CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

SECURITY

Lecture C

IMPLEMENTING MECHANISM USING CRYPTOGRAPHY

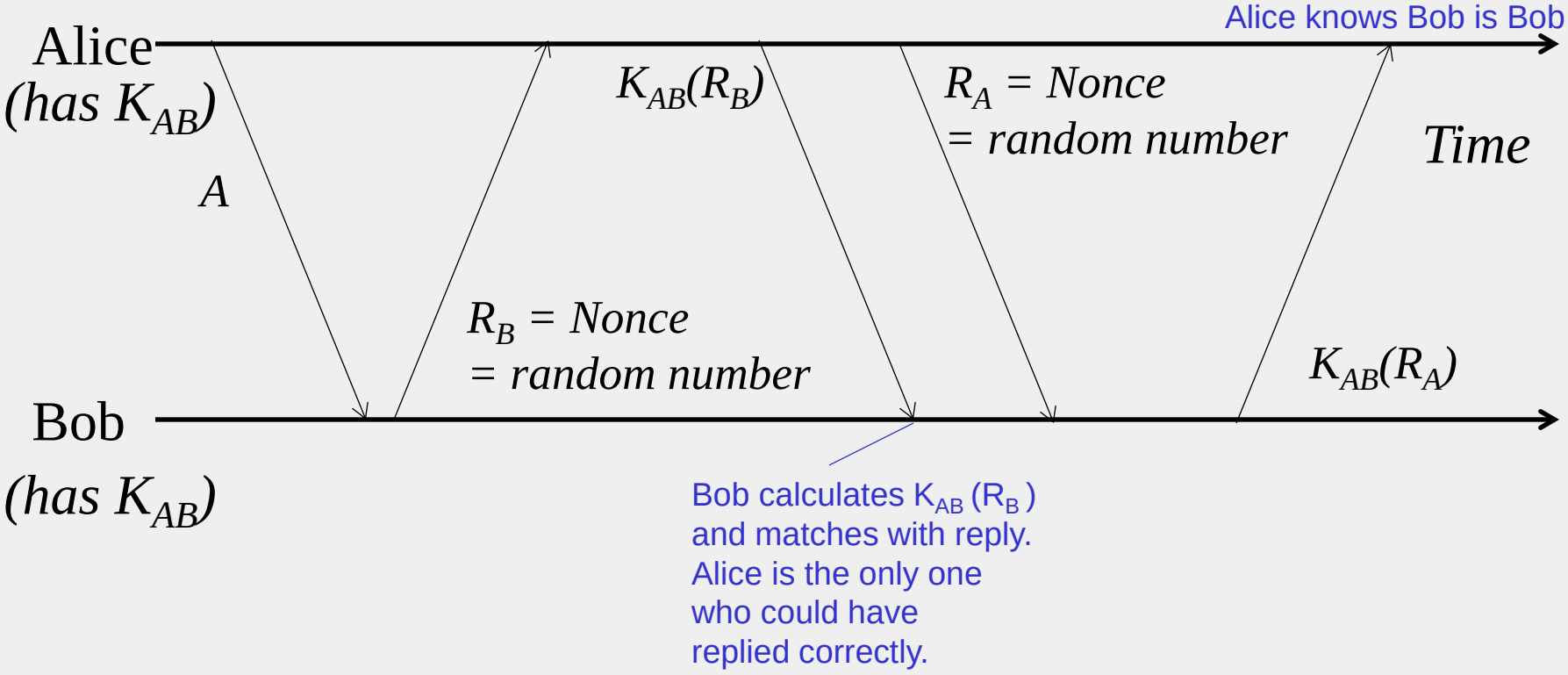
In this Lecture

- How to use cryptography to implement
  I. Authentication
  II. Digital Signatures
  III. Digital Certificates

