Cryptography Engineering

- Lecture 2 (Oct 29, 2025)

- Today's notes:
  - Review the Rust example code
  - DH handshake
  - Man-in-the-middle attacks

- Today's coding tasks:
  - Derive a secret key for AEAD via DH handshake
  - Man-in-the-Middle attacks on DHKE

Code Review

- Some useful notes:
  - Modular design
  - Reusability

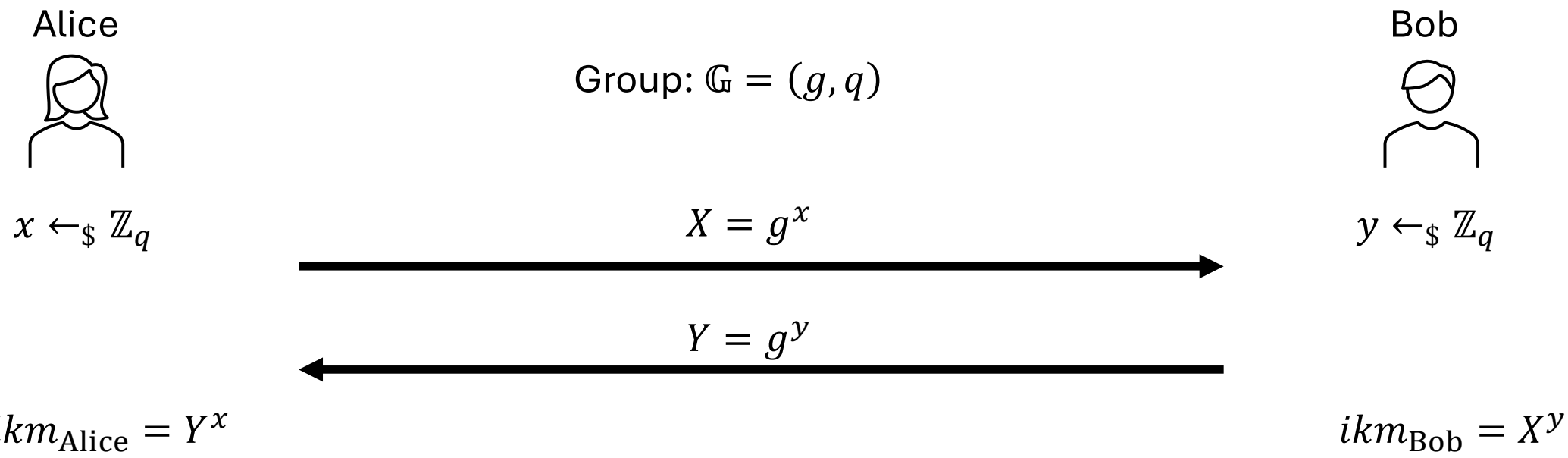
Diffie-Hellman Key Exchange

- Group (Mathematics): $(\mathbb{G}, +)$
  - **Associativity:** $a, b, c \in \mathbb{G} \implies (a + b) + c = a + (b + c)$
  - **Identity:** $\exists e \in \mathbb{G}$ s.t. $\forall a \in \mathbb{G} \implies e + a = a$
  - **Inverse:** $\forall a \in \mathbb{G}, \exists b \in \mathbb{G}$ s.t. $a + b = e$

