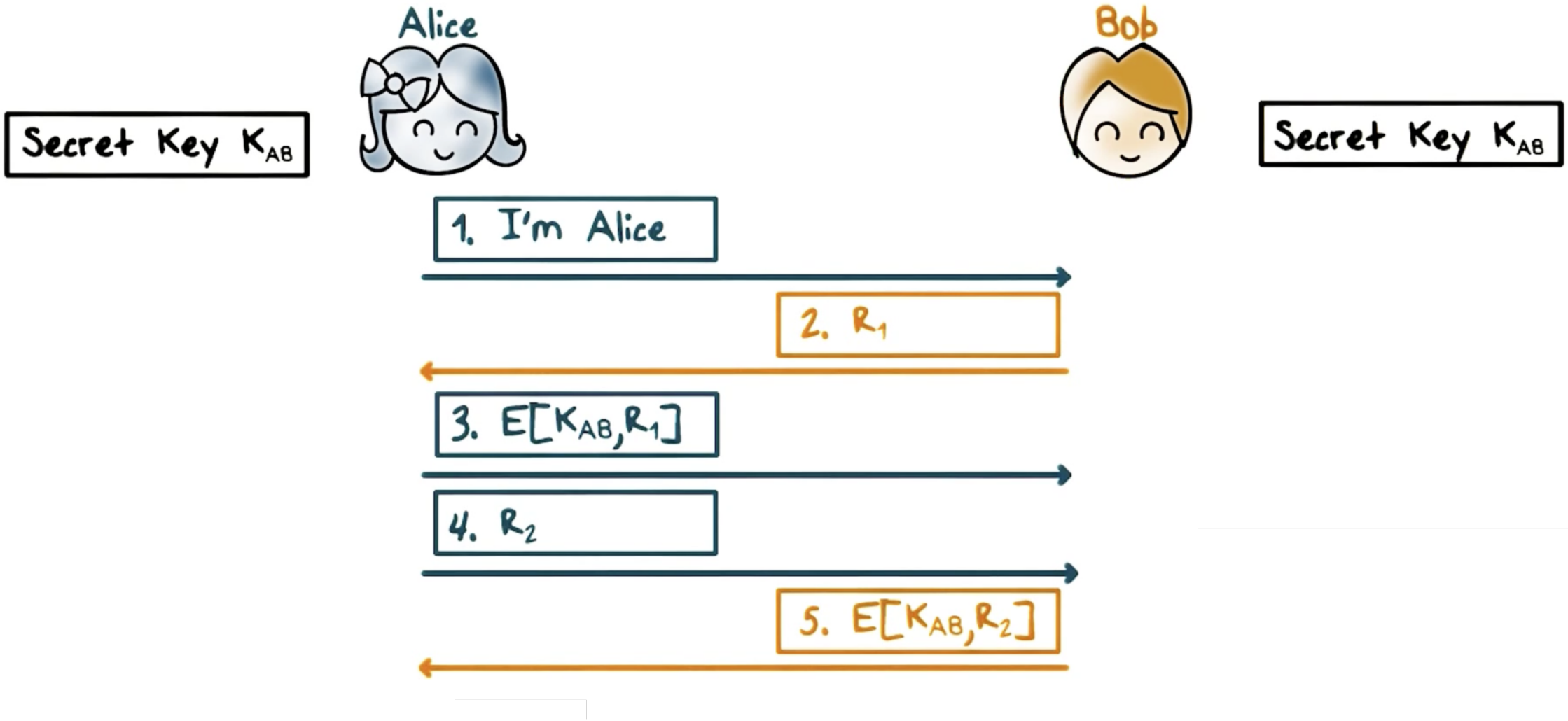
Security Protocols
Lesson Introduction

- Authentication protocols
- Key exchange protocols
- Kerberos

Why Security Protocols

- **Alice and Bob want to communicate securely over the Internet, they need to**:
  - (Mutually) authenticate
  - Establish and exchange keys
  - Agree to cryptographic operations and algorithms
- **Building blocks**:
  - Public-key (asymmetric) and secret-key (symmetric) algorithms, hash functions

