CSE/CEN 598
Hardware Security & Trust

Public Key Infrastructure

Prof. Michel A. Kinsy

Public Key Infrastructure (PKI)

- What is Public Key Infrastructure (PKI)?
  - Enables users to securely and privately exchange data over an unsecured medium without the loss of integrity or confidentiality
  - "A PKI is a set of agreed-upon standards, Certification Authorities (CA), structure between multiple CAs, methods to discover and validate Certification Paths, Operational Protocols, Management Protocols, Interoperable Tools and supporting Legislation"
    - "Digital Certificates" book by J. Feghhi, J. Feghhi, and P. Williams
    - "agreed-upon standards"
      - By who?
    - "Interoperable Tools and supporting Legislation"
      - Interesting
    - "Certification Authorities (CA)"
      - This is a technical nugget that we can work with

Public Key Infrastructure (PKI)

- A PKI is an Infrastructure to support and manage Public Key-based Digital Certificates
  - Couple concepts
    - Public Key
    - Digital Certificates
- What are the functions and components of PKI
  - Certification authority (CA)
  - Registration authority (RA)
  - PKI clients
  - Digital certificates
  - Certificate Distribution System or repository
  - Keys (Public and Private)

Public-Key Cryptography

- A Public-key cryptography (PKC) is a two-key protocol system
  - It uses one key for encryption and another for decryption
  - It also called asymmetric encryption
  - It is primarily used for authentication, non-repudiation, and key exchange
- PKC depends upon the existence of so-called one-way functions or mathematical functions that are easy to compute whereas their inverse function is relatively difficult to compute
- There are three classes of cryptosystems
  - Message Digest
  - Secret Key
  - Public Key

The Three Cryptosystem Classes

- Message Digest
  - Maps variable length plaintext into fixed length ciphertext
  - No key usage, computationally infeasible to recover the plaintext
  - Examples
    - MD2-4-5, SHA, SHA-1, etc.
- Secret Key
  - Encrypt and decrypt messages by using the same Secret Key
  - Examples
    - Blowfish, DES, IDEA, RC2-4-5, Triple-DES, etc.
- Public Key
  - Encrypt and decrypt messages by using two different Keys: Public Key, Private Key (coupled together)
  - Examples
    - DSA, RSA, etc.

Secret Key vs. Public Key

- Secret Key based approaches
  - Advantages
    - Simple model
    - Provides Integrity and confidentiality
  - Challenges
    - The same secret key must be shared by all the entities involved in the data exchange
    - High risk of being compromised and does not scale well
    - Does not provide authentication and non-repudiation
- Public Key based approaches
  - Private key is only known by the owner, therefore has less risk of being compromised
  - It ensures integrity and confidentiality by encrypting with the Receiver's Public key
  - It ensures non-repudiation by encrypting with the Sender's Private key

Public Key Infrastructure (PKI)

- A PKI is an Infrastructure to support and manage Public Key-based Digital Certificates
  - Couple concepts
    - Public Key
    - Digital Certificates
- What are digital certificates and digital signatures?
  - A Digital Signature is a data item that vouches the origin and the integrity of a Message
    - The originator of a message uses a signing key (Private Key) to sign the message and send the message and its digital signature to a recipient
    - The recipient uses a verification key (Public Key) to verify the origin of the message and that it has not been tampered with while in transit

Public Key Infrastructure (PKI)

- What are digital certificates and digital signatures?
  - A Digital Signature is a data item that vouches the origin and the integrity of a Message
- Problem with Digital Signature is how to linked the "Identity" of the Signer to signature
  - Why should one trusts who the Sender claims to be?
- Digital Certificate
  - A Digital Certificate is a binding between an entity's Public Key and one or more Attributes relating its Identity
  - The entity can be a Person, an Hardware Component, or a Service
  - A Digital Certificate is issued (and signed) by someone
    - Usually the issuer is a Trusted Third Party

Digital Certificates (CA)

- Challenges to resolve with CA
  - How are Digital Certificates Issued?
  - Who is issuing them?
  - Why should one Trust the Certificate Issuer?
  - How can one check if a Certificate is valid?
  - How can one revoke a Certificate?
  - Who is revoking Certificates?

