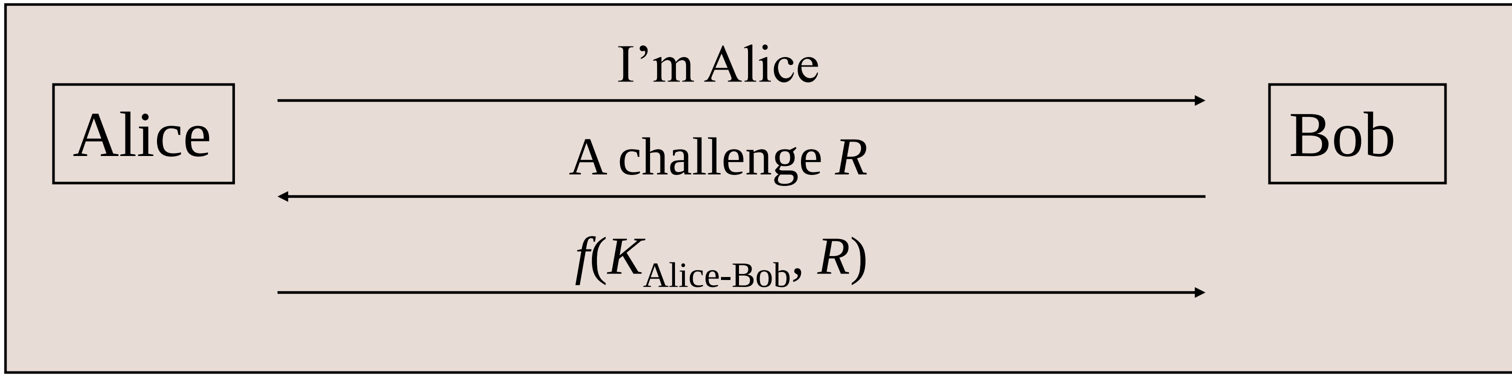
CIS 6930/4930 Computer and Network Security

Topic 6.2 Authentication Protocols

Authentication Handshakes

- Secure communication almost always includes an initial authentication handshake.
  - Authenticate each other
  - Establish session keys
  - *This process is not trivial; flaws in this process undermine secure communication*

Authentication with Shared Secret

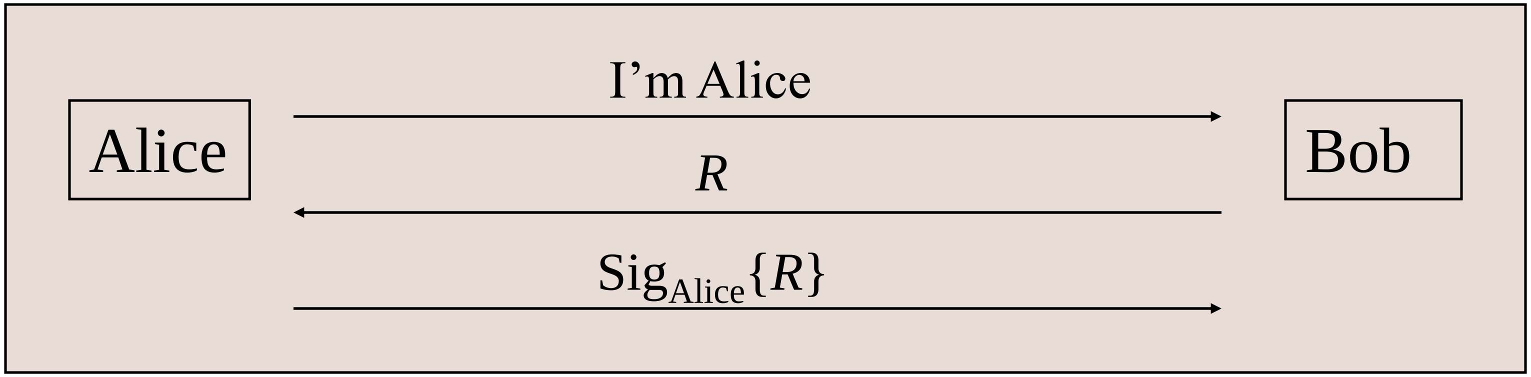


- Weaknesses
  - Authentication is not mutual; Trudy can convince Alice that she is Bob
  - Trudy can hijack the conversation after the initial exchange
  - If the shared key is derived from a password, Trudy can mount an off-line password guessing attack

Authentication with Shared Secret (Cont'd)

- Weaknesses
  - Alice still cannot authenticate Bob
  - Trudy can easily get pairs of (plaintext , ciphertext)
  - Trudy can hijack the conversation after the initial exchange
  - Other vulnerability?