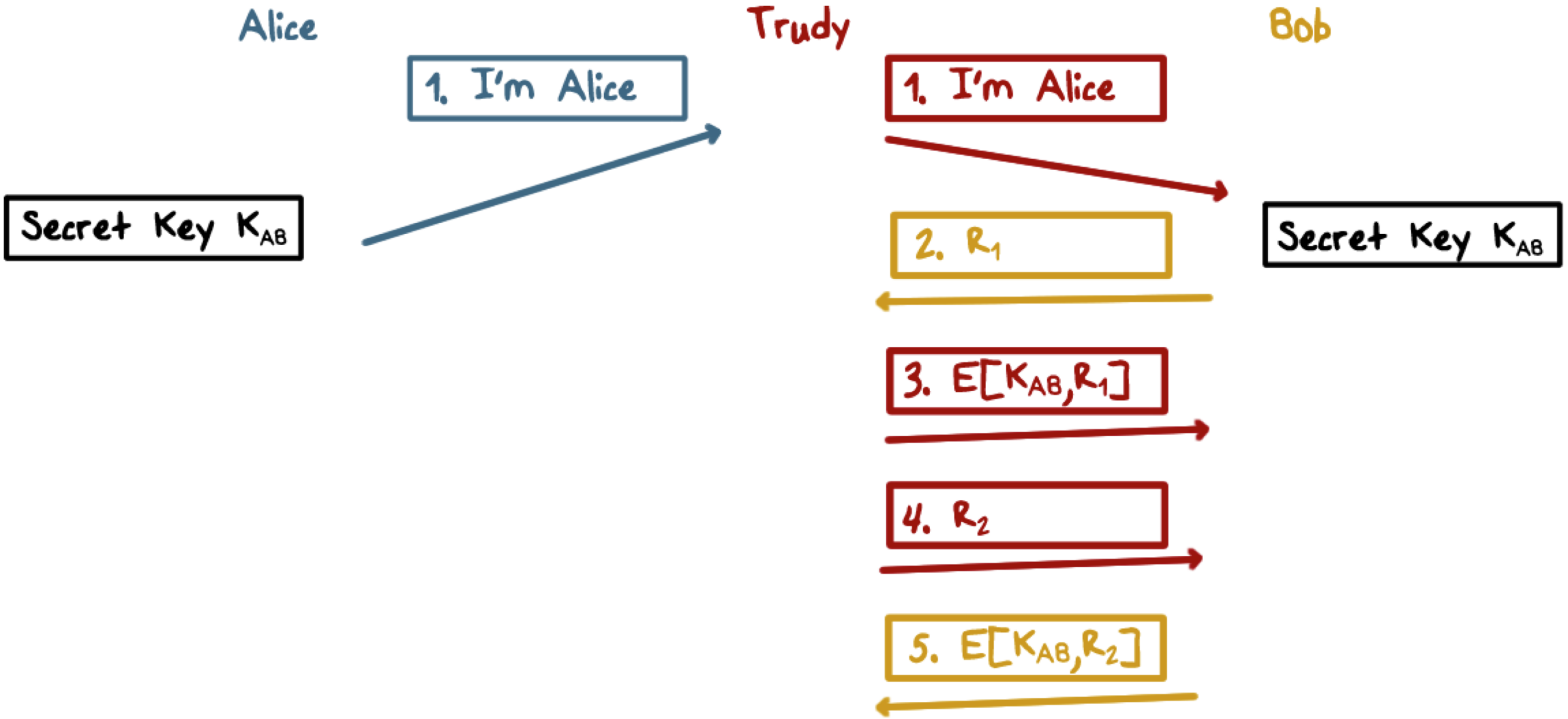
Mutual Authentication: Shared Secret



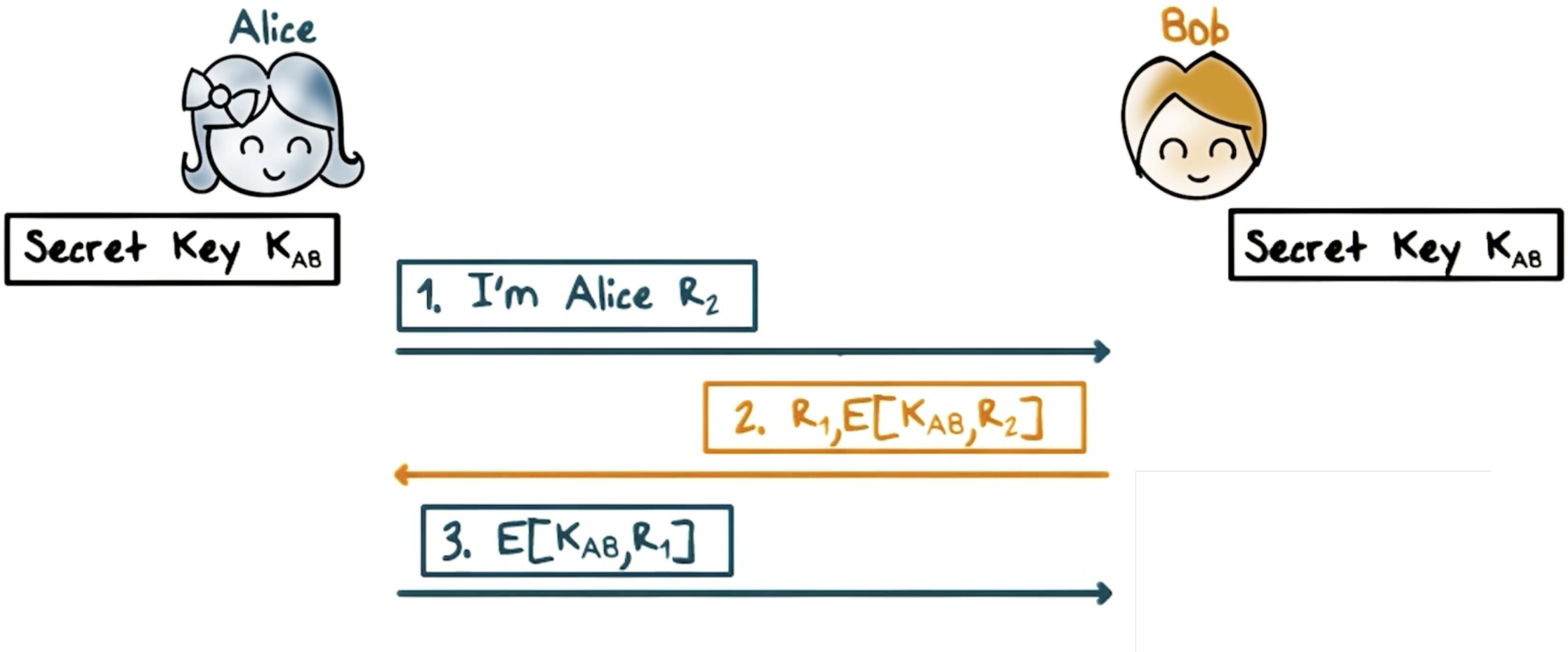
Mutual Authentication: Shared Secret

- **R_1 and R_2 should not be easily repeatable and predictable**
  - Otherwise an adversary, Trudy, can record and replay challenge and/or response to impersonate Alice or Bob
- Use **large random values**
- K_{AB} needs to be protected at Alice and Bob (end points of communication)