Certification Authorities (CA)

- The basic functions of the CA
  - Key Generation
  - Digital Certificate Generation
  - Certificate Issuance and Distribution
  - Revocation
  - Key Backup and Recovery System
  - Cross-Certification
- Certificate Distribution System
  - Provide a repository or a set of repositories for
    - Digital Certificates
    - Certificate Revocation Lists (CRLs)

Registration Authority (RA)

- The basic functions of the RA
  - Registration of Certificate Information
  - Face-to-Face Registration
  - Remote Registration
  - Automatic Registration
  - Revocation
- How should we do these for hardware?
  - Excellent question

Public Key Infrastructure (PKI)

- PKI-enabled Applications
- Functionality required for PKI
  - Cryptographic functionality
  - Secure storage of Personal Information
  - Digital Certificate Handling
  - Directory Access
  - Communication Facilities

Let Us Put Everything Together: Key-Based Security

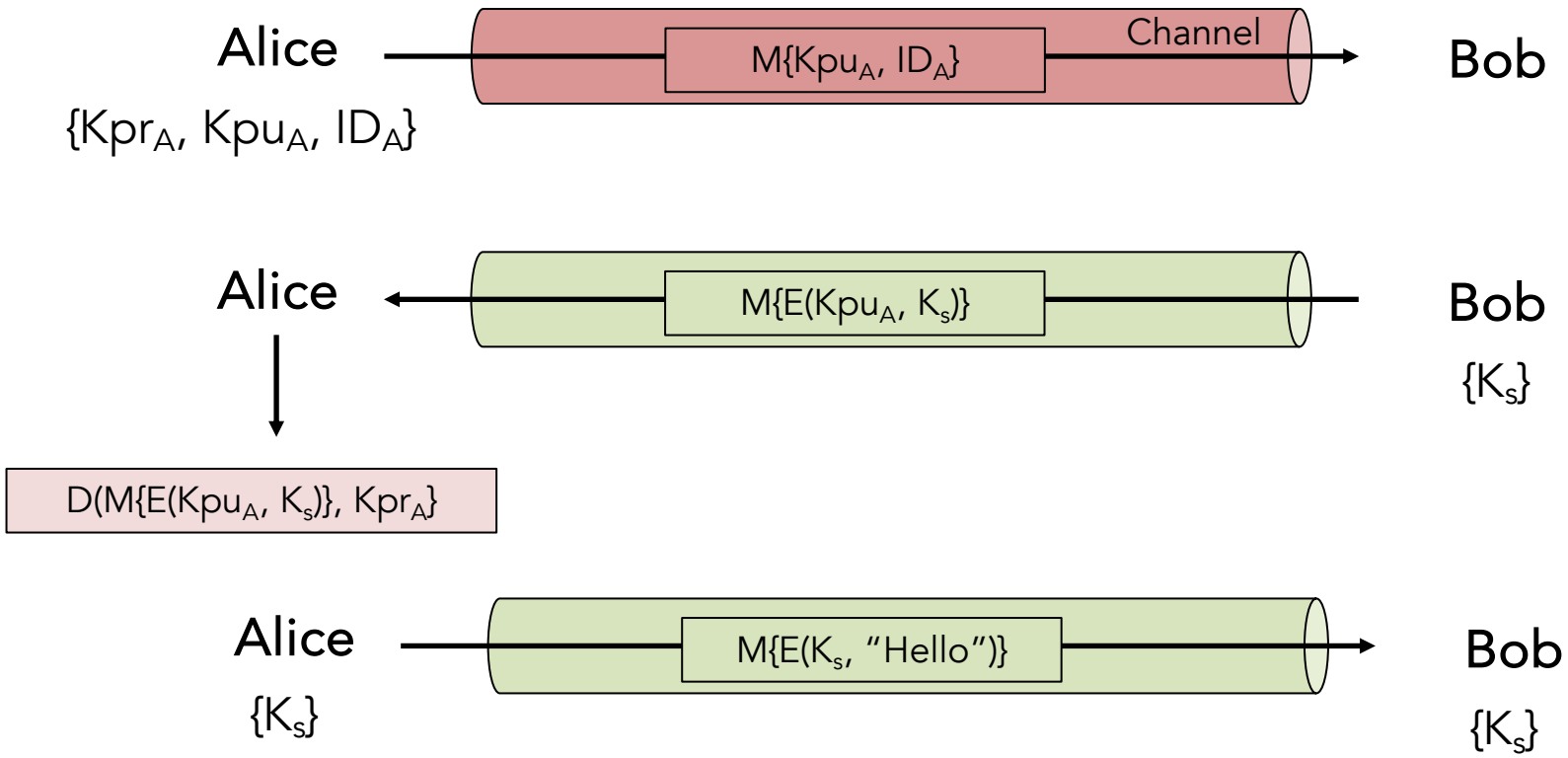
- Symmetric schemes require both parties to share a common secret key
  - The challenge is how to securely distribute the key
  - Frequent key changes can also be a challenge
- Let us say we have two participants Alice and Bob who need to digitally share a key/secret