Authentication with Public Key



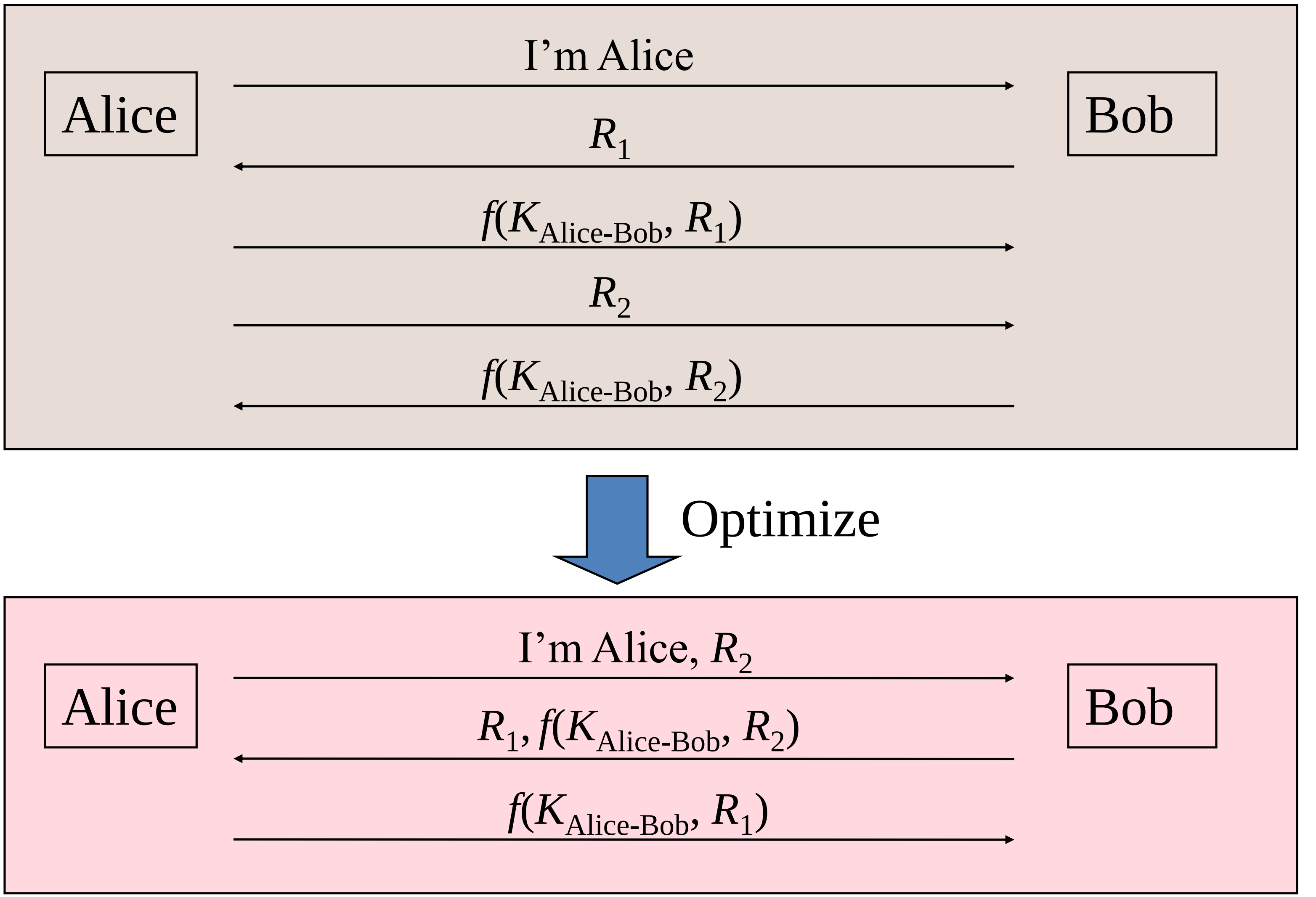
- Weaknesses
  - Authentication is not mutual; Trudy can convince Alice that she is Bob
  - Trudy can hijack the conversation after the initial exchange
  - Trudy can trick Alice into signing something

Authentication with Public Key (Cont'd)

A variation

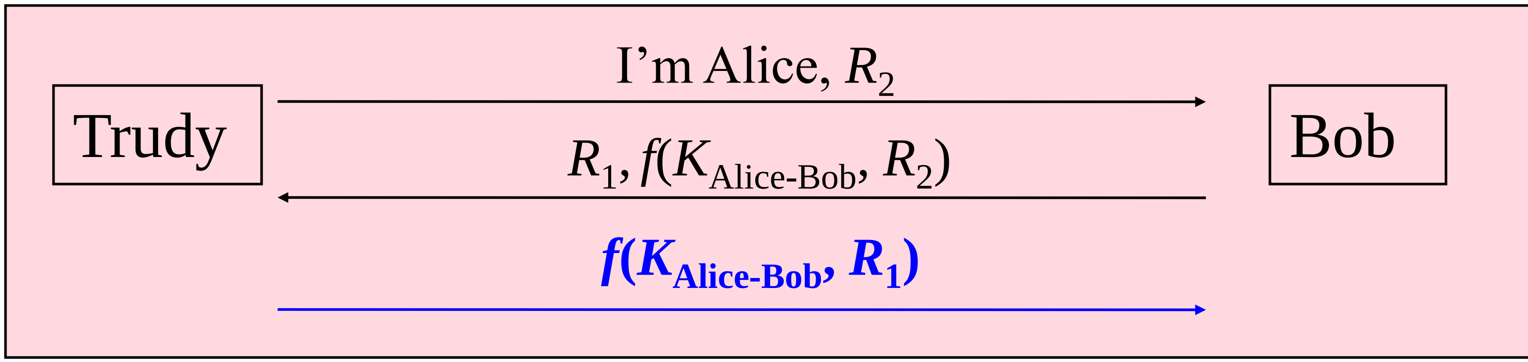
What Bob can trick Alice to do?

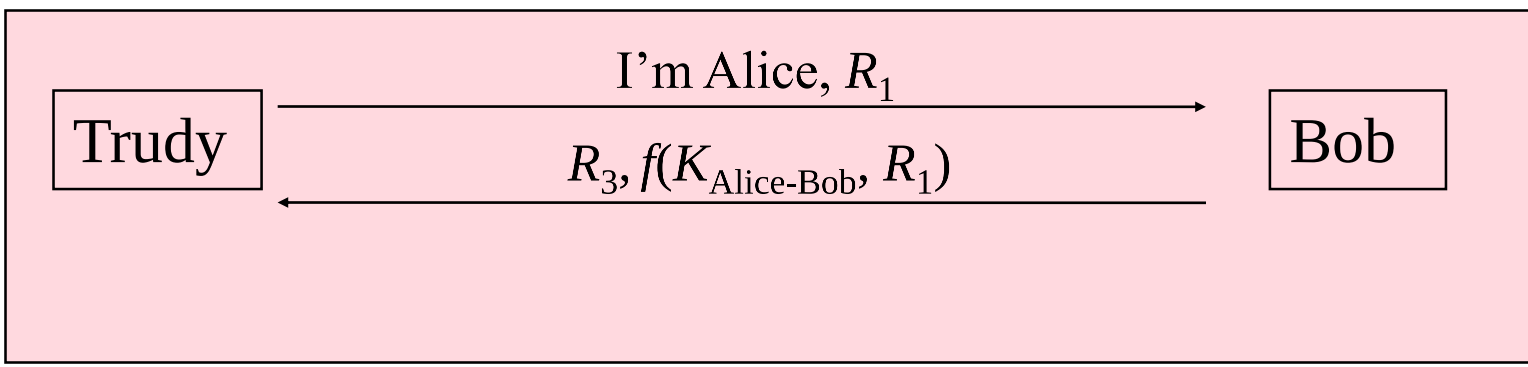
Mutual Authentication



Mutual Authentication (Cont’d)