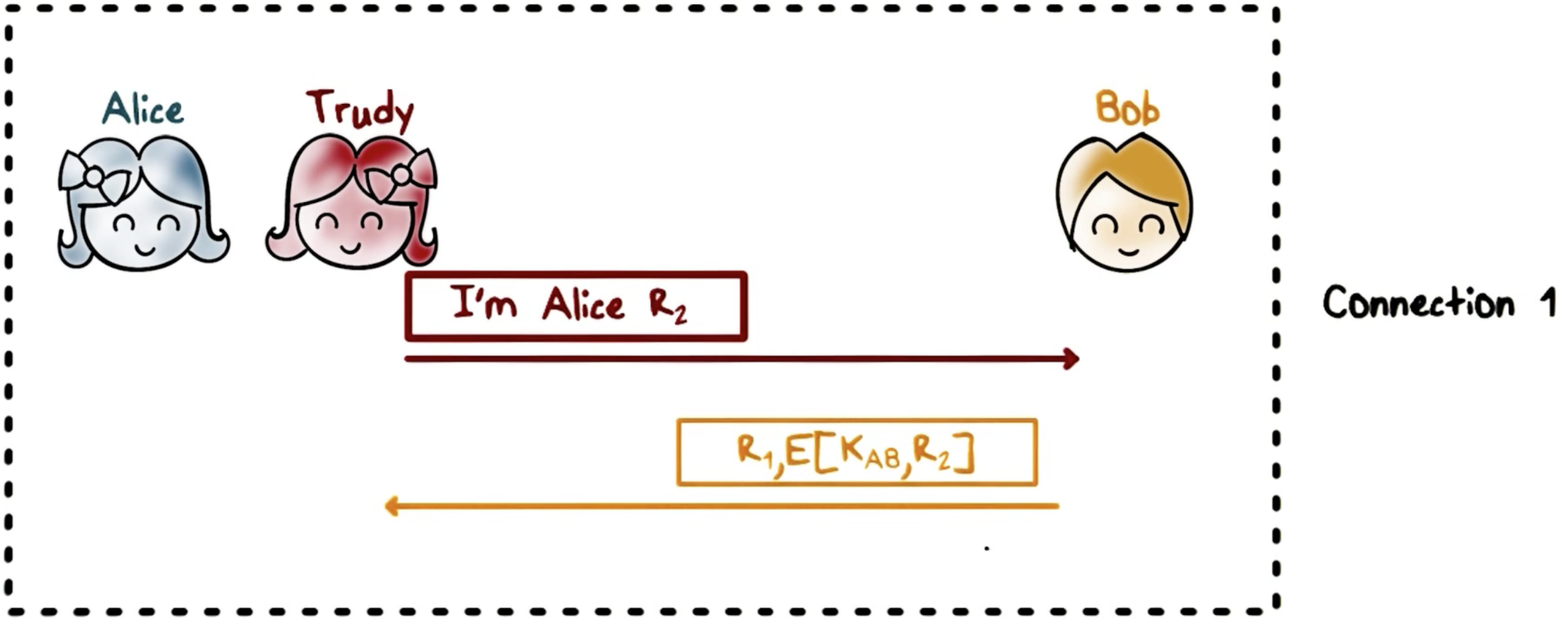
Mutual Authentication: Shared Secret



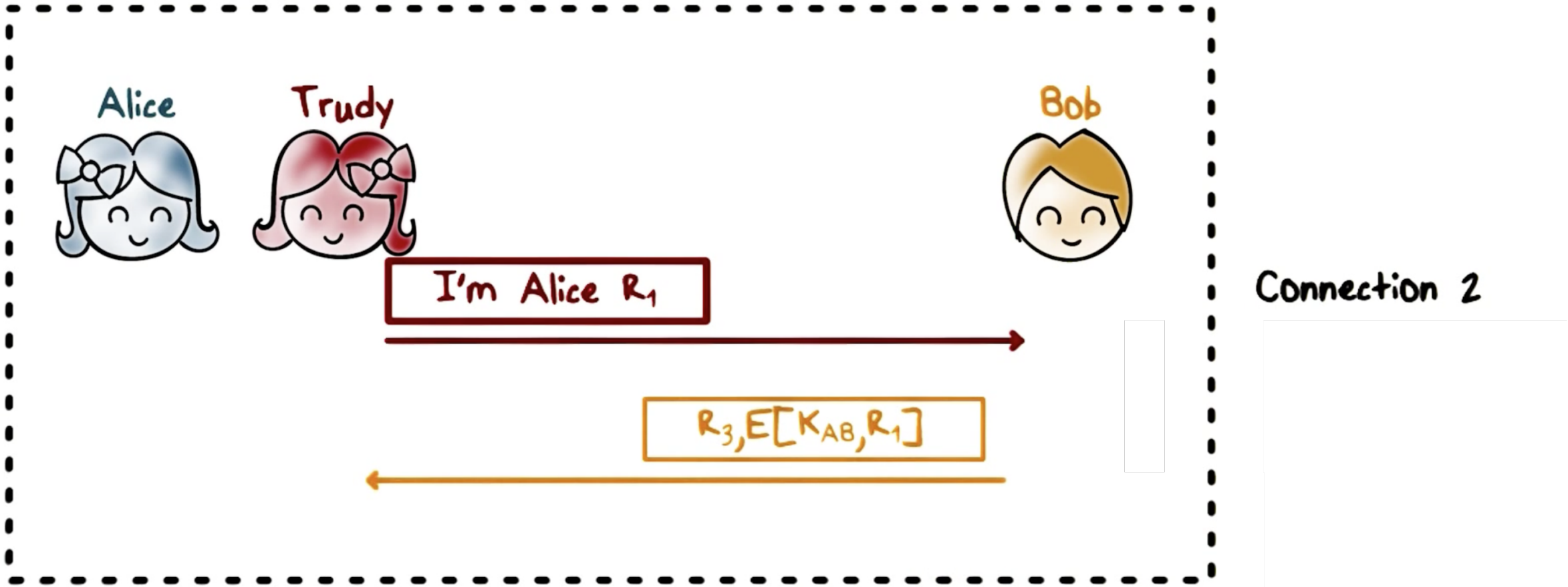
Mutual Authentication: Simplified



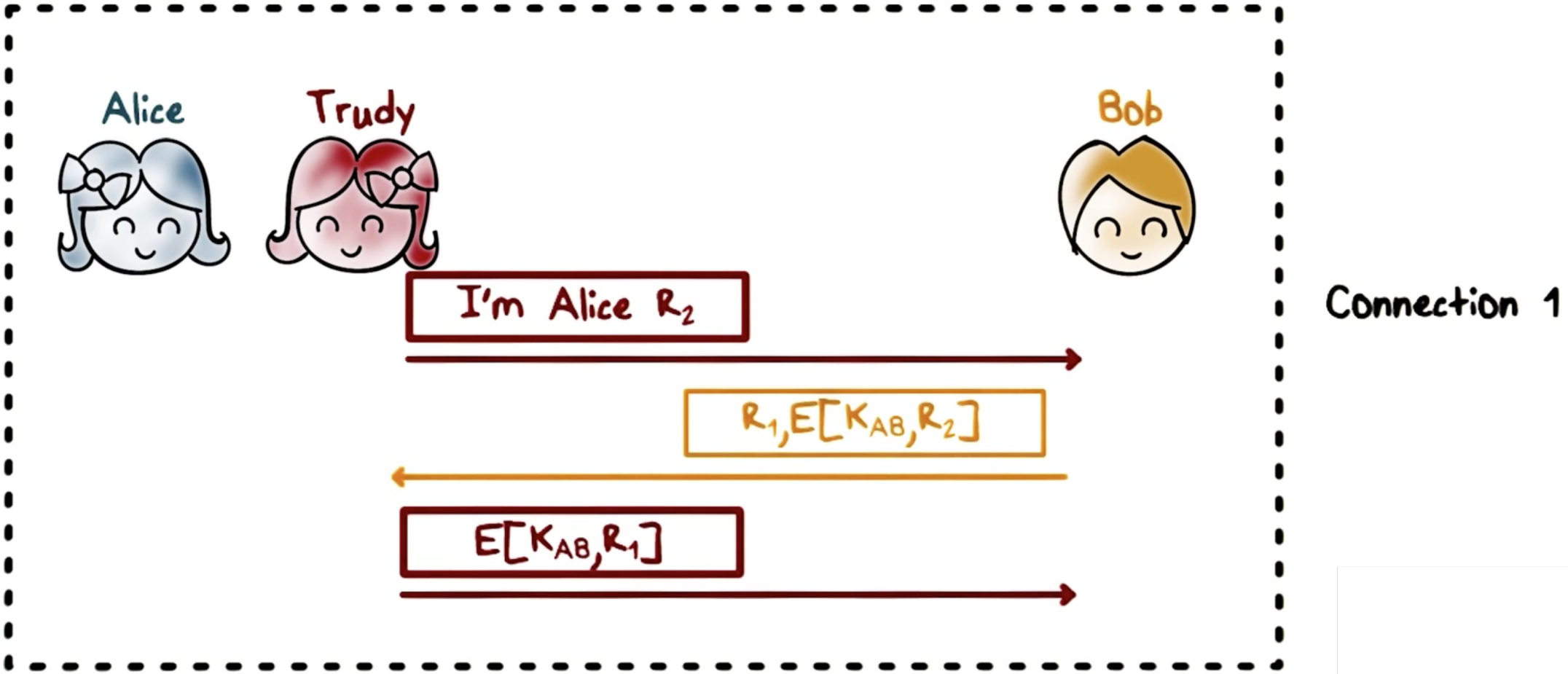
Mutual Authentication: Reflection Attack



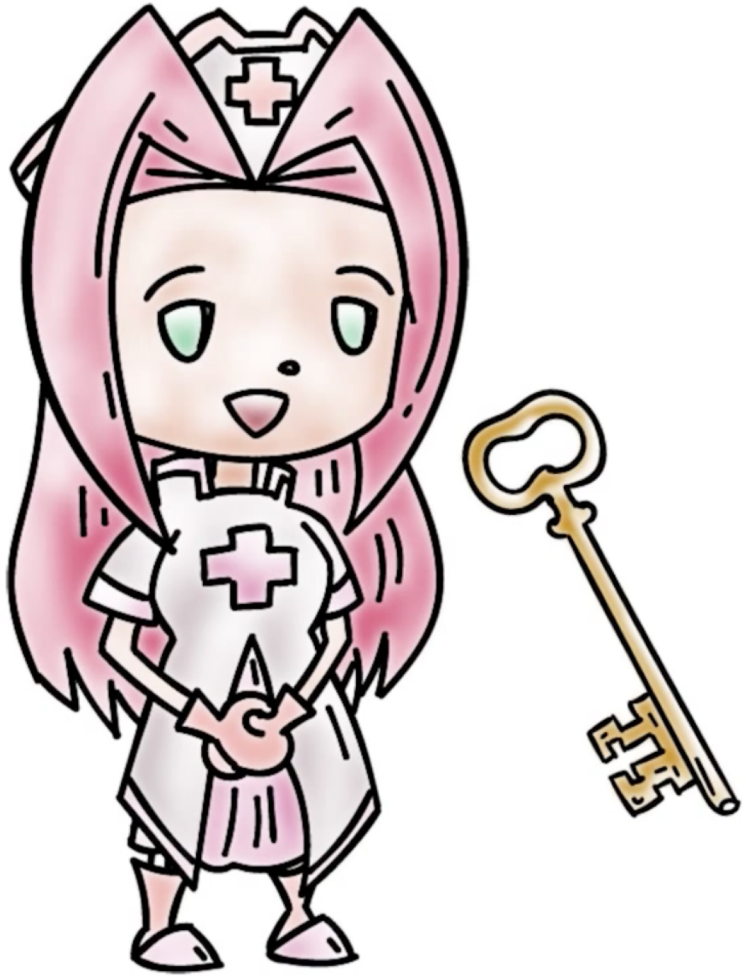
Mutual Authentication: Reflection Attack