Let Us Put Everything Together: Key-Based Security

- If Alice and Bob have secure communications with a trusted third party, they could use that to established a common key
  - But this just kicking the can down the road

Let Us Put Everything Together: Key-Based Security

- Merkle key distribution scheme

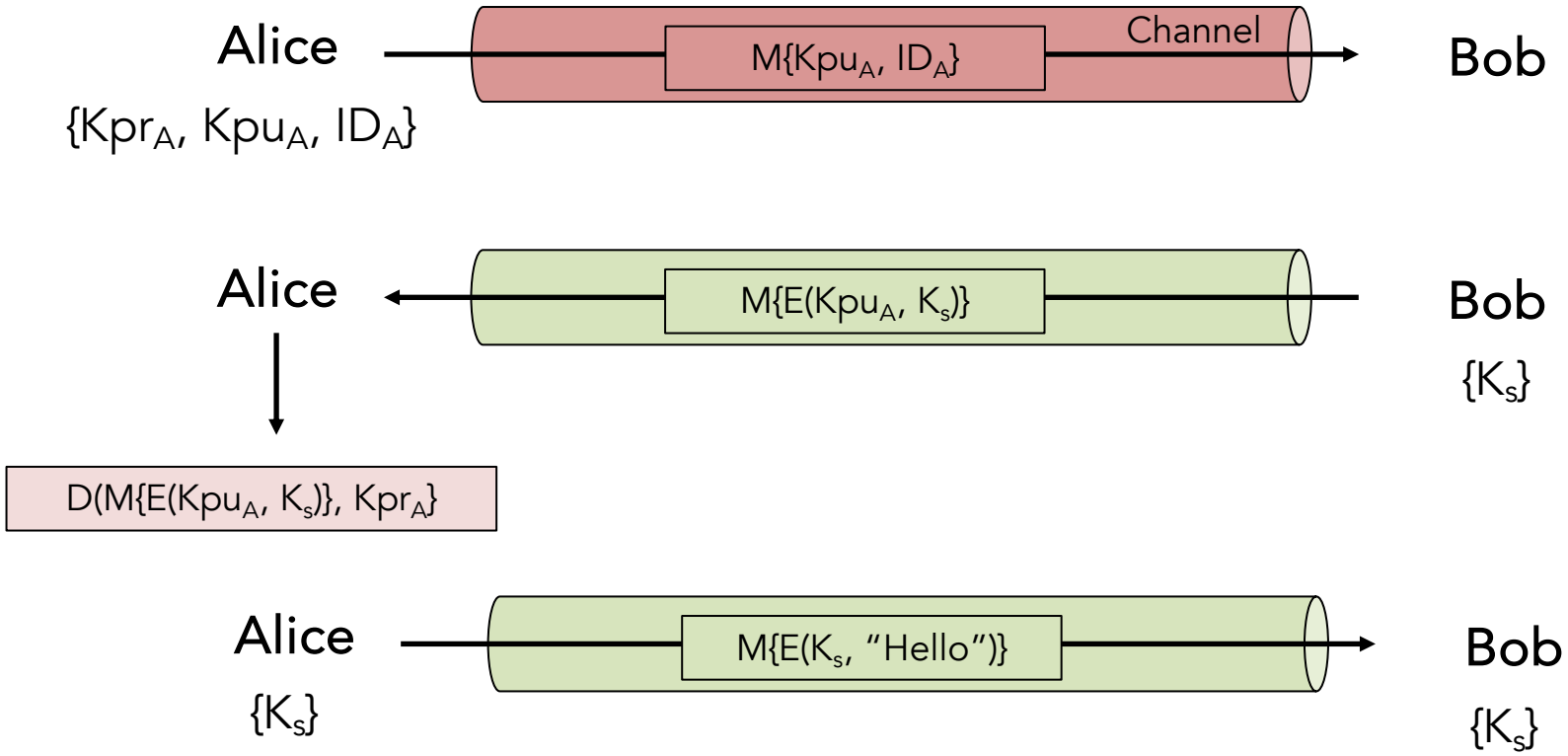


Let Us Put Everything Together: Key-Based Security

- Recall Diffie-Hellman protocol
- Diffie and Hellman formalized Merkle's proposal
  - Public information
    p is a prime number
    g is a generating element of Z_p
  - Alice's
    - Private Key : a
    - Public Key : $g^a \bmod p$
  - Bob's
    - Private Key : b
    - Public Key : $g^b \bmod p$
- Key Exchange
  - Alice obtains g^b and computes
    - $(g^b)^a = g^{ab} \bmod p = K_s$
  - Bob obtains g^a and computes
    - $(g^a)^b = g^{ab} \bmod p = K_s$
  - Alice and Bob have agreed upon key K_s

Let Us Put Everything Together: Key-Based Security

- Merkle key distribution scheme
  - Vulnerable to an active man-in-the-middle attack



Let Us Put Everything Together: Key-Based Security

- Public key based security
  - Recall the general public-private key encryption scheme
  - How should the public key distribution work?
    - Decentralized key distribution
      - Public announcement
    - Centralized key distribution
      - Publicly available directory
      - How to secure the directory itself
        - Public-key authority
        - Public-key certificates
  - Each approach has some pros and cons

Let Us Put Everything Together: Key-Based Security

- Public key based security
  - Recall the general public-private key encryption scheme
  - How should the public key distribution work?
    - Public announcement
      - Owners can distribute public keys to recipients in one-to-one, one-to-many or one to all fashions
      - There is a considerable communication overhead
      - Forgery and impersonation are problems
    - Publicly available directory
    - Public-key authority
    - Public-key certificates
  - Each approach has some pros and cons

Let Us Put Everything Together: Key-Based Security

- Public key based security
  - Recall the general public-private key encryption scheme
  - How should the public key distribution work?
    - Public announcement
    - Publicly available directory
      - Create a directory for the public keys and have owners register them
        - The directory needs to be trusted
        - The scheme needs to allow for key replacements or invalidations
        - Owner should register key securely with the directory
      - Forgery and impersonation are still problems
    - Public-key authority
    - Public-key certificates
  - Each approach has some pros and cons

Let Us Put Everything Together: Key-Based Security

- Public key based security
  - Recall the general public-private key encryption scheme
  - How should the public key distribution work?
    - Public announcement
    - Publicly available directory
      - Public-key authority
        - To implement security control over distribution of keys from directory
        - Central authority maintains a dynamic directory of public keys of all parties
        - The parties know the authority's public key
      - Public-key certificates
  - Each approach has some pros and cons