- Reflection attack

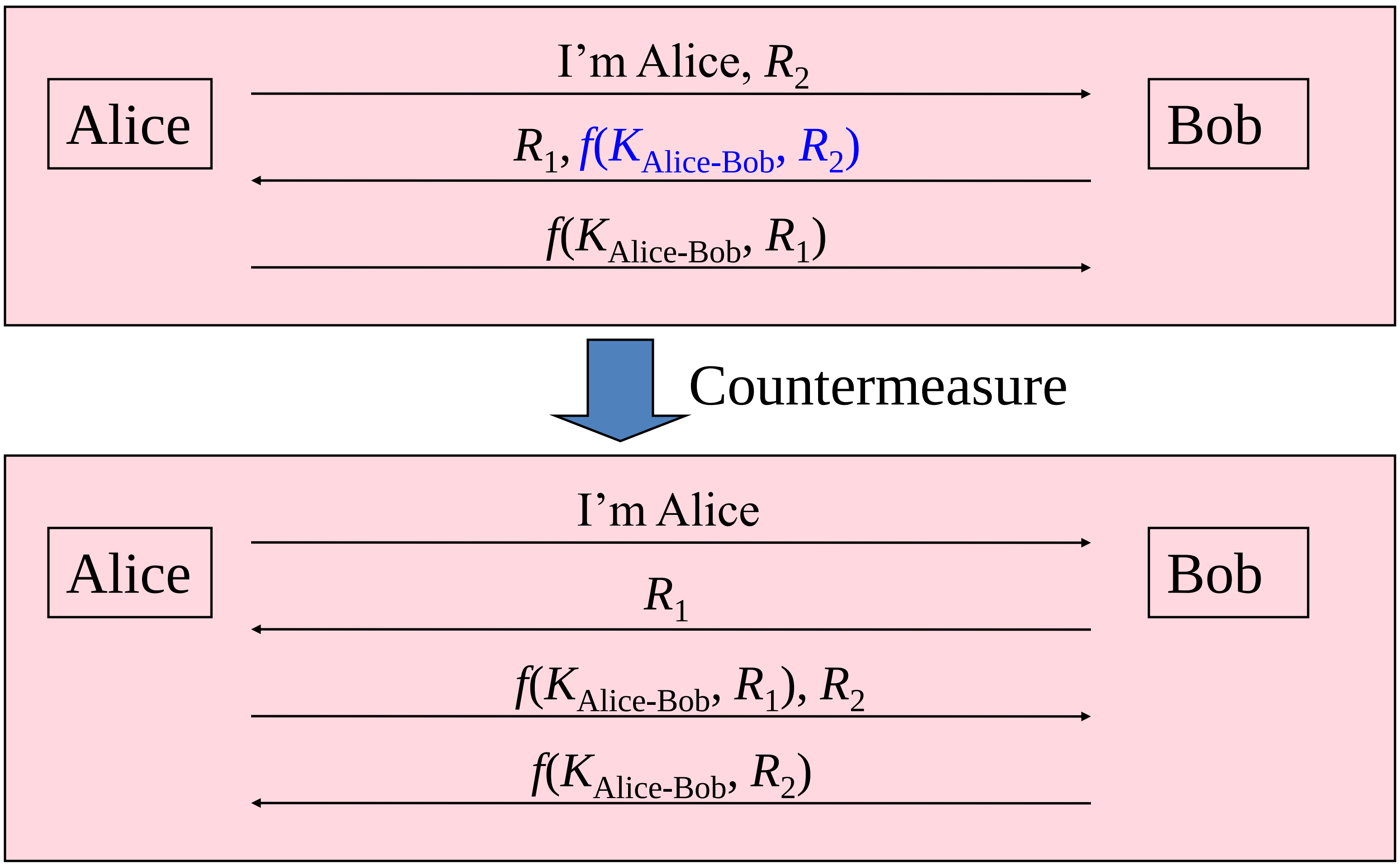




Reflection Attacks (Cont'd)

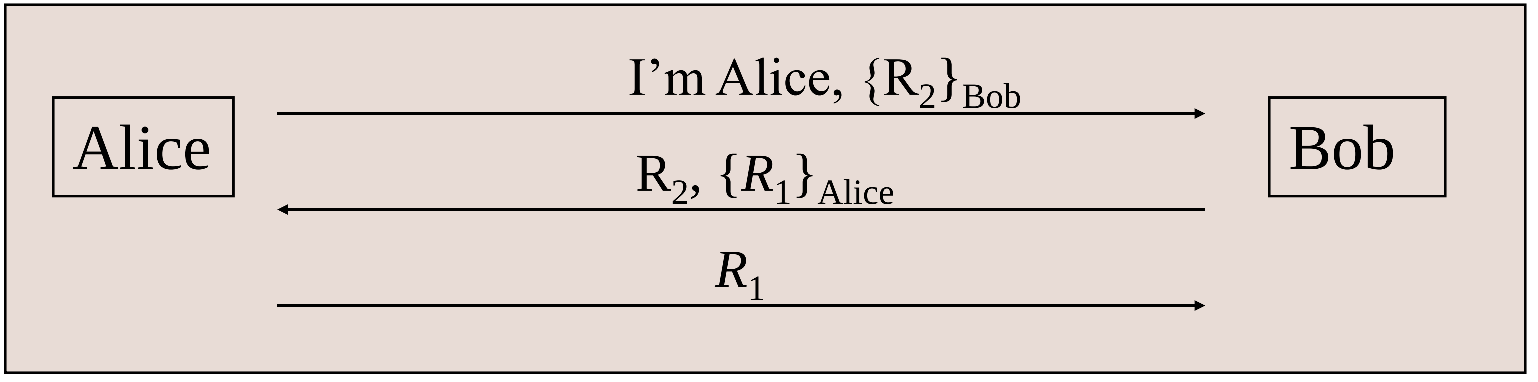
- Lesson: Don't have Alice and Bob do exactly the same thing
  - Different keys
    - Totally different keys
    - $K_{Alice\text{-}Bob} = K_{Bob\text{-}Alice} + 1$
  - Different Challenges: Alice and Bob's challenges cannot be the same
  - The initiator should be the first to prove its identity
   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

Mutual Authentication (Cont’d)