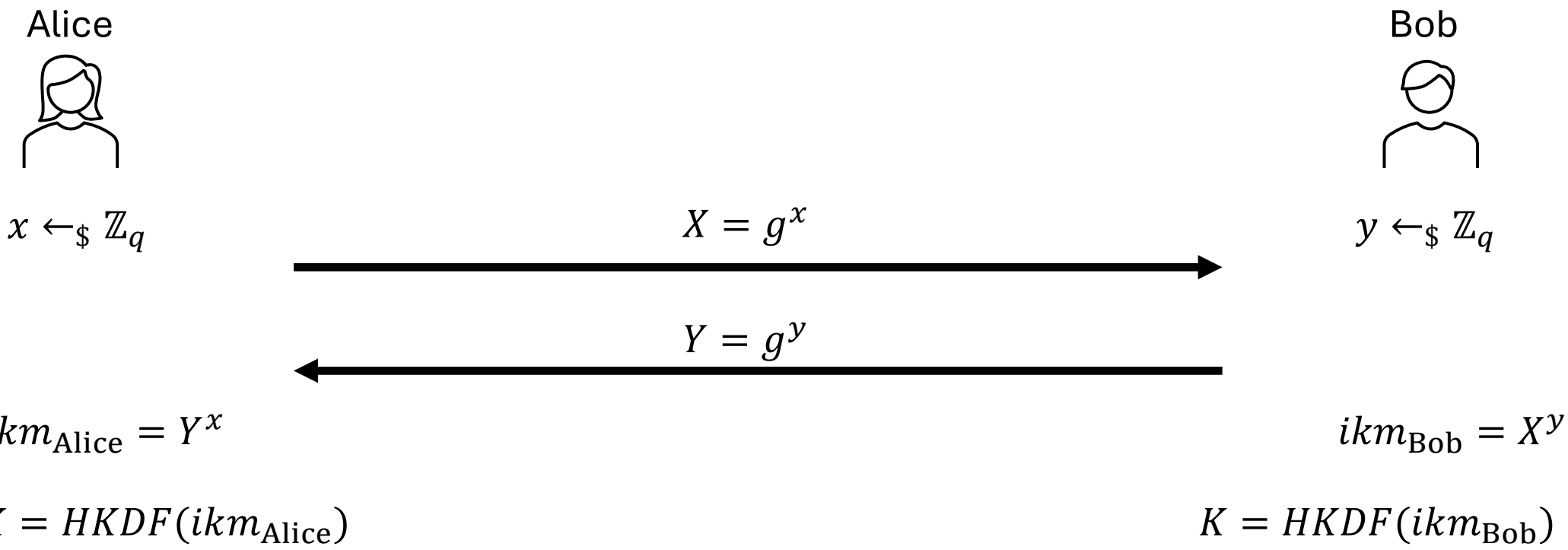
  - Example 1: $(\mathbb{R}, +)$ is a group, but $(\mathbb{Z}, \times)$ is not
  - Example 2: $(\mathbb{R}, \times)$ is **not** a group (why?), but $(\mathbb{R}^*, \times)$ is a group
  - Quick question: $(\{1,2,3, \dots, q\}, \times \ mod\ q)$ is a group if and only if q is _____

- In cryptography, we usually use **finite** groups to build cryptosystems.
  - Generator and order
  - Elliptic Curve Groups