Mutual Authentication: Reflection Attack

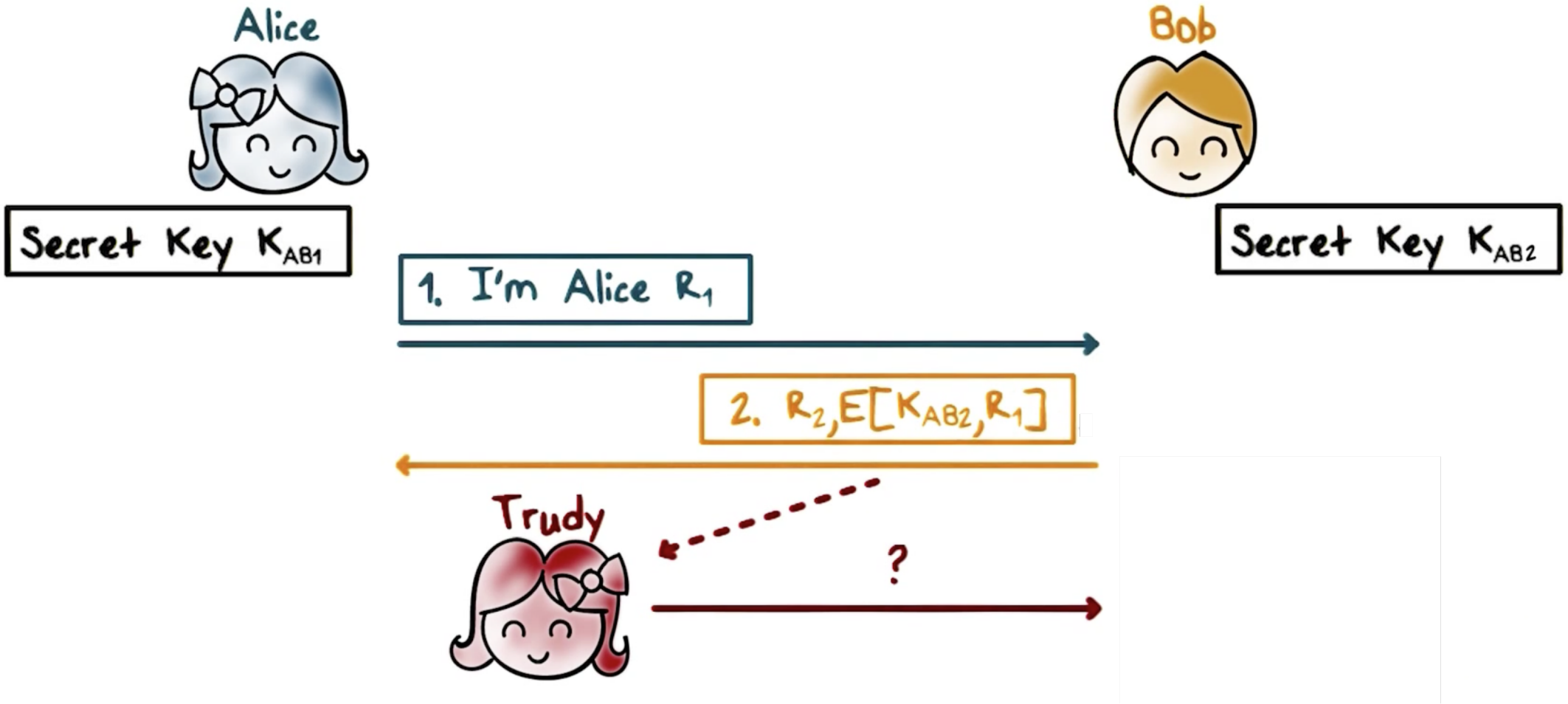


Mutual Authentication: Reflection Attack



- **Fixes:**
  - Different keys for **initiator** and **responder**
    - Trudy can’t get Bob to encrypt using Alice’s key

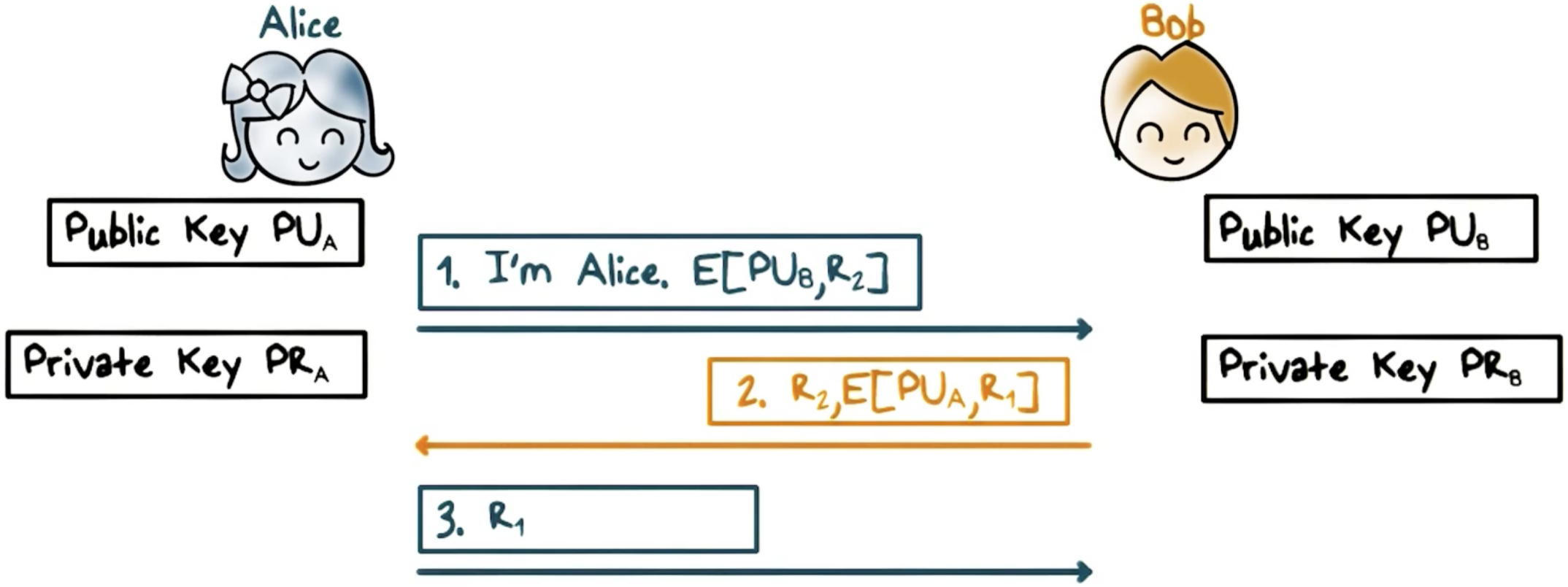
Mutual Authentication: Reflection Attack