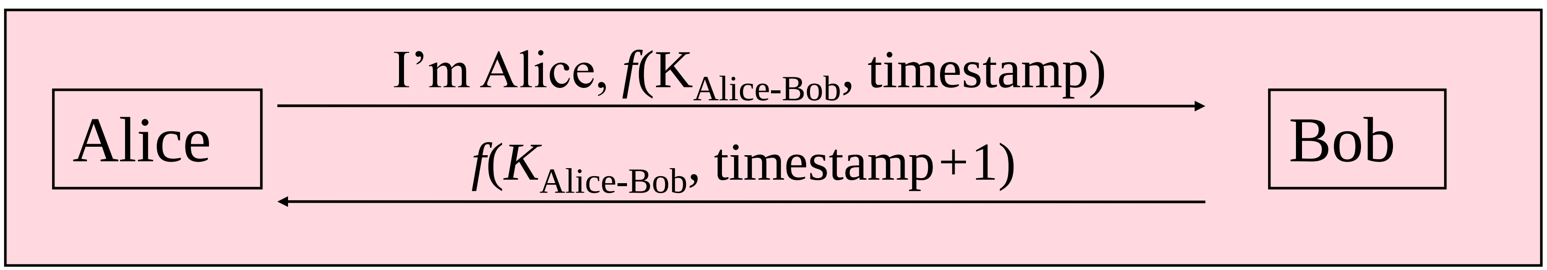
Mutual Authentication (Cont'd)

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



Mutual Authentication (Cont’d)

- Mutual authentication with timestamps
  - Require synchronized clocks
  - Alice and Bob have to encrypt different timestamps

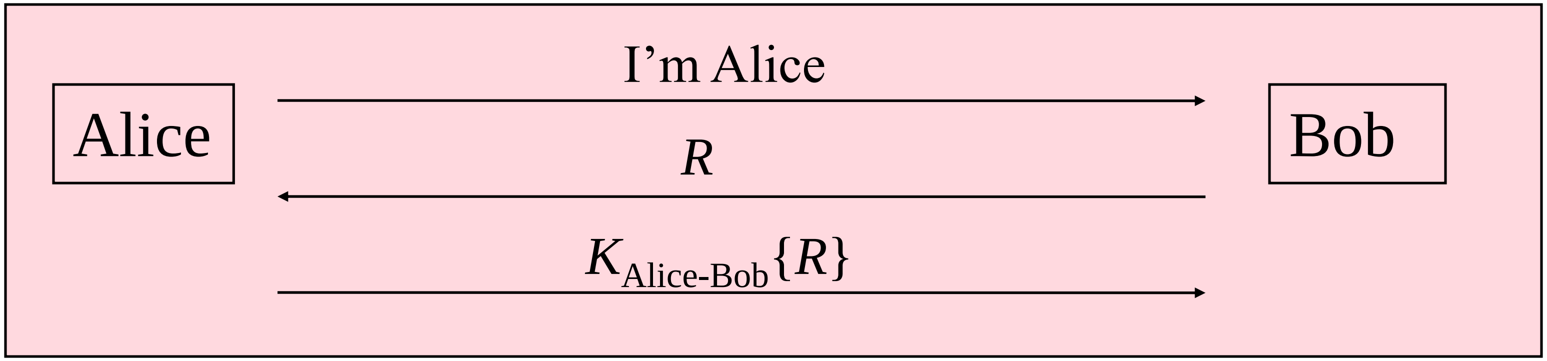


Integrity/Encryption for Data

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

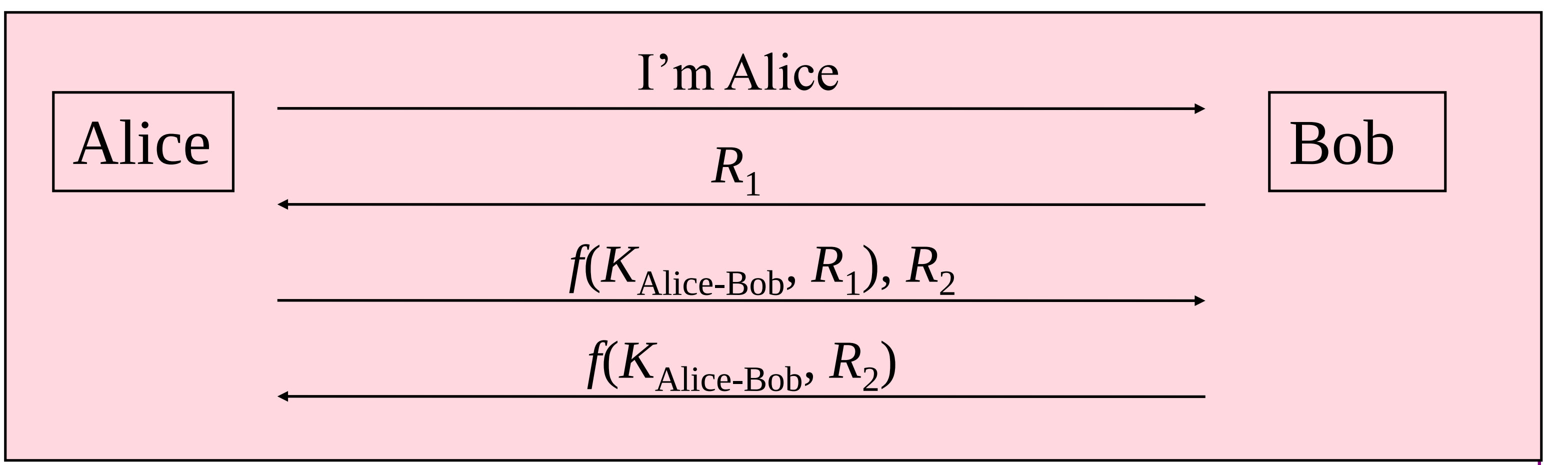
Establishment of Session Keys

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



Establishment of Session Keys

- Secret key based authentication
  - Can we use $f(K_{\text{Alice-Bob}}, R_1)$, or $f(K_{\text{Alice-Bob}}, R_2)$ as the session key?
  - Can we use $f(K_{\text{Alice-Bob}}, R_1+1)$, or $f(K_{\text{Alice-Bob}}, R_2+1)$ as the session key?
  - Can we use $f(K_{\text{Alice-Bob}}+1, R_1)$, or $f(K_{\text{Alice-Bob}}+1, R_2)$ as the session key?