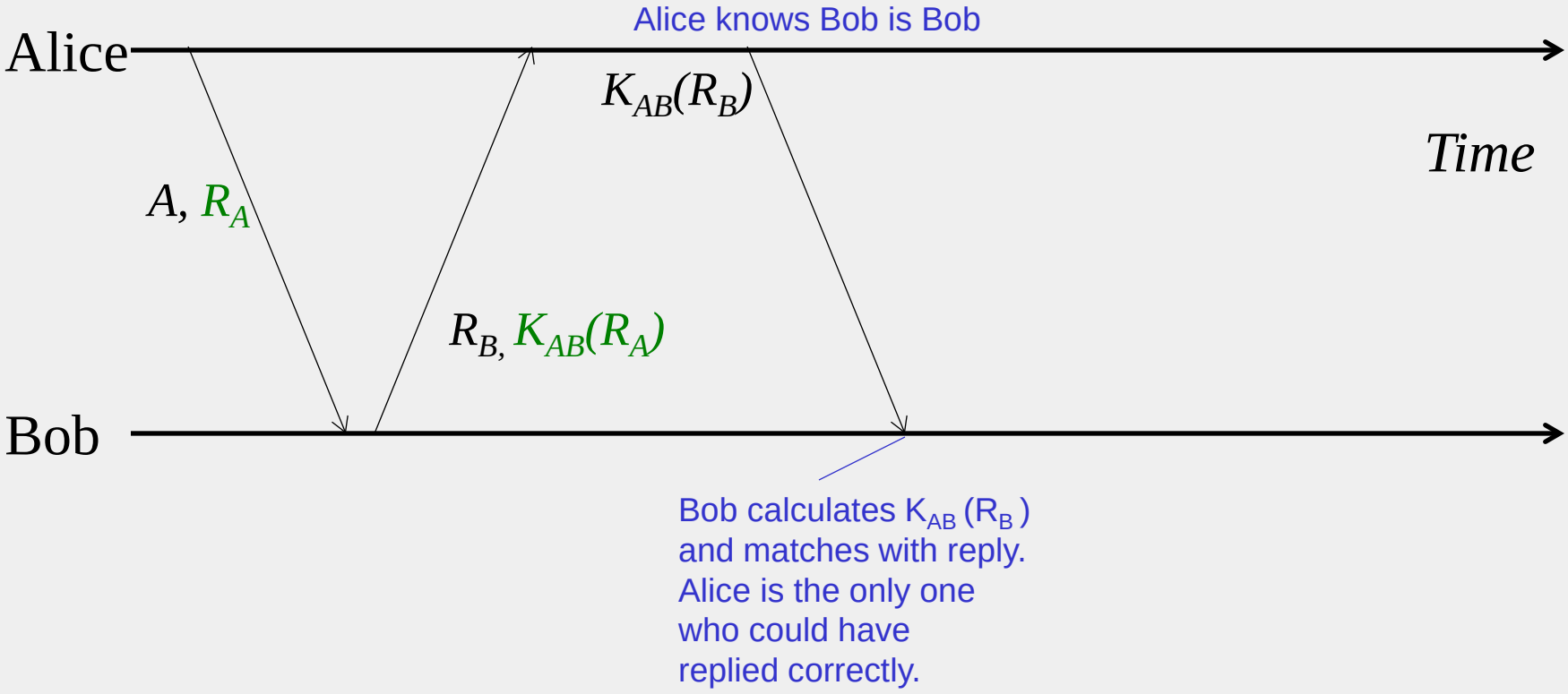
I. Authentication

- Two principals verify each other's identities
- Two flavors
  - **Direct authentication**: directly between two parties
  - **Indirect authentication**: uses a trusted third-party server
    - Called authentication server
    - E.g., a Verisign server