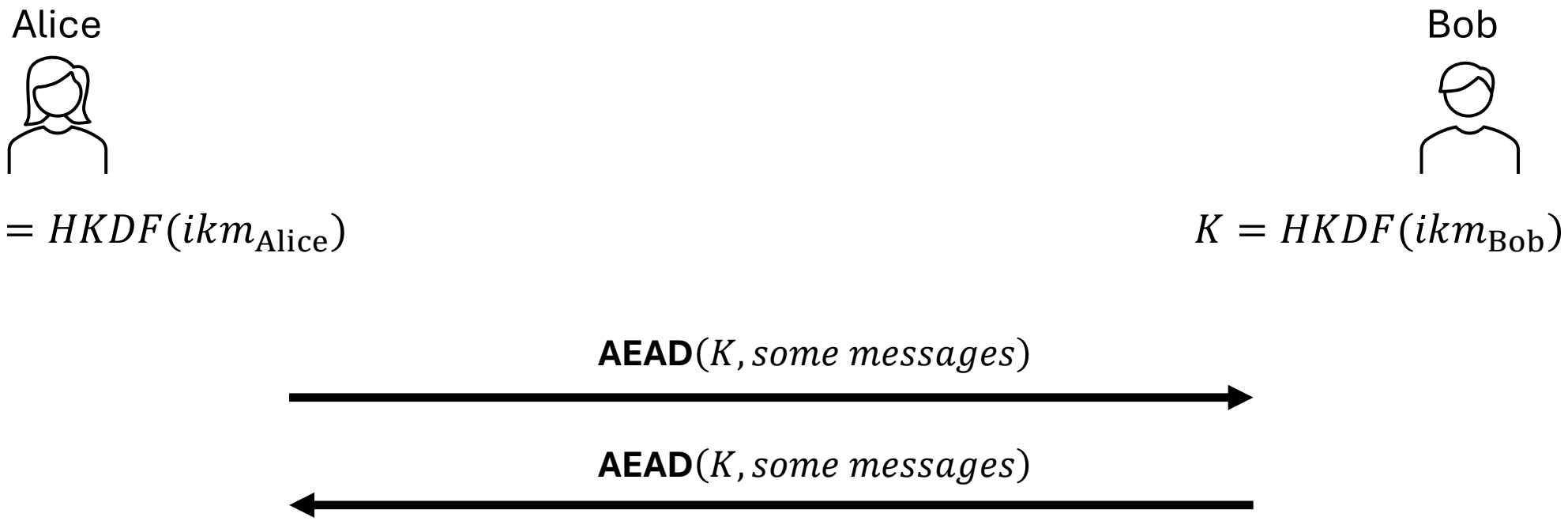
Diffie-Hellman Key Exchange



Diffie-Hellman Key Exchange

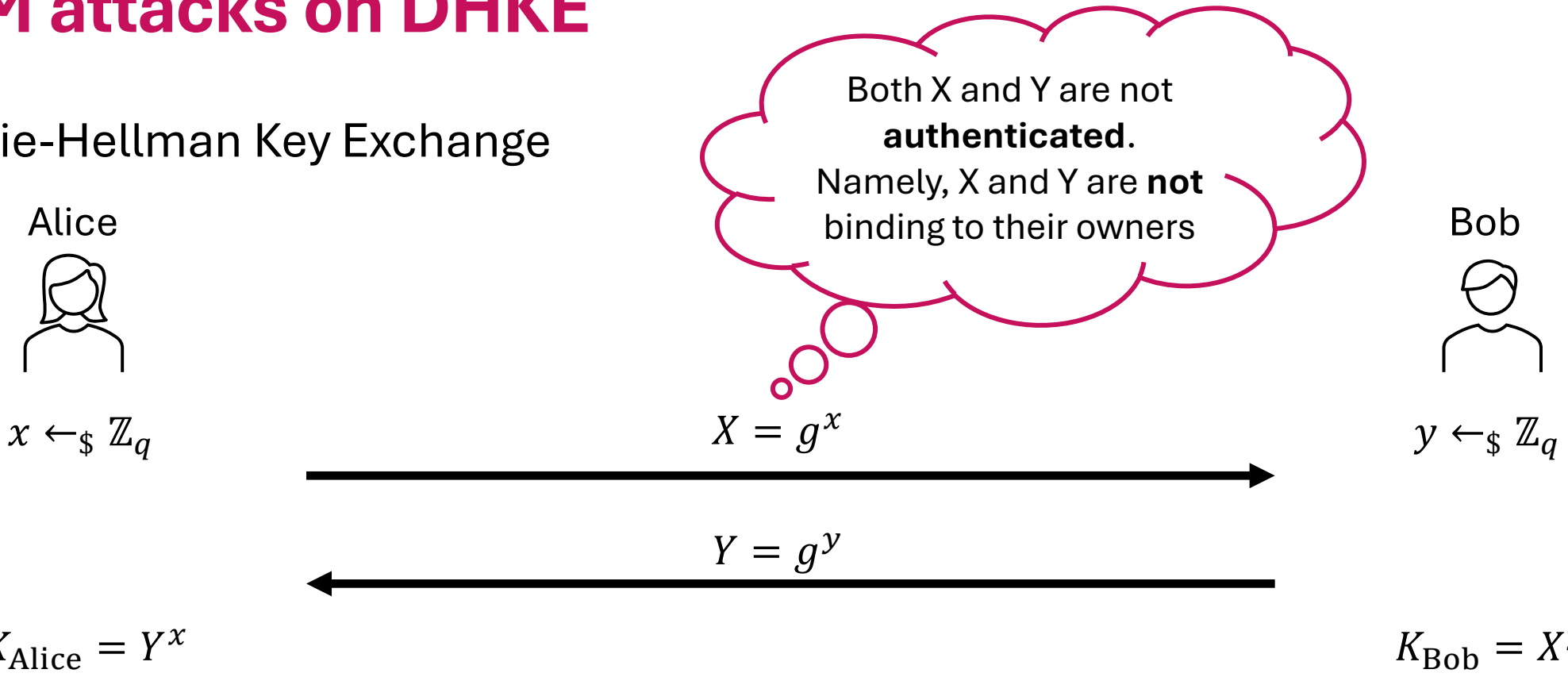


Diffie-Hellman Key Exchange



MitM attacks on DHKE