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 $R_1 \oplus R_2$
  - 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

Establishment of Session Keys (Cont'd)

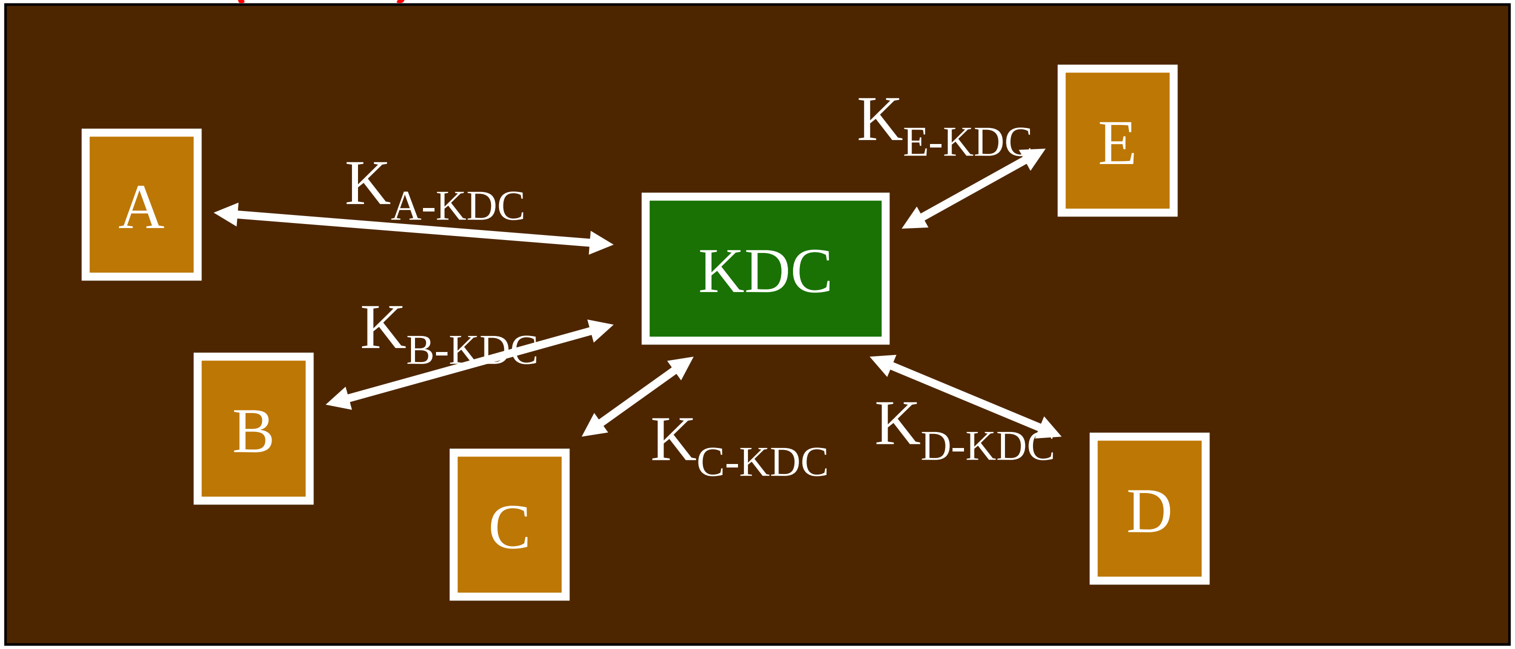
- One-way public key based authentication
  - It's only necessary to authenticate the server
  - Alice encrypts R with the server Bob's public key
  - Diffie-Hellman key exchange
    - Bob signs the D-H public key

Trusted Key Servers

- How do a large number of users authenticate each other?
  - inefficient / impractical for every pair of users to negotiate a secret key or share passwords
- Alternative: everybody shares a key with (and authenticates to) a single trusted third party
- Assumes there is a way to negotiate a key with the *third party*

Trusted… (cont'd)

- Shared keys between the *Key Distribution Center (KDC)* and users



(Simplified) Example of Use

- Alice wishes to communicate securely with Bob; Alice has previously negotiated $K_{A\text{-}KDC}$ with the KDC, Bob has negotiated $K_{B\text{-}KDC}$

1. Alice requests from the KDC a session key to use with Bob
2. KDC generates session key K_S, sends to Alice, encrypted with $K_{A\text{-}KDC}$
3. KDC also sends K_S to Bob, encrypted with $K_{B\text{-}KDC}$