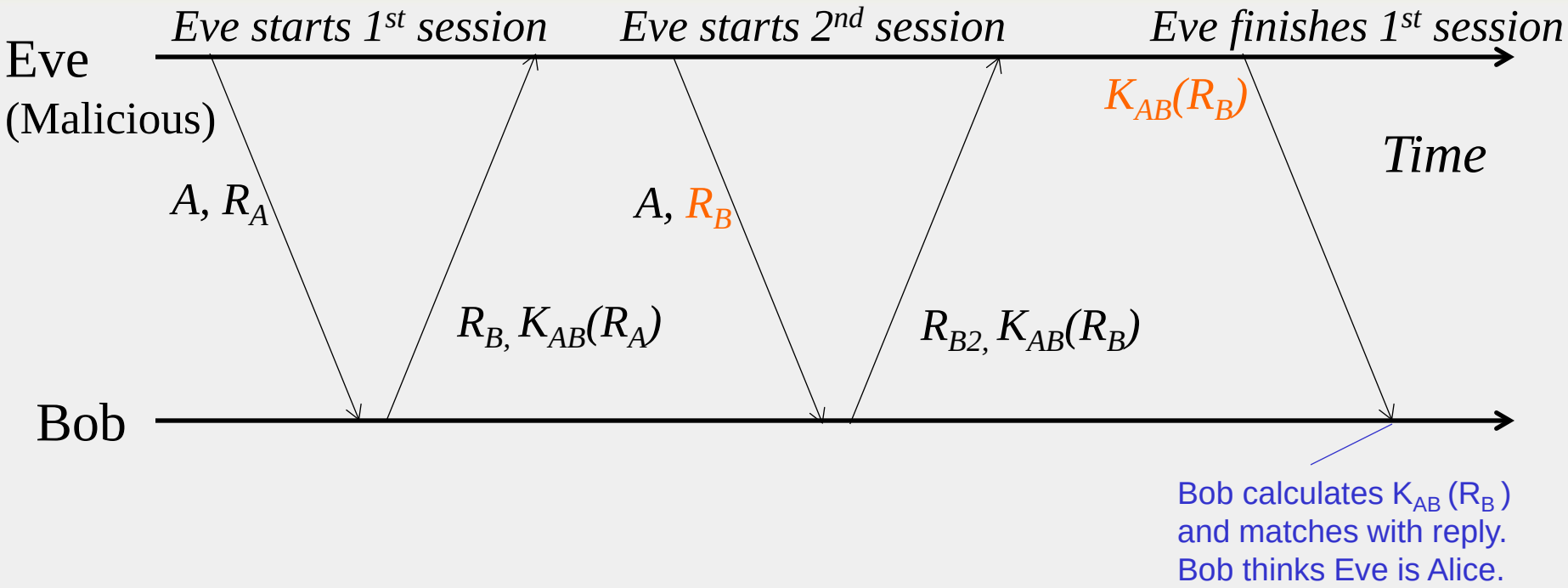
Direct Authentication Using Shared Key



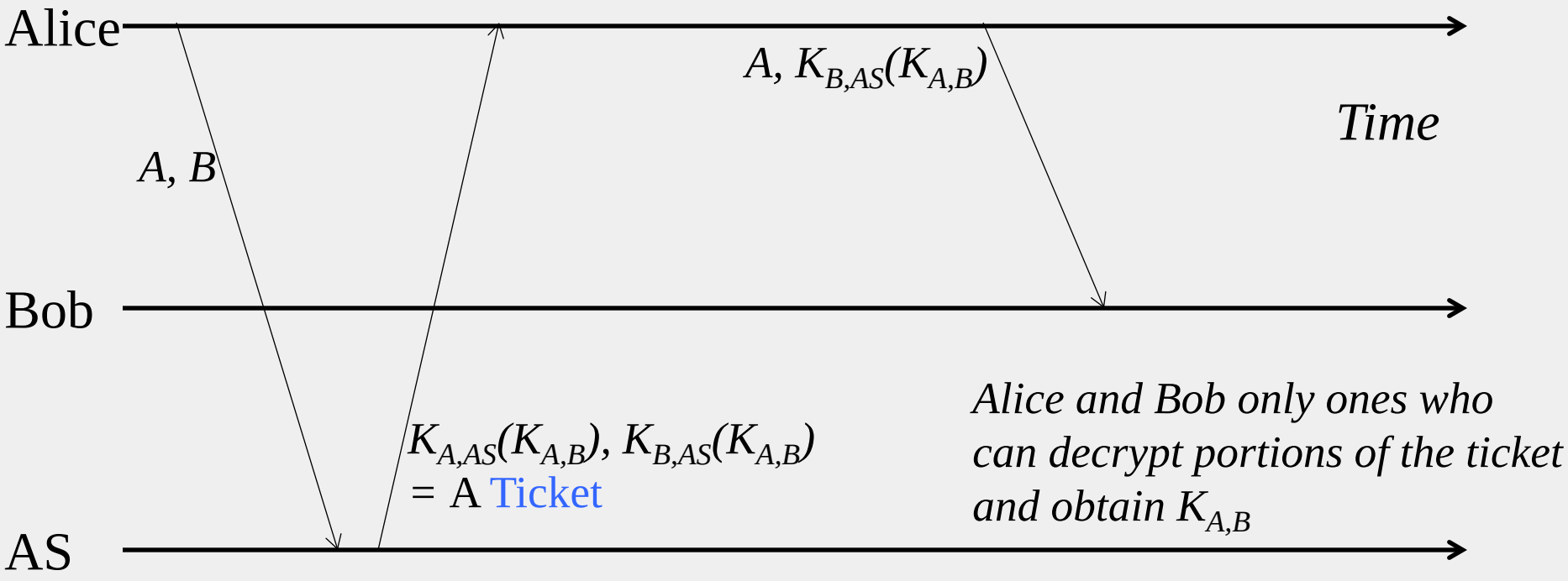
Why Not Optimize Number of Messages?