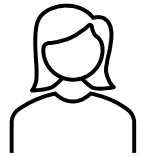
- Diffie-Hellman Key Exchange



MitM attacks on DHKE

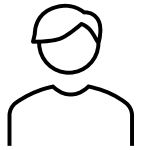
- Diffie-Hellman Key Exchange

Alice



Adversary

Bob

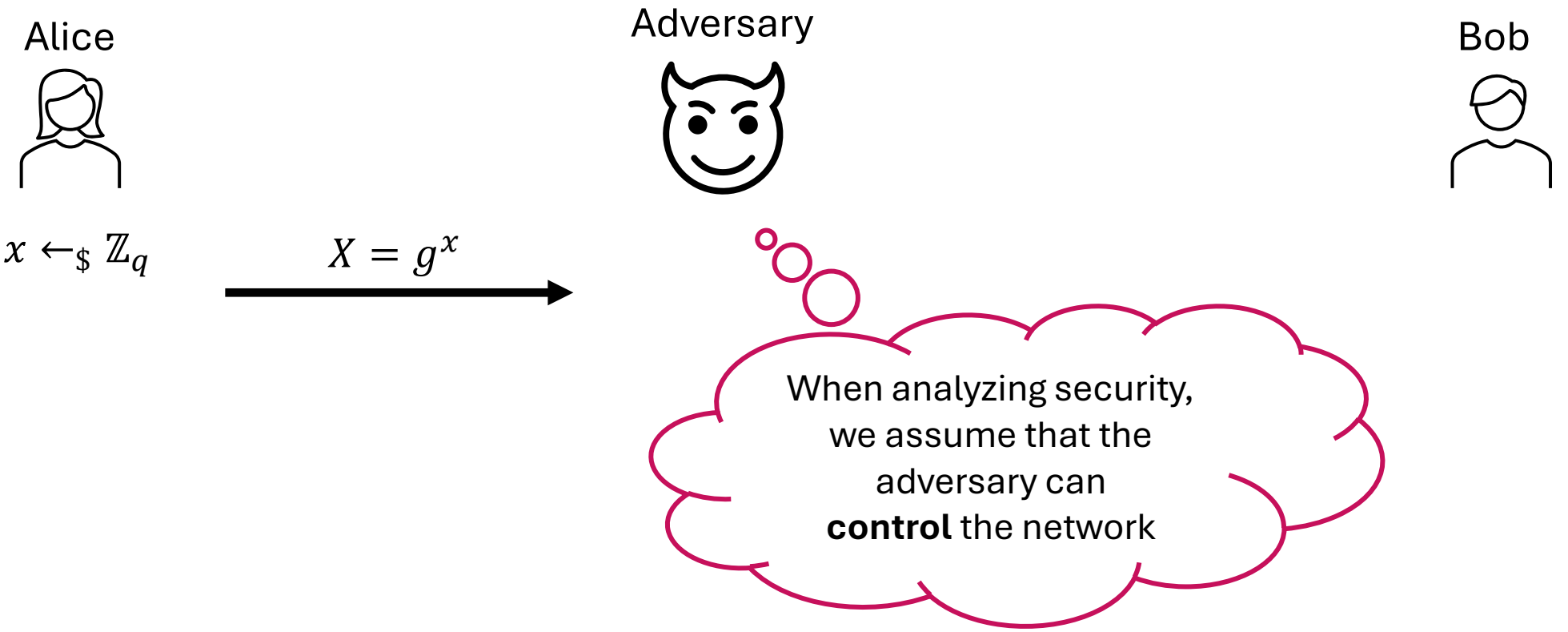