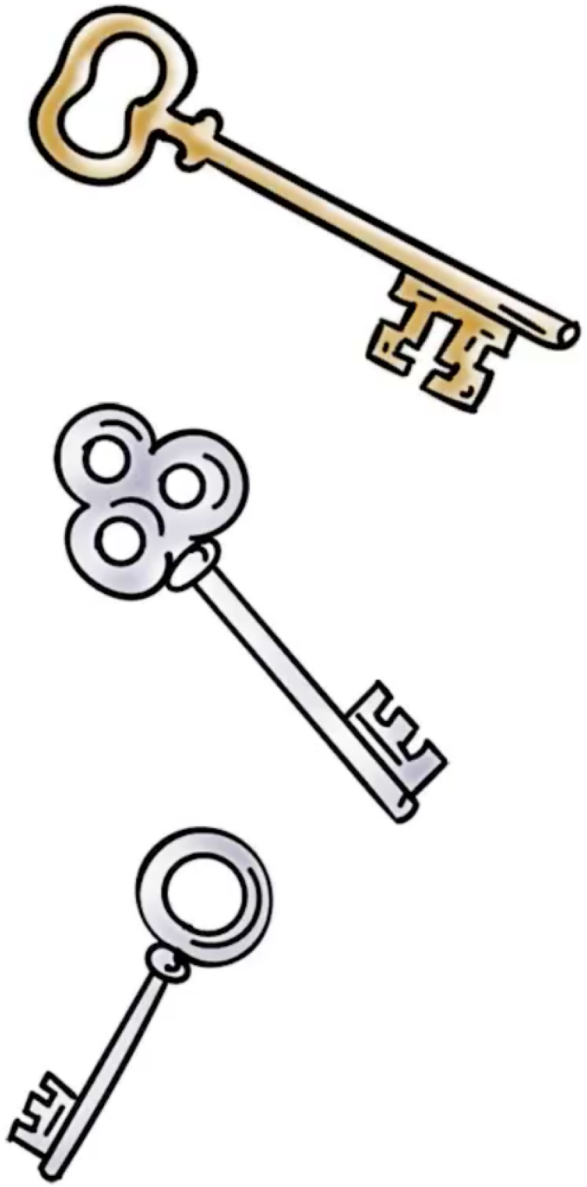
Mutual Authentication: Reflection Attack

- **Different type of challenges for initiator and responder**
  - **e.g.,** even number for initiator and odd number for responder

Mutual Authentication Public Keys

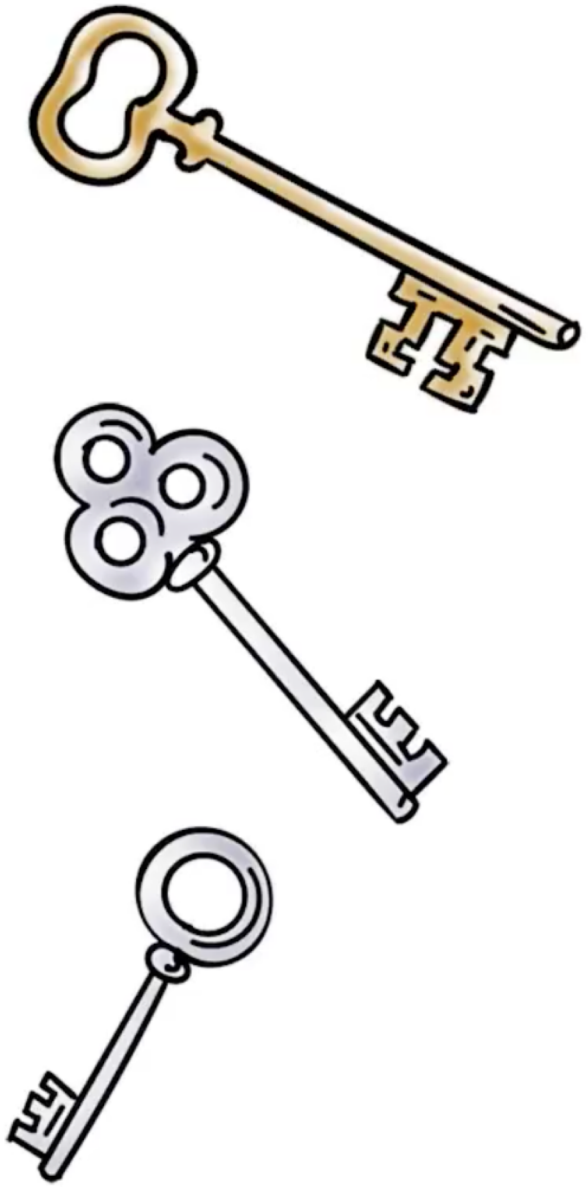


- **Variant:**
  - Sign instead of encrypt



Session Keys

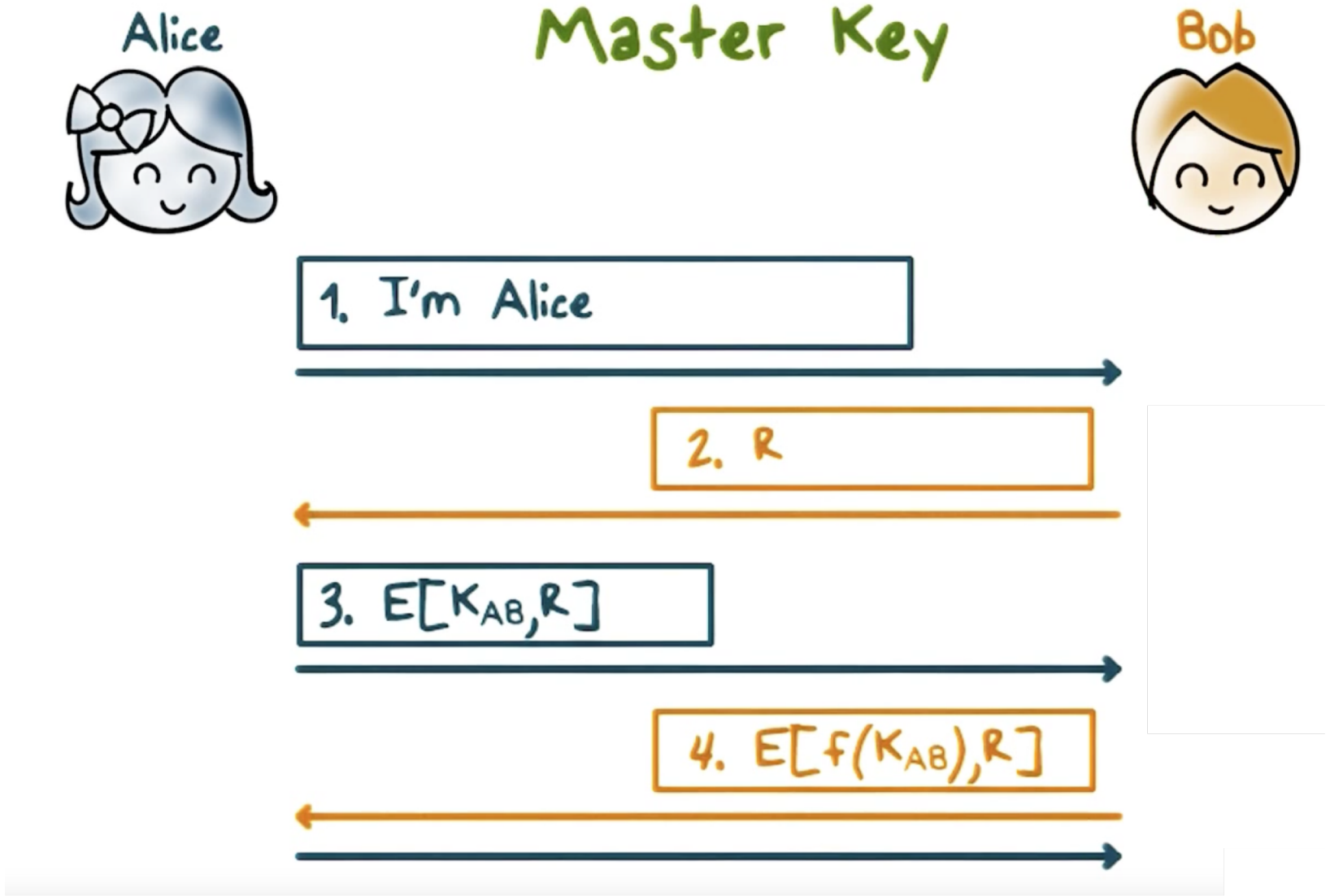
- **Authentication first**
- A new key is used for each session
- **Using shared (master) secret**
  - Encrypt the new key
- **Using public keys**