- Alice and Bob can then communicate using K_S

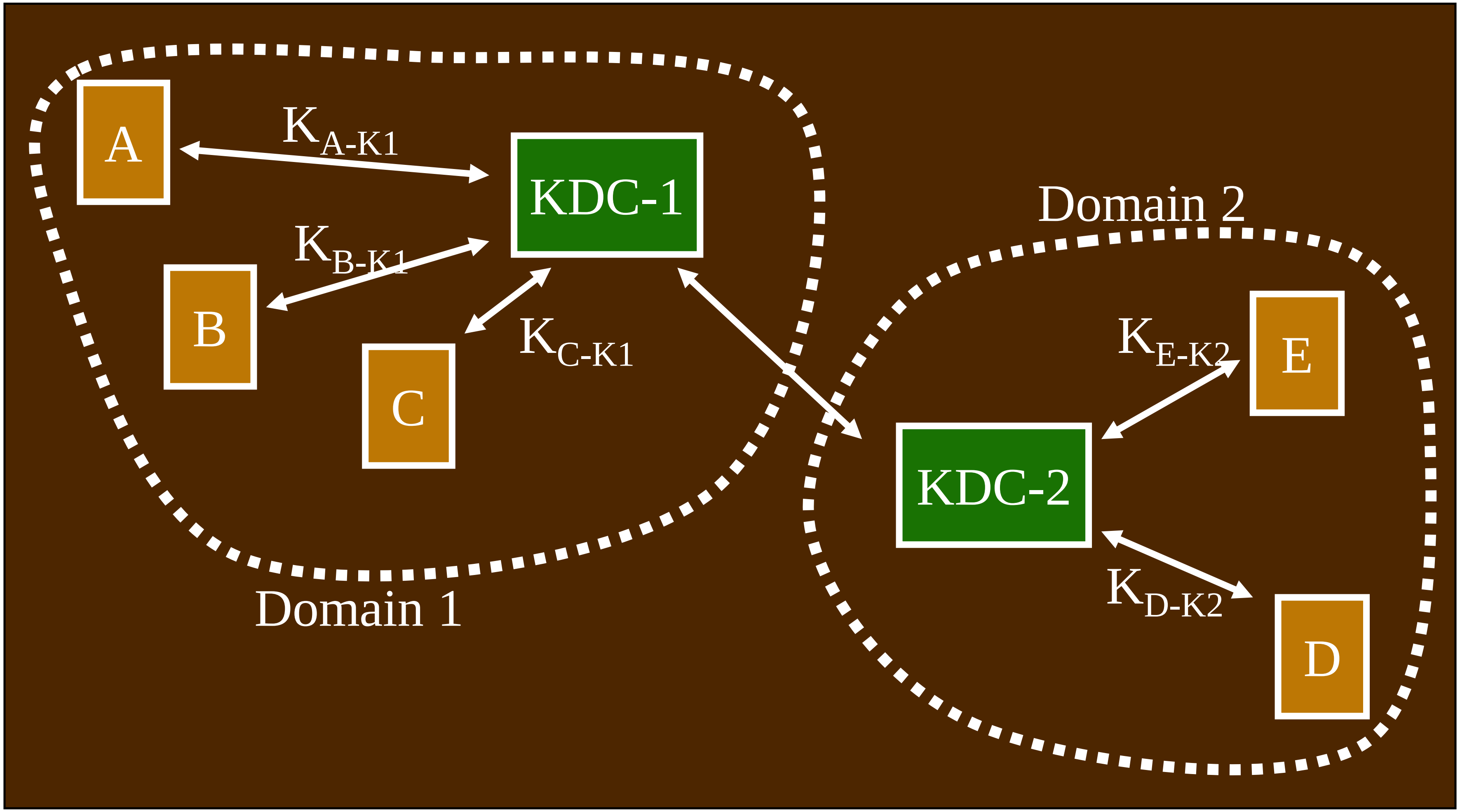
Assessment

- Simplifies mutual authentication / key negotiation, but…
  - secure against attacks?
  - robust to failures?
  - efficient?

A Hierarchy of KDCs

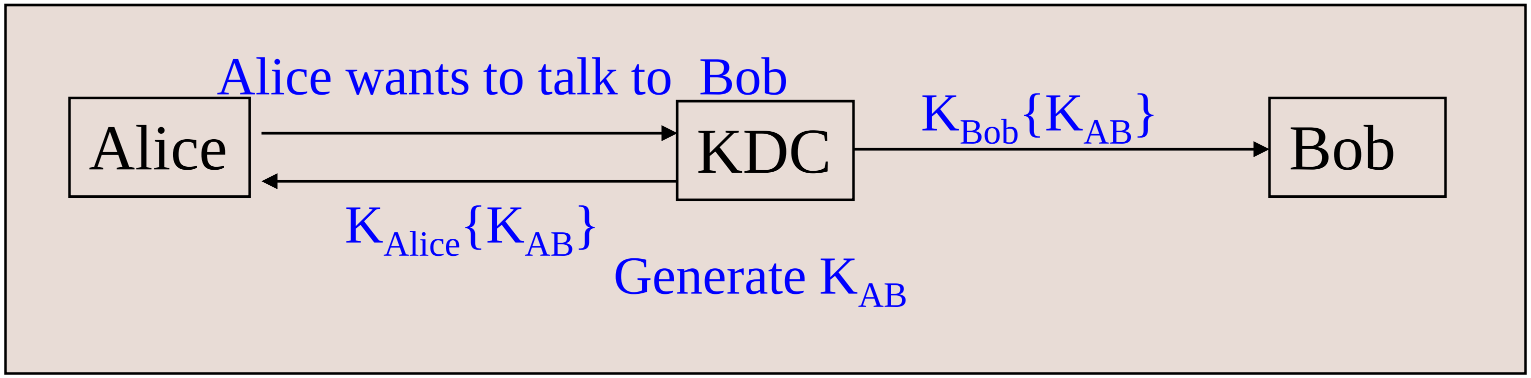
- For an Internet, not practical to have a single KDC
  - instead, imagine one KDC per *domain*
- To communicate securely with user in your own domain, just contact your domain’s KDC
- To talk with user in another domain, your KDC needs to contact the other domain’s KDC
  - KDCs must be able to authenticate each other and communicate securely

Hierarchy… (cont'd)



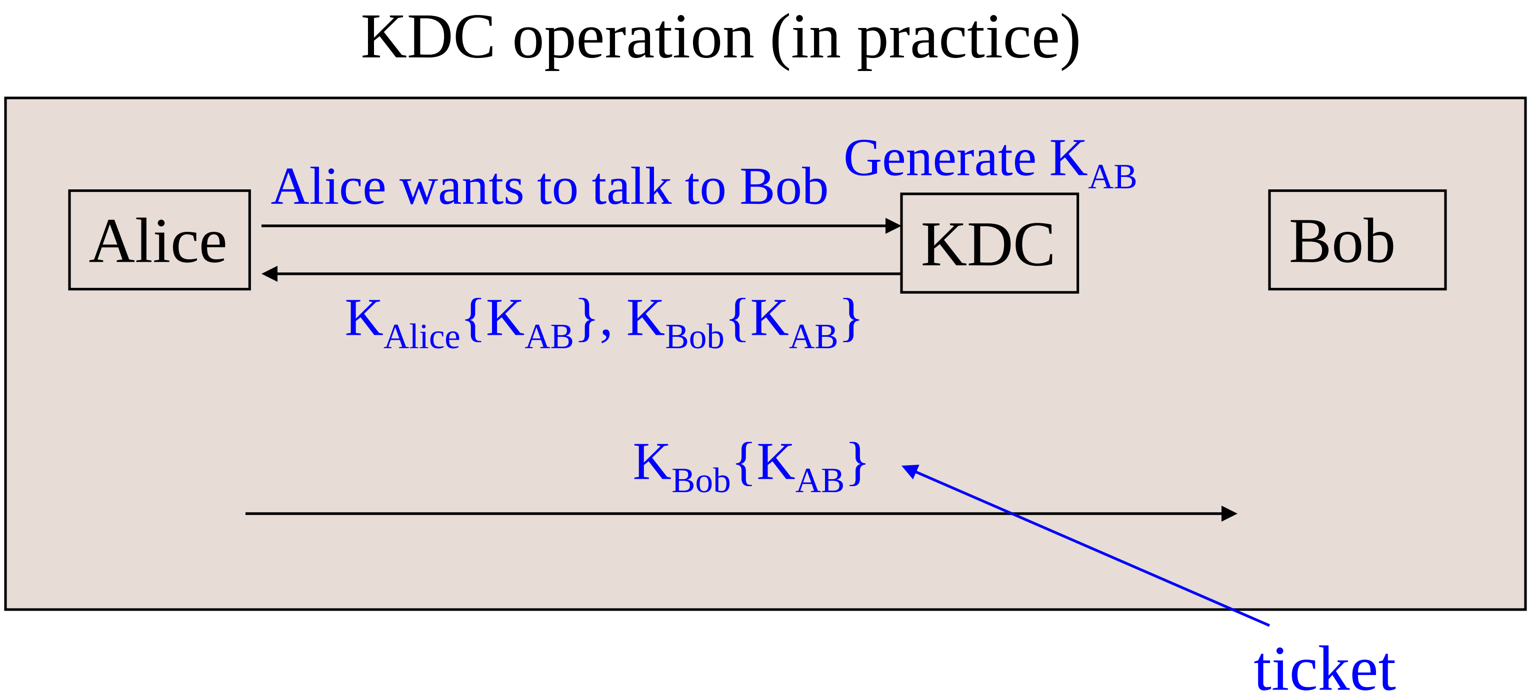
Mediated Authentication (With KDC)

