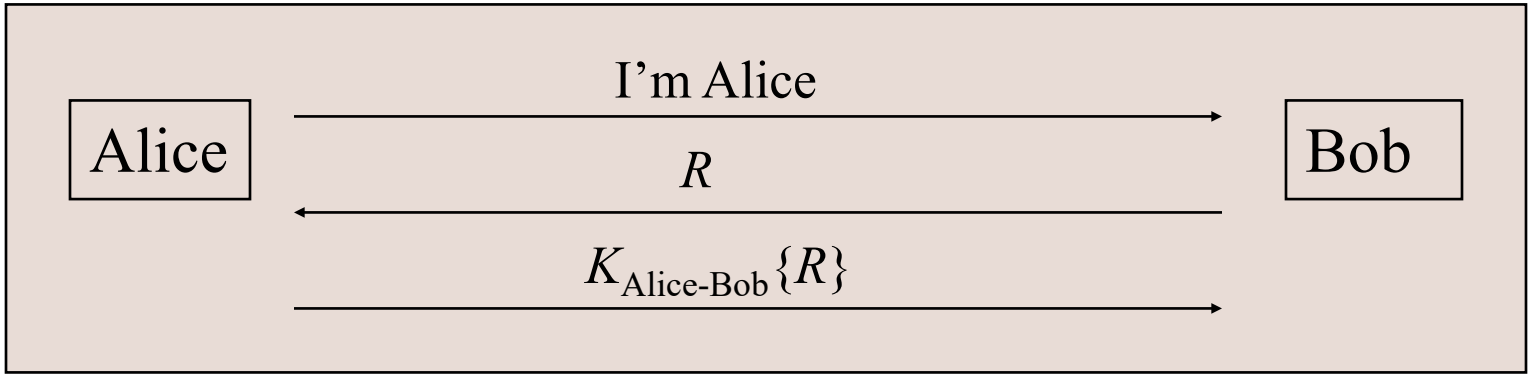


INTEGRITY/ENCRYPTION FOR DATA

- Communication after mutual authentication should be cryptographically protected as well
  - Require a session key established during mutual authentication

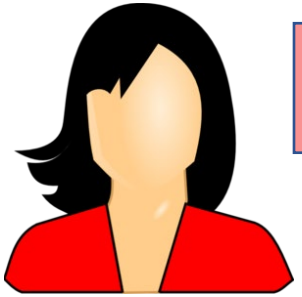
ESTABLISHMENT OF SESSION KEYS

- Secret key based authentication
  - Assume the following authentication happened.
  - Can we use $K_{Alice\text{-}Bob}\{R\}$ as the session key?
  - Can we use $K_{Alice\text{-}Bob}\{R+1\}$ as the session key?
  - Can we use $K_{Alice\text{-}Bob}+1\{R\}$ as the session key?
  - In general, modify $K_{Alice\text{-}Bob}$ and encrypt R. Use the result as the session key.



ESTABLISHMENT OF SESSION KEYS

- Public key based authentication
  - RSA based key negotiation



Alice

Public key: Ka,p
Private key: Ka,i

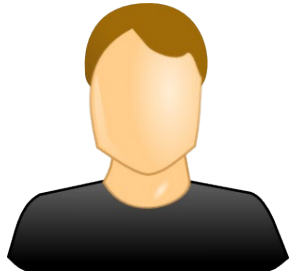
Generate random number R1

Send R1 encrypted using Kb,p

1) Get R1,
2) Generate random number R2,
3) get key as K = H(*R*1⊕*R*2),

Send R2 encrypted using Ka,p

1) get R2
2) get symmetric key as K = H(*R*1⊕*R*2),

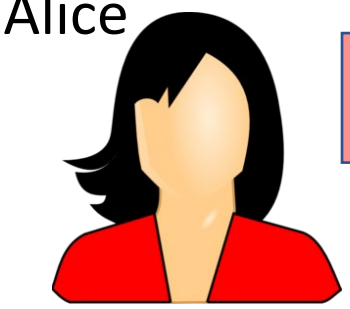


Bob

Public key: Kb,p
Private key: Kb,i

ESTABLISHMENT OF SESSION KEYS

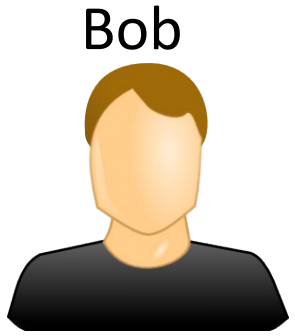
- Public key based authentication
  - Diffie-Hellman negotiation
    - Alice and Bob signs the quantity they send



Public key: Ka,p
Private key: Ka,i

Generate random number S_A

Generate random number S_B



Public key: Kb,p
Private key: Kb,i

Compute, **sign** and send $T_A = g^{S_A} \bmod p$

Compute $T_A^{S_B} \bmod p = \underline{\mathbf{g^{S_A S_B} \bmod p}}$

Compute, **sign** and send $T_B = g^{S_B} \bmod p$

Compute $T_B^{S_A} \bmod p = \underline{\mathbf{g^{S_A S_B} \bmod p}}$

TWO-WAY PUBLIC KEY BASED AUTHENTICATION

- Approach I
  - Alice chooses and encrypts R_1 with Bob's public key
  - Bob chooses and encrypts R_2 with Alice's public key
  - Session key is $H(R_1 \oplus R_2)$
  - Trudy will have to compromise both Alice and Bob