$x \leftarrow_{\$} \mathbb{Z}_q$

$X = g^x$

MitM attacks on DHKE

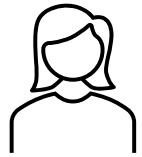
- Diffie-Hellman Key Exchange



MitM attacks on DHKE

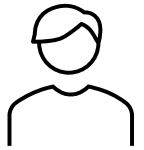
- Diffie-Hellman Key Exchange

Alice



Adversary

Bob

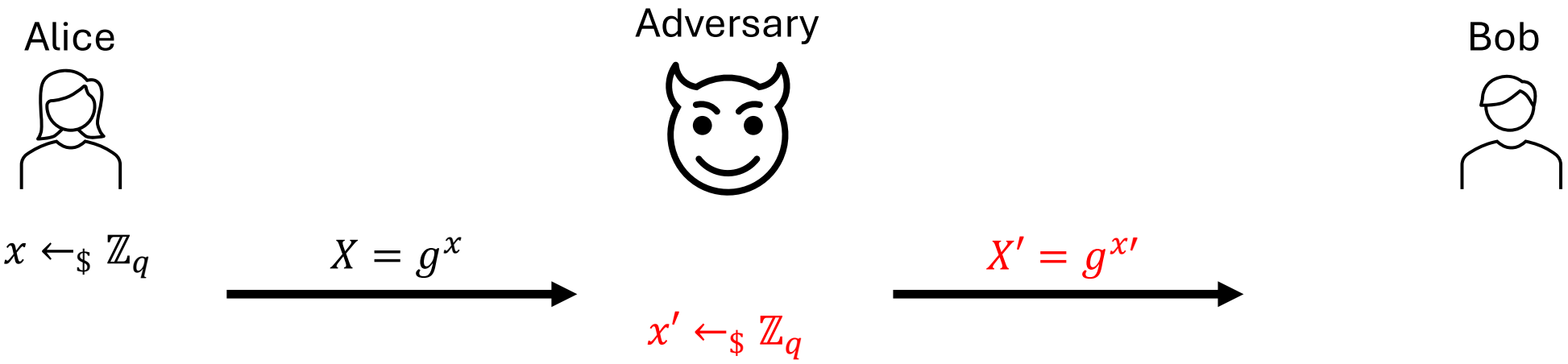


$x \leftarrow_{\$} \mathbb{Z}_q$

$X = g^x$

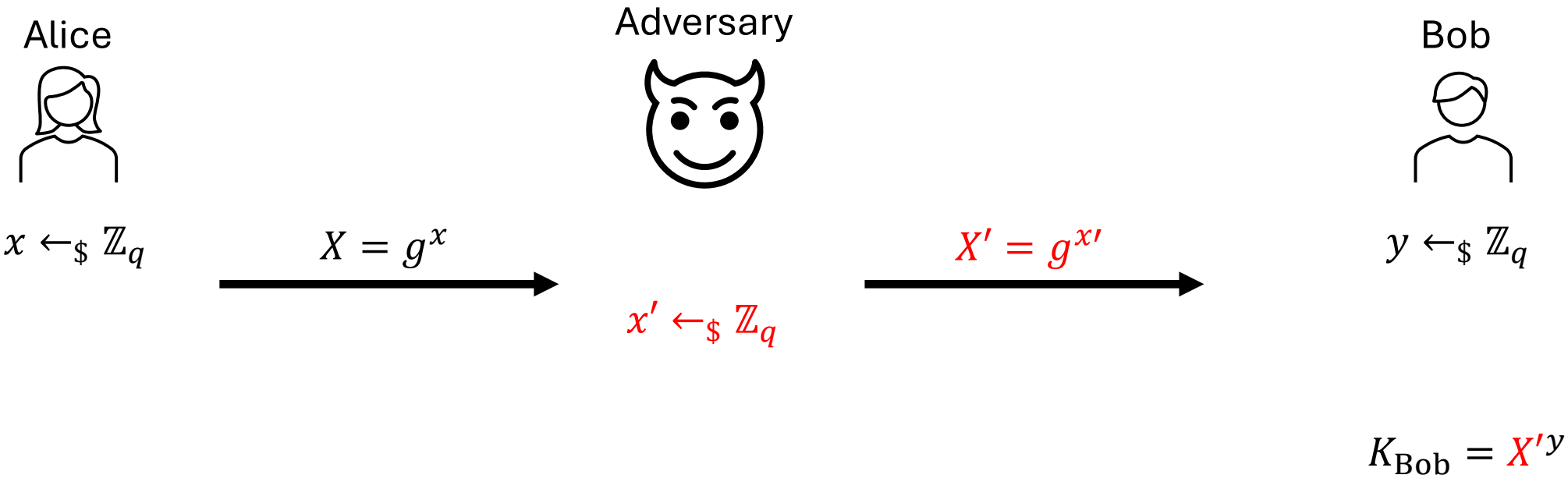
MitM attacks on DHKE