Public Key Authority

- What is initially known by the parties

Trusted Third Party

Public Key Authority

$\{Kpr_{Au}, Kpu_{Au}, Kpu_{A}, Kpu_{B}\}$

Alice

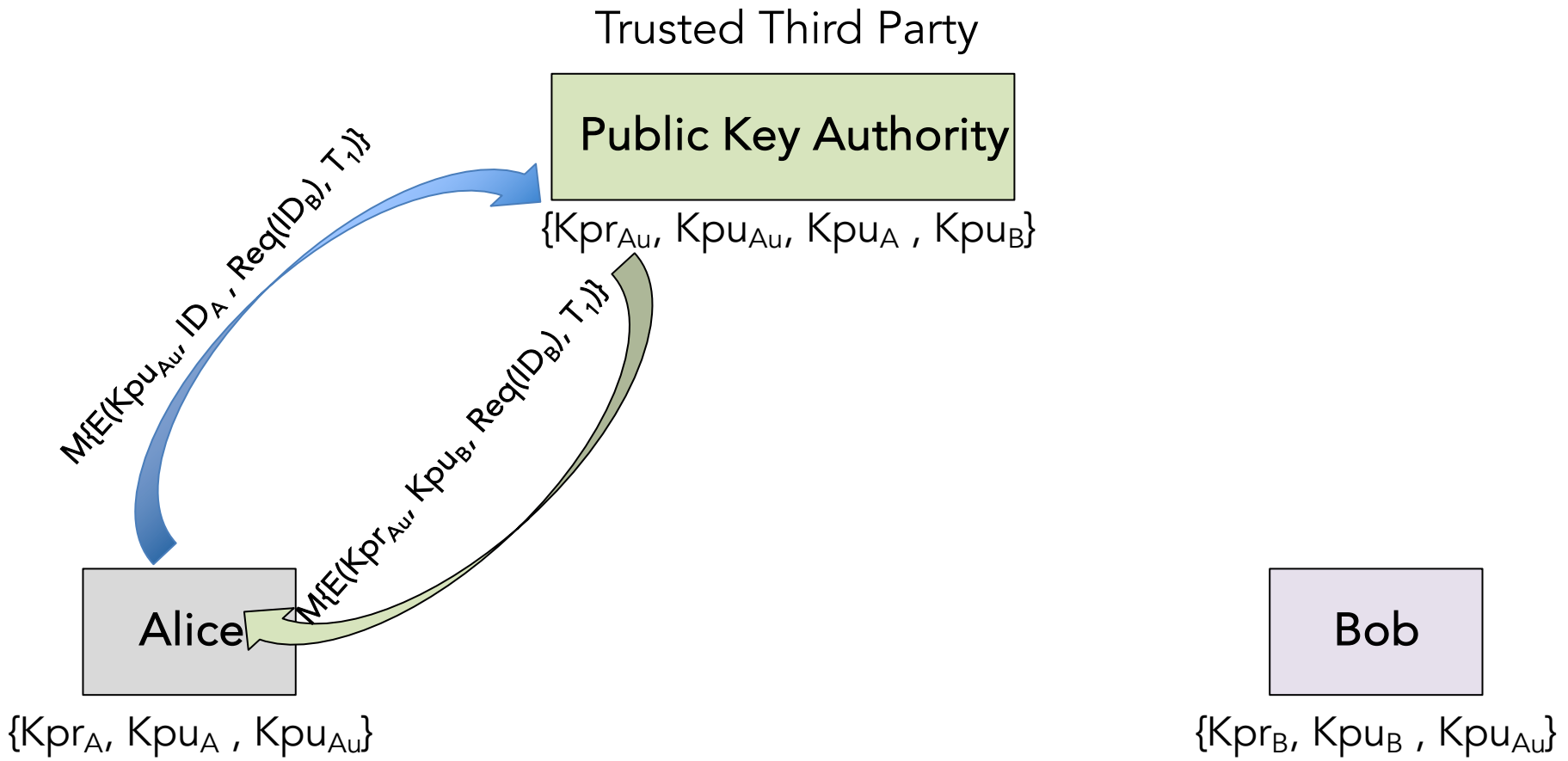
$\{Kpr_{A}, Kpu_{A}, Kpu_{Au}\}$

Bob

$\{Kpr_{B}, Kpu_{B}, Kpu_{Au}\}$

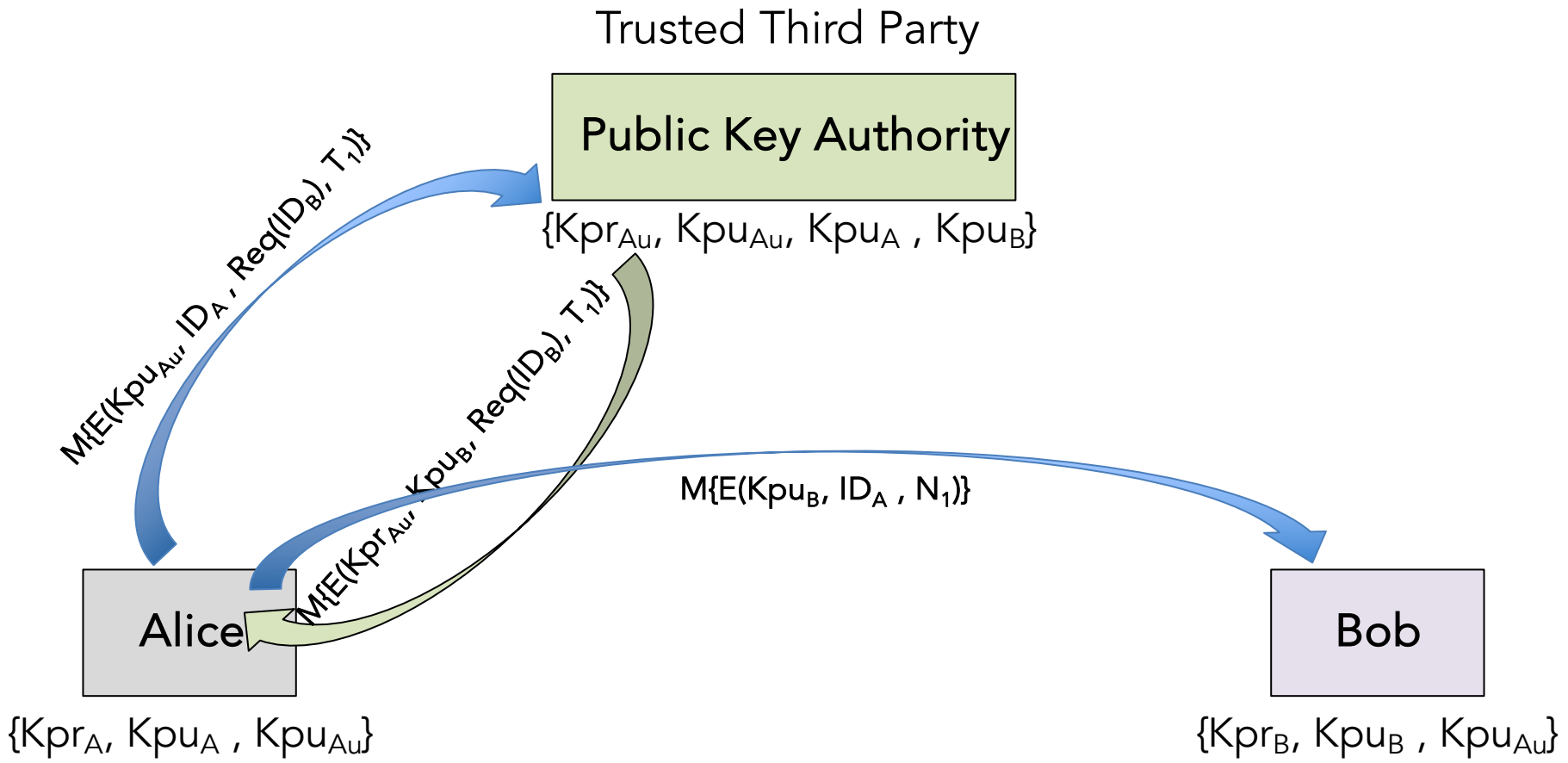
Public Key Authority

- What is initially known by the parties



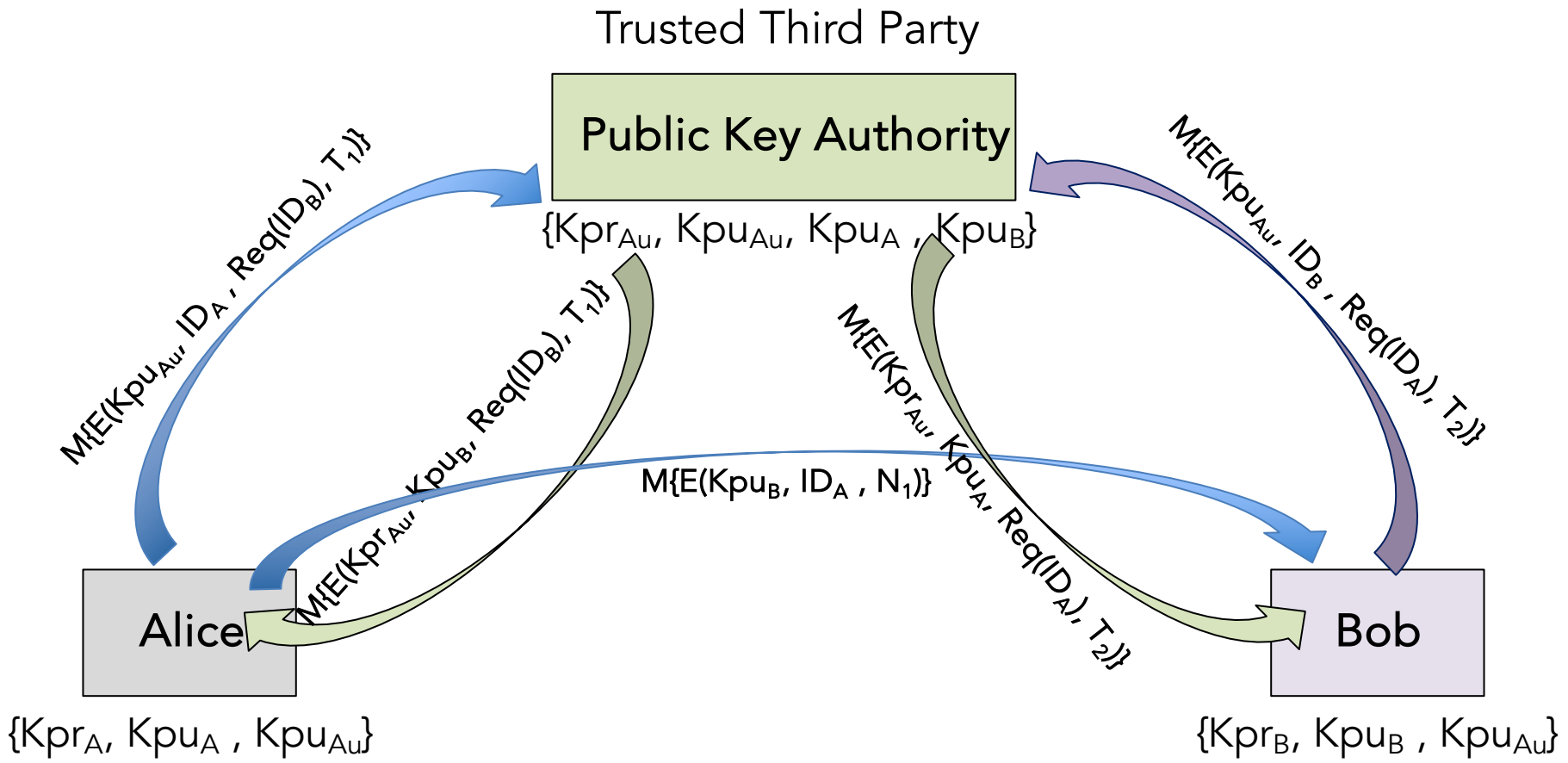
Public Key Authority

- What is initially known by the parties



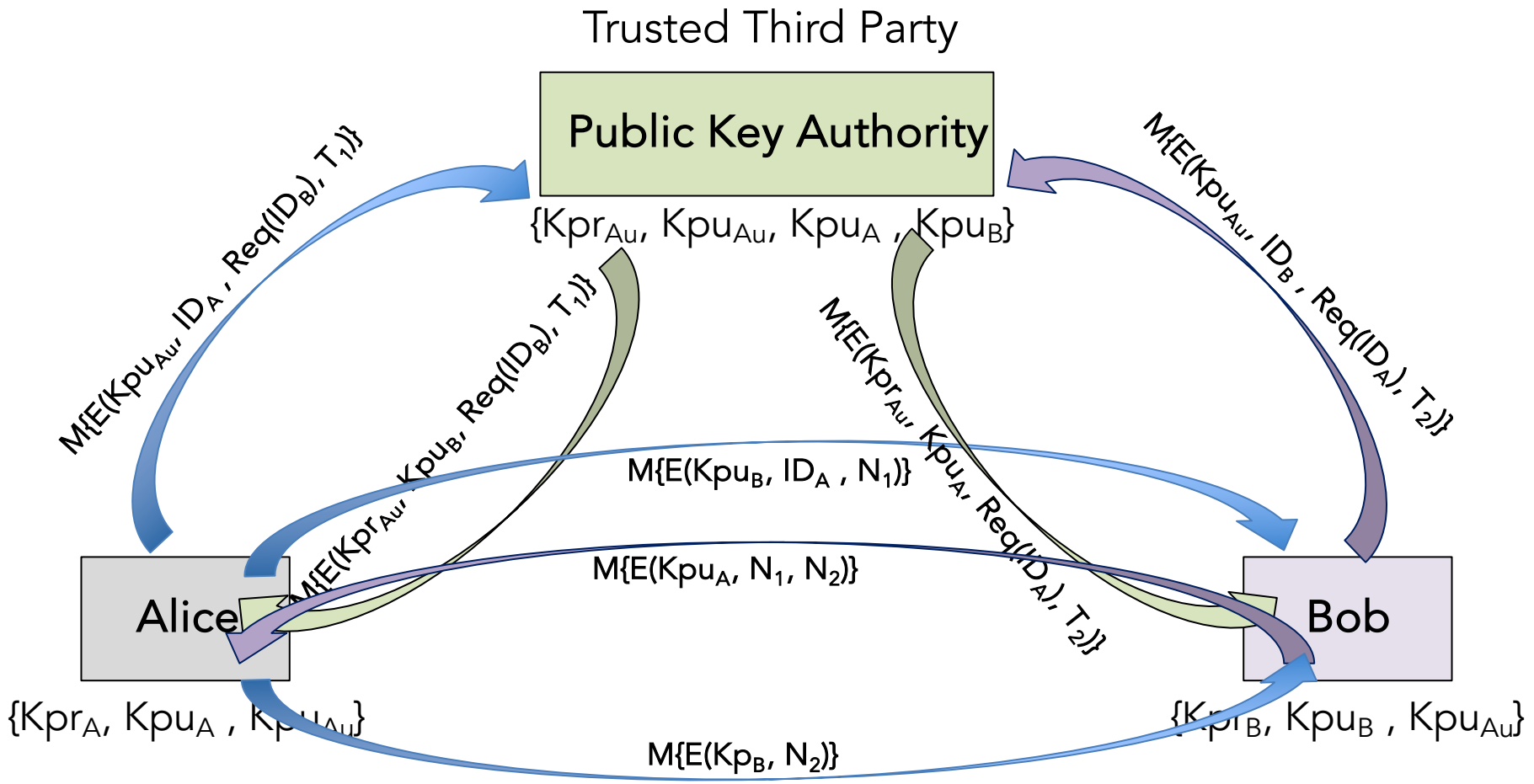
Public Key Authority