- Approach II
  - Alice and Bob establish the session key with Diffie-Hellman key exchange
  - Alice and Bob signs the quantity they send

SUMMARY

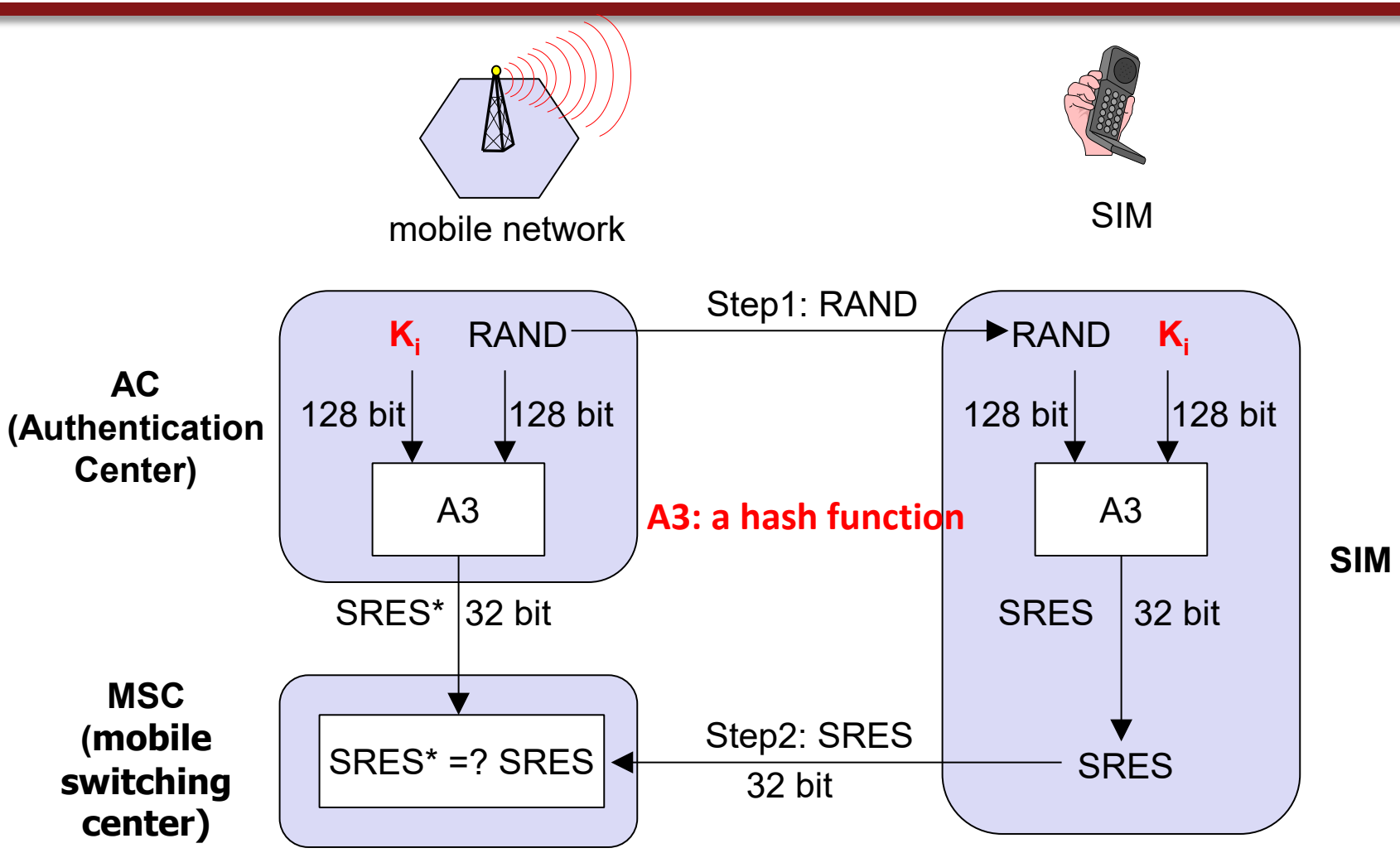
- Design a perfect authentication protocol requires non-trivial efforts
  - Can be based on symmetric or public key systems
- Some guidelines to check a protocol:
  - The initiators should authenticate themselves first
  - Need asymmetric challenge-response, be aware of reflection attacks
    - Make two parties do different things
  - Provide mutual authentication
  - Avoid message decryption
- Design based on public key:
  - RSA key negotiation
  - Diffie-Hellman with authentication

REAL-WORLD EXAMPLE

Fake GSM base station! Why?

Reading materials, not required in homeworks/exams

GSM - AUTHENTICATION



It is mutual authentication? Why?

Reading materials, not required in homeworks/exams