- Diffie-Hellman Key Exchange



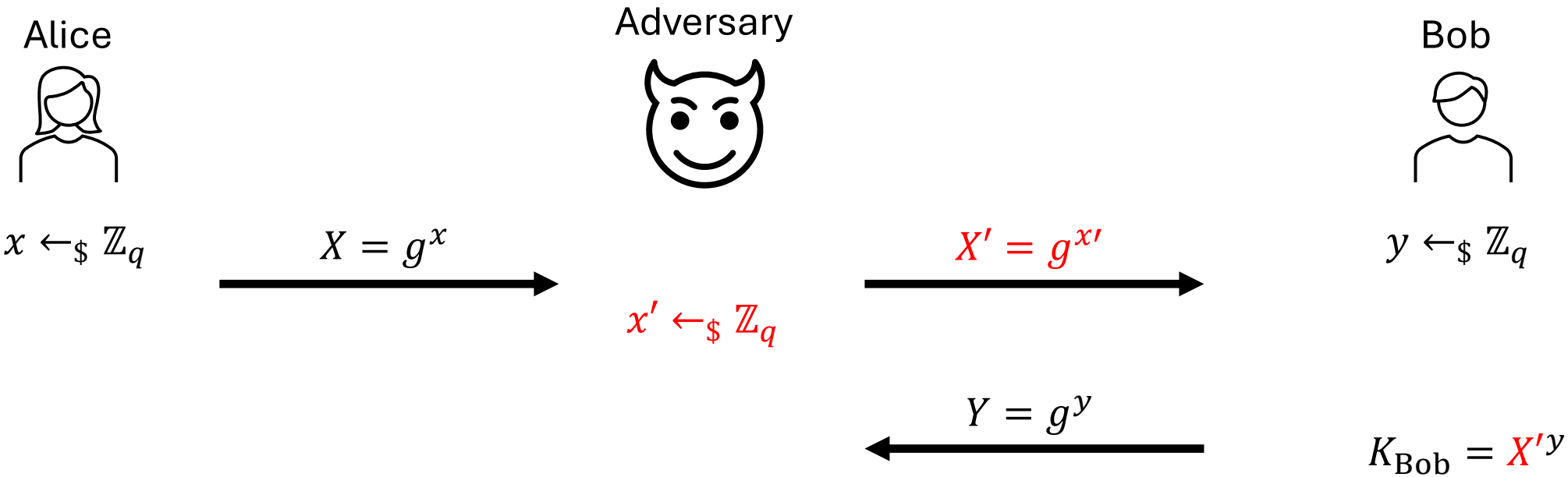
MitM attacks on DHKE

- Diffie-Hellman Key Exchange



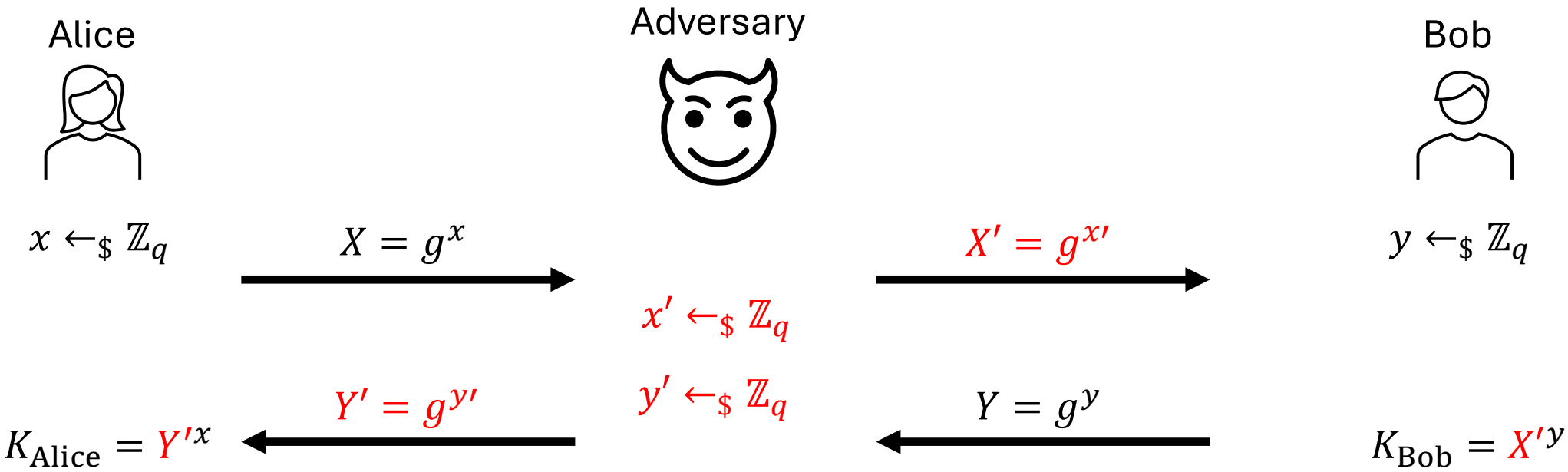
MitM attacks on DHKE