KDC operation (in principle)



- Some concerns
  - Trudy may claim to be Alice and talk to KDC
    - Trudy cannot get anything useful
  - Messages encrypted by Alice using K_{AB} may arrive at Bob before KDC's message $K_{Bob}\{K_{AB}\}$ arrive
  - It may be difficult for KDC to connect to Bob

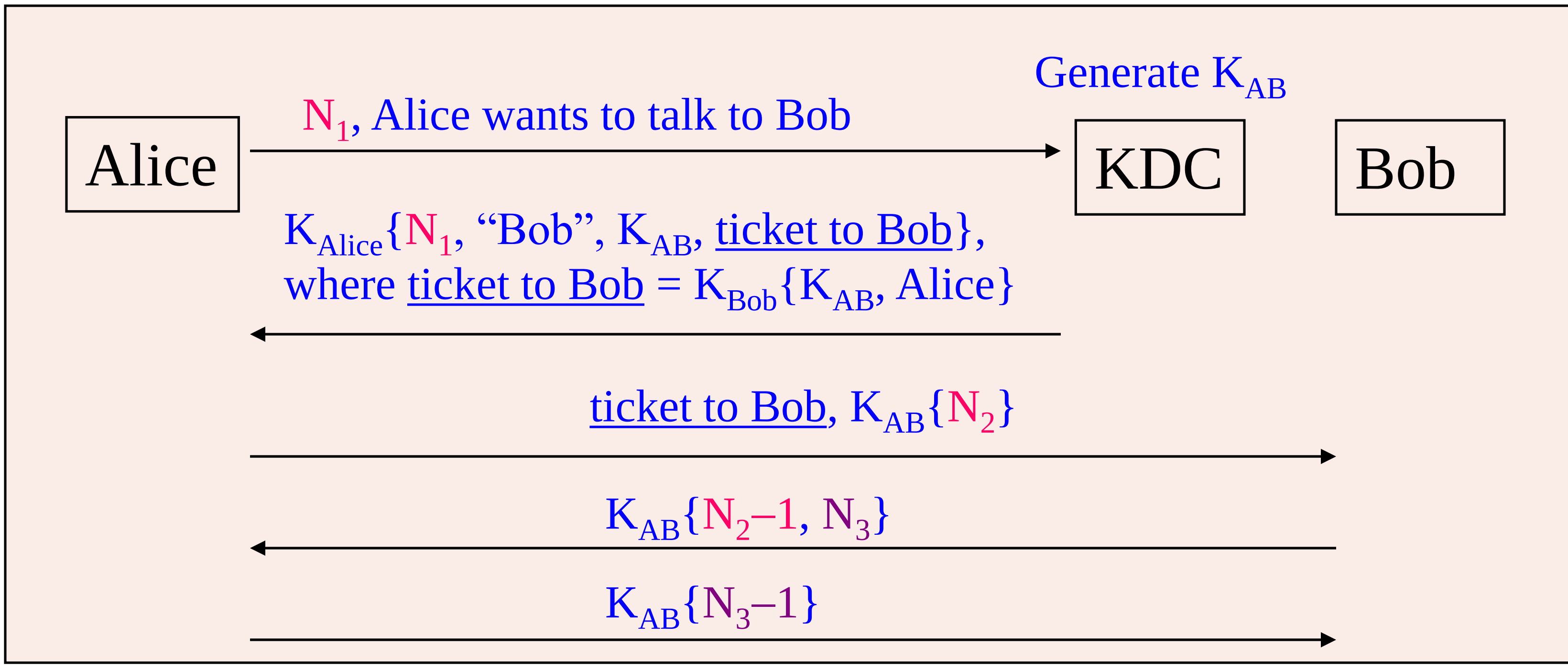
Mediated Authentication (With KDC)



- Must be followed by a mutual authentication exchange
  - To confirm that Alice and Bob have the same key

Needham-Schroeder Protocol

- Classic protocol for authentication with KDC
  - Many others have been modeled after it (e.g., Kerberos)