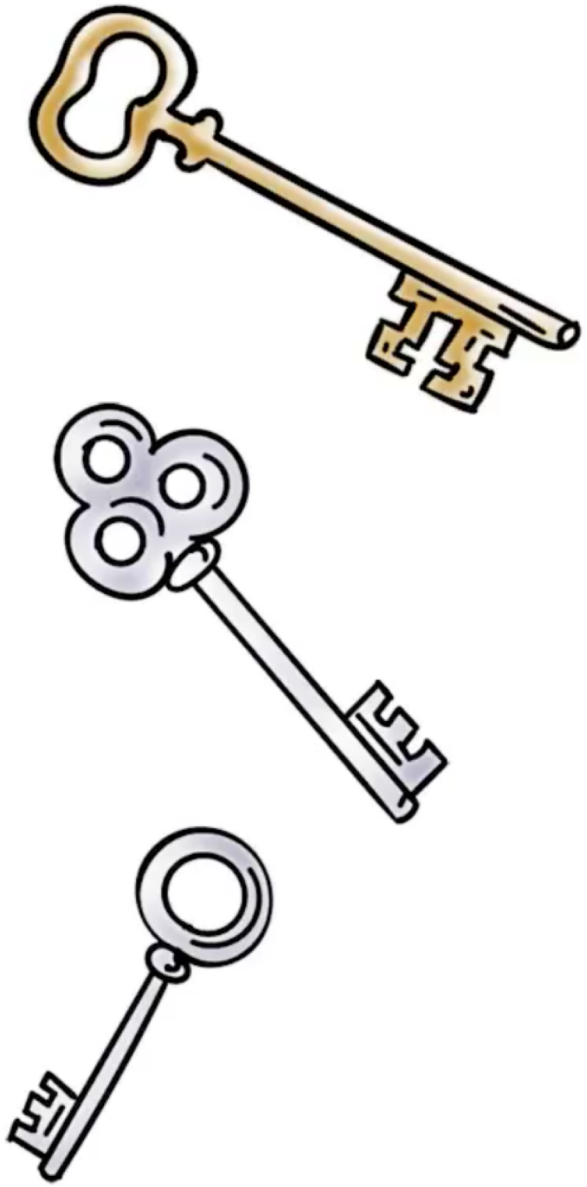


Session Keys

- **Establish a shared key for the session**, even if a there is already a shared secret key.
- **T**ypically a long term secret key is called a Master key, possibly derived from a password.
- The master key is used to **authenticate and establish a new session key**.

Session Keys





Session Keys

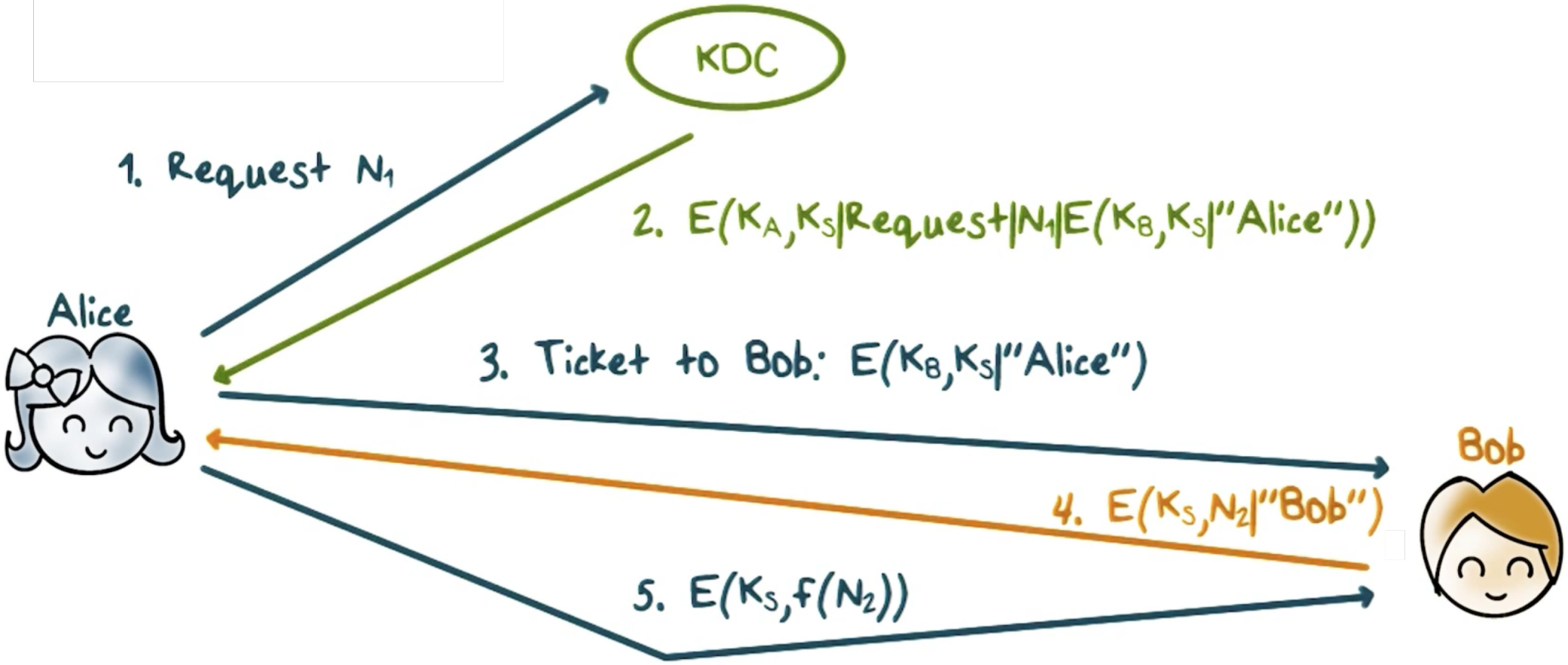
- Alice → Bob: $E(PR_A, E(PU_B, K))$
- **Diffie-Hellman with signing, i.e.,**
  - Alice → Bob: $E(PR_A, Y^A)$
  - Bob → Alice: $E(PR_B, Y^B)$

Key Distribution Center (KDC)