Unfortunately, This Subject to Replay Attack



Indirect Authentication Using Authentication Server and Shared Keys



II. Digital Signatures

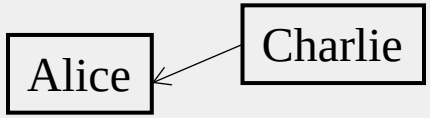
- Just like “real” signatures
  - Authentic, Unforgeable
  - Verifiable, Non-repudiable
- To sign a message M, Alice encrypts message with her own private key
  - Signed message: [M, $K_{Apriv}(M)$]
  - Anyone can verify, with Alice’s public key, that Alice signed it
- To make it more efficient, use a one-way hash function, e.g., SHA-1, MD-5, etc.
  - Signed message: [M, $K_{Apriv}(Hash(M))$]
  - Efficient since hash is fast and small; don’t need to encrypt decrypt full message

III. Digital Certificates

- Just like “real” certificates
- Implemented using digital signatures
- Digital Certificates have
  - Standard format
  - Transitivity property, i.e., chains of certificates
  - Tracing chain backwards must end at trusted authority (at root)

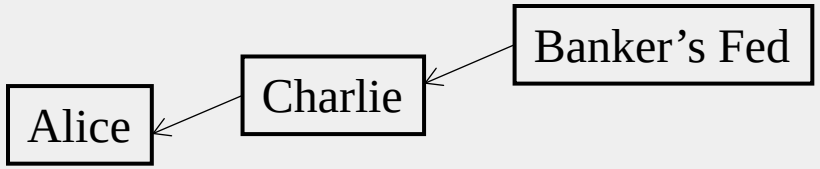
Example: Alice's Bank Account

1. Certificate Type: Account
2. Name: Alice
3. Account number: 12345
4. Certifying Authority: Charlie's Bank
5. Signature
   - K_{Cpriv}(Hash(Name+Account number))



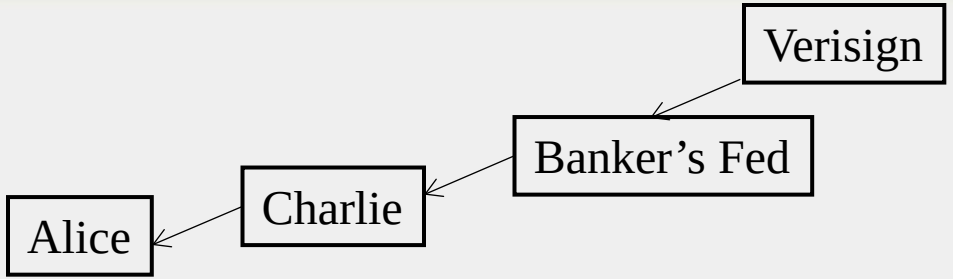
Charlie's Bank, in Turn Has another Certificate

1. Certificate Type: Public Key
2. Name: Charlie's Bank
3. Public Key: K_{Cpub}
4. Certifying Authority: Banker's Federation
5. Signature
   - K_{Fpriv}(Hash(Name+Public key))



Banker's Federation, Has another Certificate from the Root Server

1. Certificate Type: Public Key
2. Name: Banker's Federation
3. Public Key: K_{Fpub}
4. Certifying Authority: Verisign
5. Signature
   - $K_{verisign\ priv}$(Hash(Name+Public key))



IV. Authorization

- **Access Control Matrix**
  - For every combination of (principal,object) say what mode of access is allowed
  - May be very large (1000s of principals, millions of objects)
  - May be sparse (most entries are “no access”)
- **Access Control Lists (ACLs)** = per object, list of allowed principals and access allowed to each
- **Capability Lists** = per principal, list of files allowed to access and type of access allowed
  - Could split it up into capabilities, each for a different (principal,file)

Security: Summary

- Security challenges abound
  - Lots of threats and attacks
- CIA properties are desirable policies
- Encryption and decryption
- Shared key vs public/private key systems
- Implementing authentication, signatures, certificates
- Authorization