- What is initially known by the parties



Public Key Authority

- What is initially known by the parties



Public Key Certificates

- Certificates allow key exchange without real-time access to public-key authority
  - A certificate binds the identity (ID) of the owner and its public key
    - Period of validity
    - Rights of use
  - Signed by a trusted third party or public key certificate authority (CA)
    - $\text{Alice}_{\text{Certificate}} = f_{\text{binding\_function}}(\text{ID}_A, \text{Expiration}, \text{Kpu}_A, \text{use}_{A1}, \text{signature})$
  - Key: Can be verified by anyone who knows CA's public key

Public Key Certificates

- Certificates allow key exchange without real-time access to public-key authority
  - Alice needs Bob's public key
    - Alice takes Bob's certificate
    - She apply certification authority (CA)'s public key to Bob's certificate to get Bob's public key
- Lower number of message exchanges
  - Compared to the public key authority protocol
  - But there are a number of issues with both
    - How certificate requests from users are processed
    - How users' identity is validated
    - Algorithm used to generate and sign the user's certificate Efficient revocation of certificates

Key Hierarchy

- Public key encryption algorithms are slow
  - Inefficient
- Symmetric key encryption algorithms are many orders of magnitude faster than public key encryption algorithms
- Typically there is a hierarchy of keys
  - Session key or temporary key used for encryption of data between users for one logical session then discarded
    - Symmetric key
  - Master key used to negotiate session keys
    - Public key

Next Class

- Information Channels, Covert Channels, & Side Channels