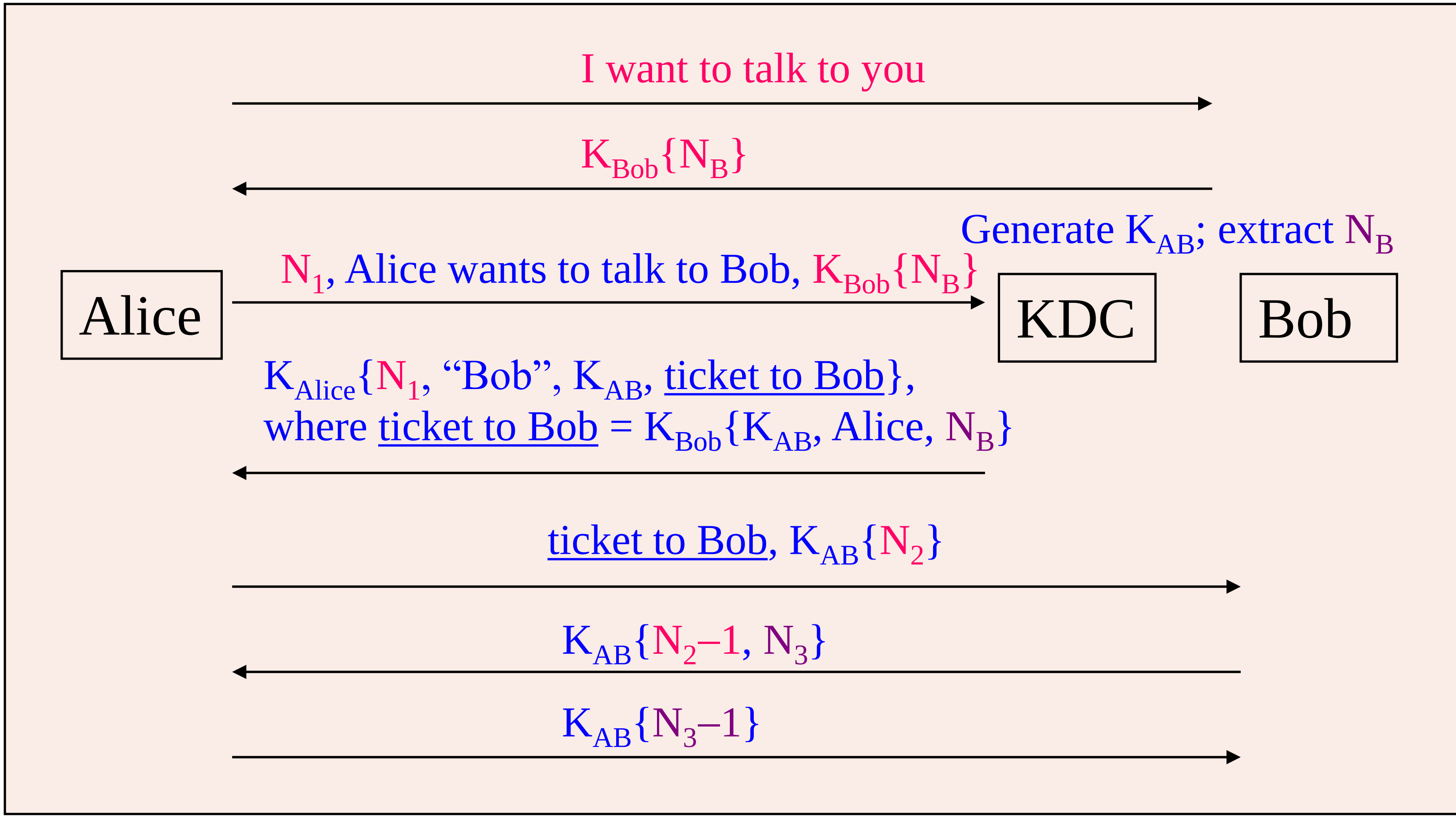


How is Bob authenticated? How is Alice authenticated? How is KDC authenticated? What are the N's used for? Why is N-1 needed?

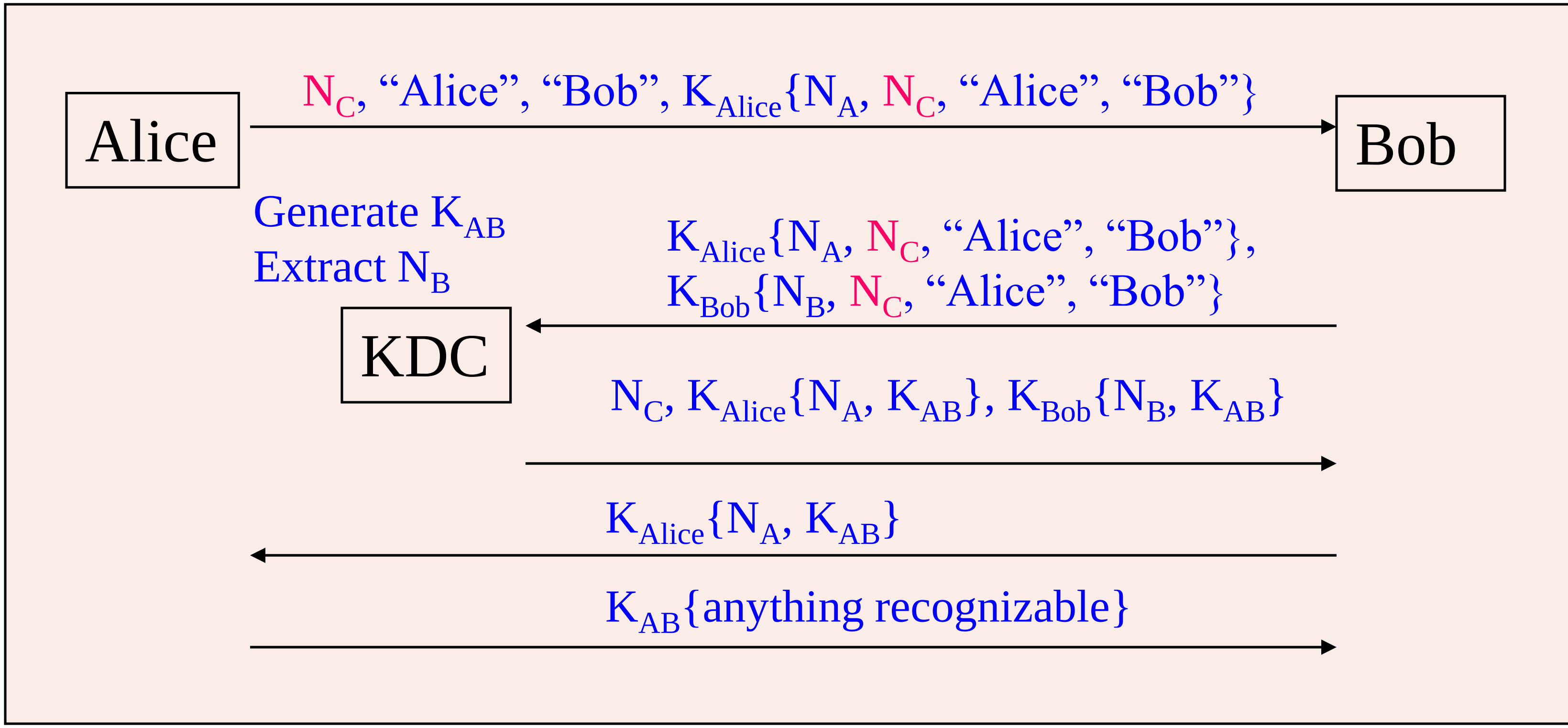
Needham-Schroeder Protocol (Cont'd)

- A vulnerability
  - When Trudy gets a previous key K_{AB} used by Alice, Trudy may reuse a previous ticket issued to Bob for Alice
  - Essential reason
    - The ticket to Bob stays valid even if Alice changes her key

Expanded Needham-Schroeder Protocol



Otway-Rees Protocol



- Only has five messages
- KDC checks if N_C matches in both cipher-texts
  - Make sure that Bob is really Bob