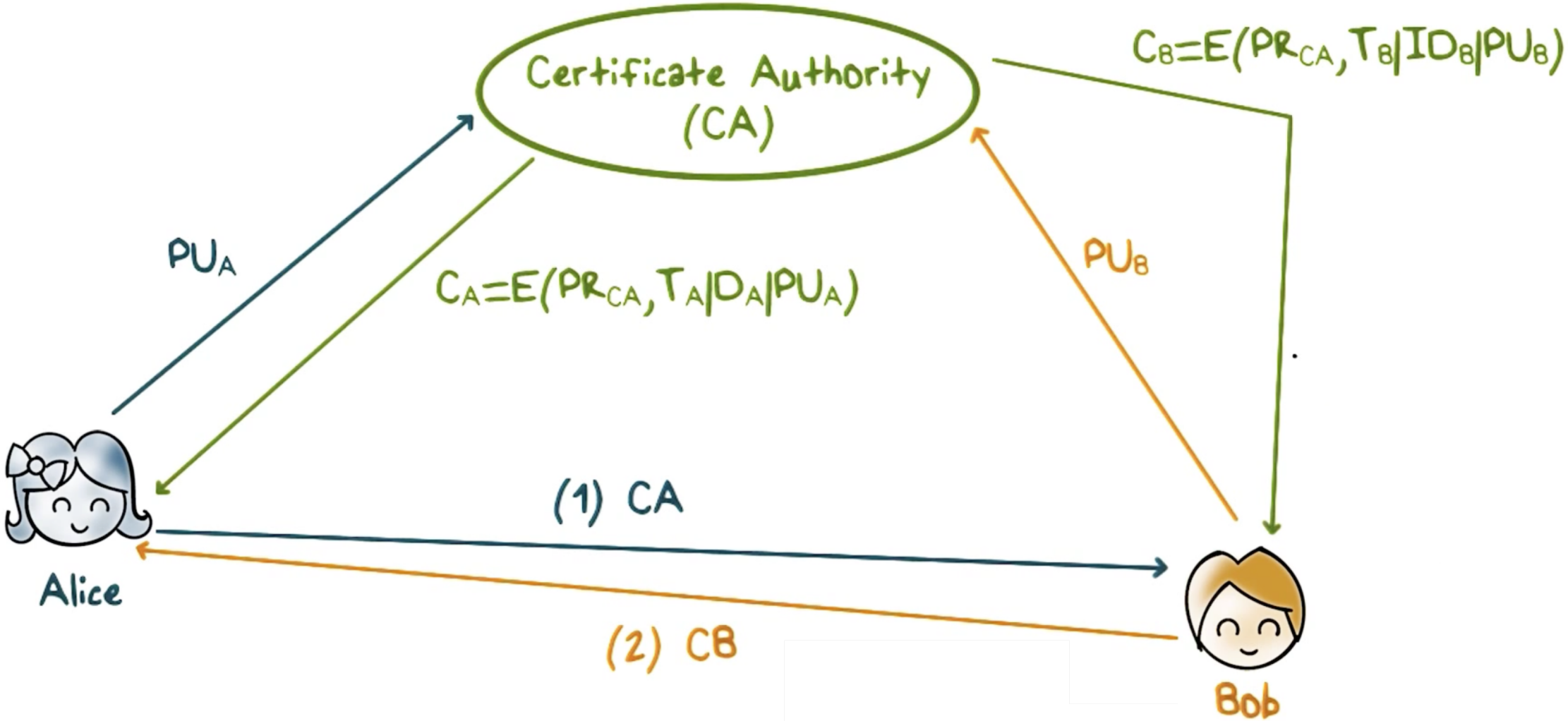
- **Shared Master Keys do not scale easily**
- Each communication pair needs to share a master key

Key Distribution Center (KDC)

K_A, K_B are master keys shared with KDC, K_s is a session key



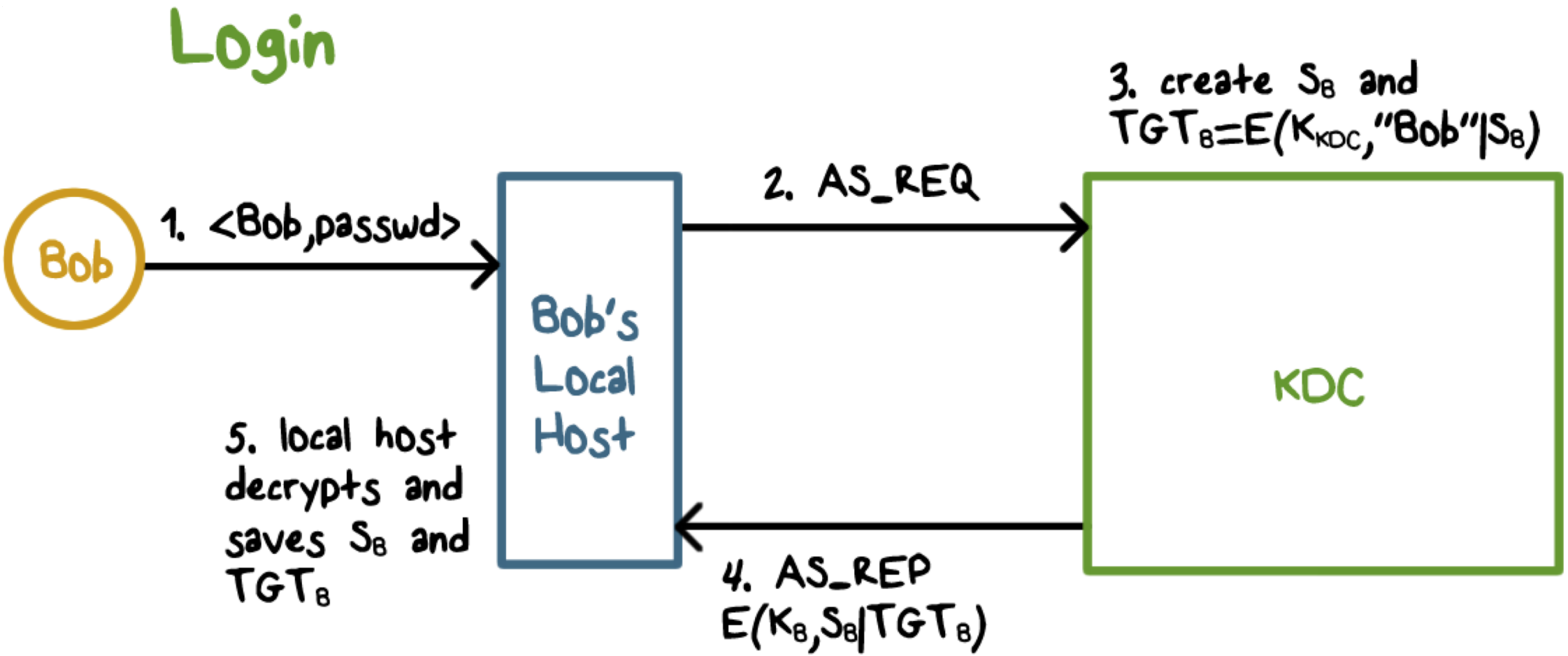
Exchanging Public Key Certificates



Kerberos

- Authentication and access control in a network environment
- Every principal has a master (secret) key
  - Human user's master key is derived from password
  - Other resources must have their keys configured in
- All principals' master keys are stored in the KDC database, protected/encrypted

Kerberos


Login
Bob
1. <Bob,passwd>
Bob's Local Host
2. AS_REQ
3. create S_B and $TGT_B = E(K_{KDC}, "Bob" | S_B)$
KDC
4. AS_REP $E(K_B, S_B | TGT_B)$
5. local host decrypts and saves S_B and TGT_B

Kerberos

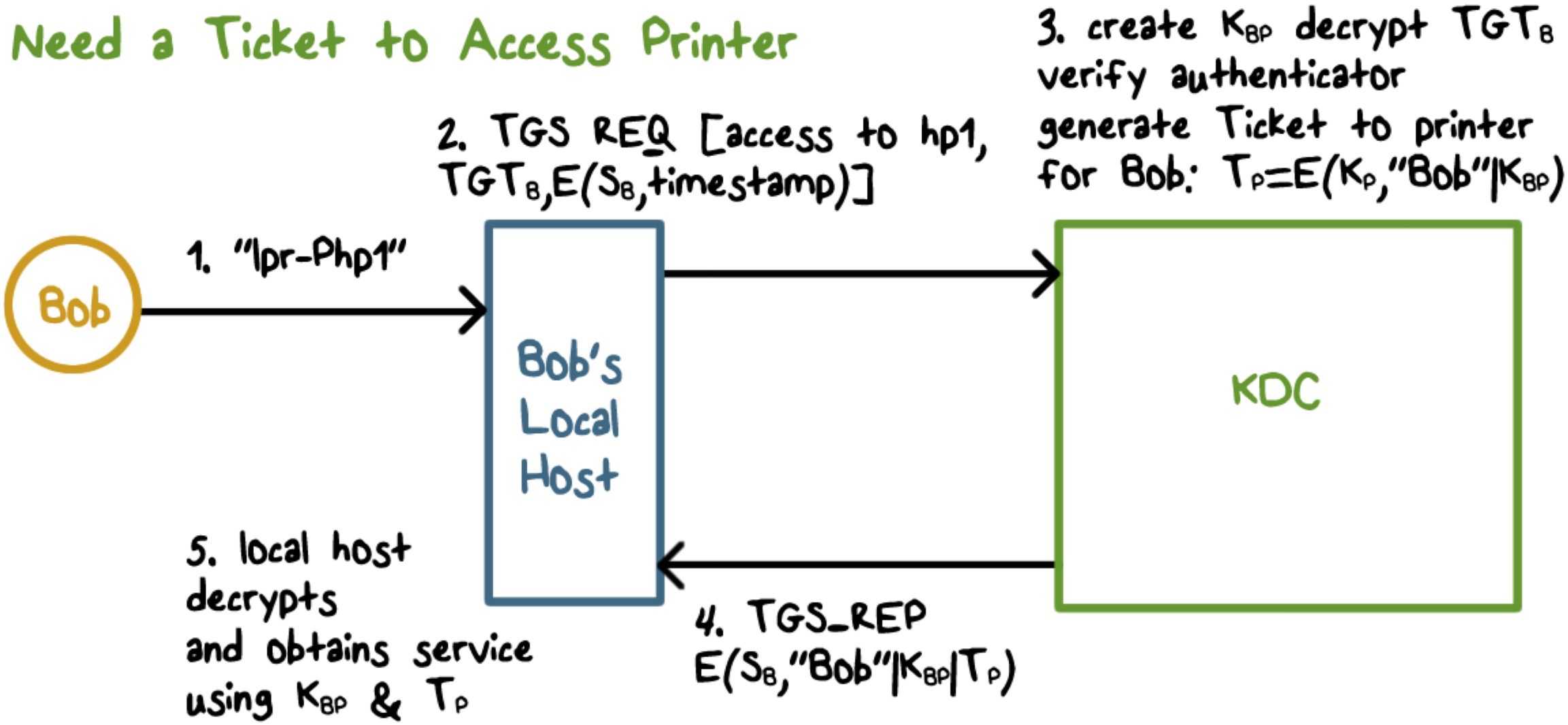
Kerberos Benefits:

- Localhost does not need to store passwords
- The master key that the user shares with the KDC is only used once every day

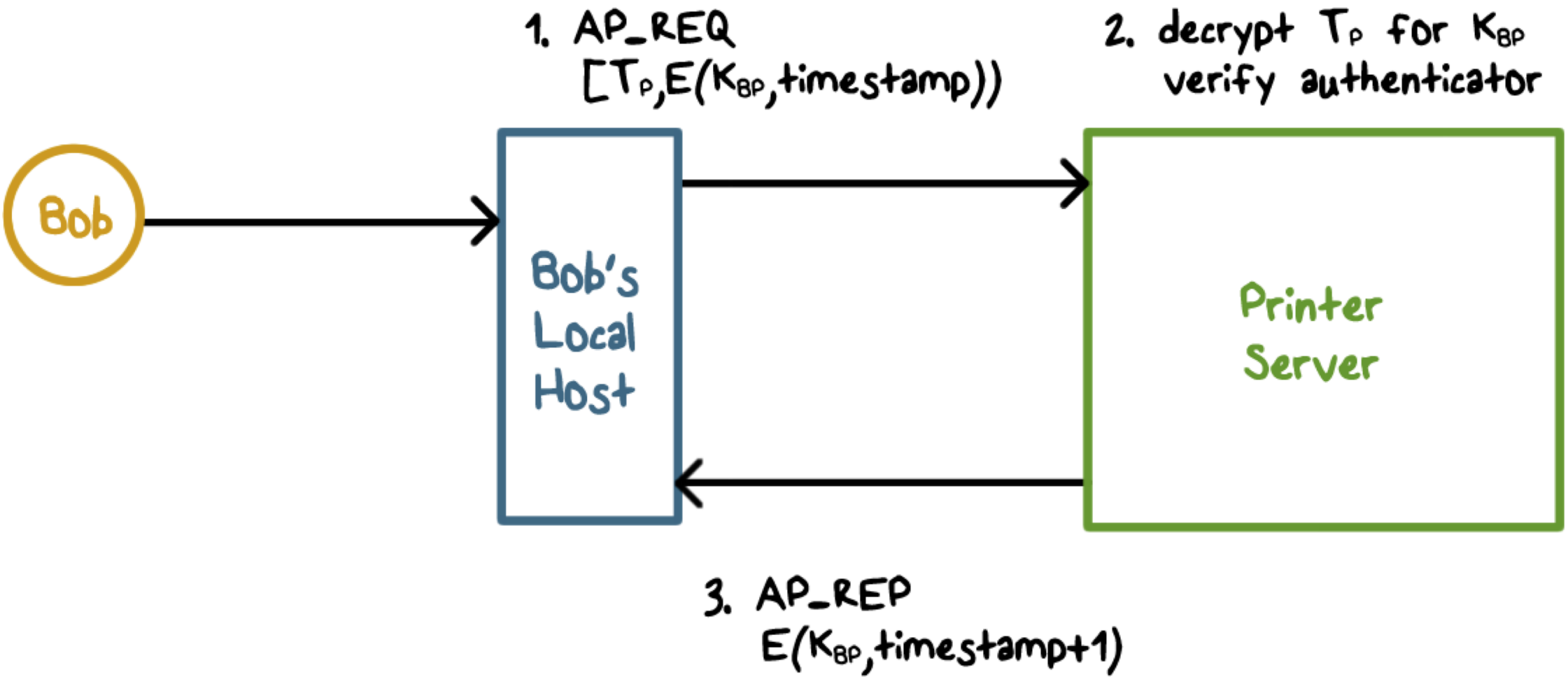
This limits exposure of the master key

Accessing the Printer

Need a Ticket to Access Printer



Accessing the Printer



Security Protocols

Lesson Summary

- Secret key based and public key based authentication
  - Random challenge and response
  - Impersonation attacks
- Establish session key based on pre-shared secret key or public keys and authentication exchanges, use KDC or CA
- Kerberos: authentication and access control, tickets, and ticket-granting ticket.