- Diffie-Hellman Key Exchange



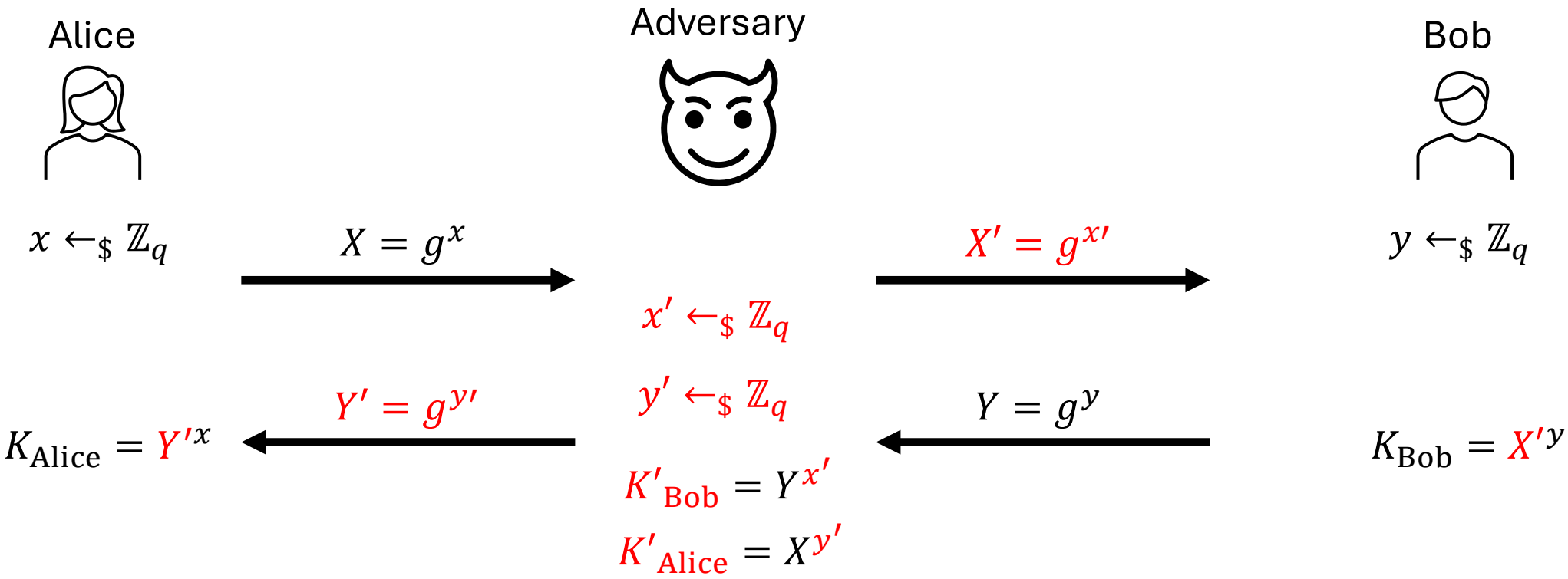
MitM attacks on DHKE

- Diffie-Hellman Key Exchange



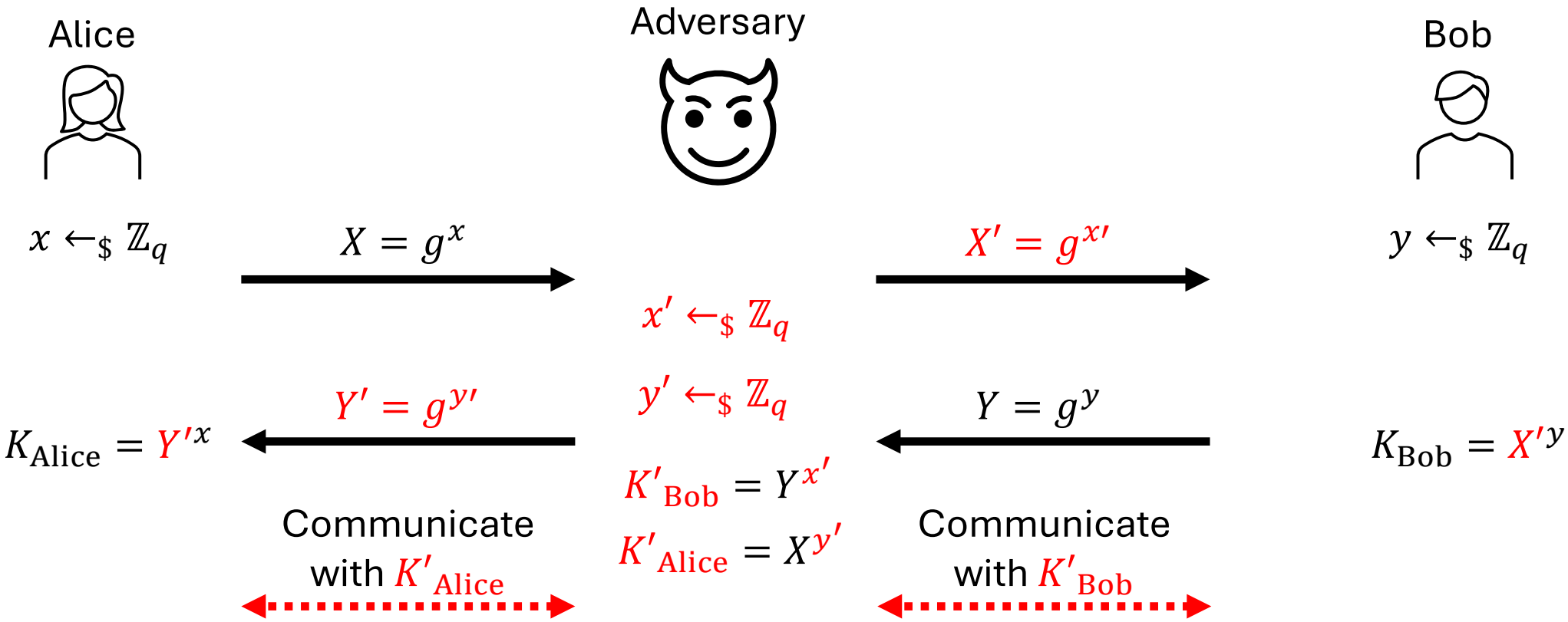
MitM attacks on DHKE

- Diffie-Hellman Key Exchange



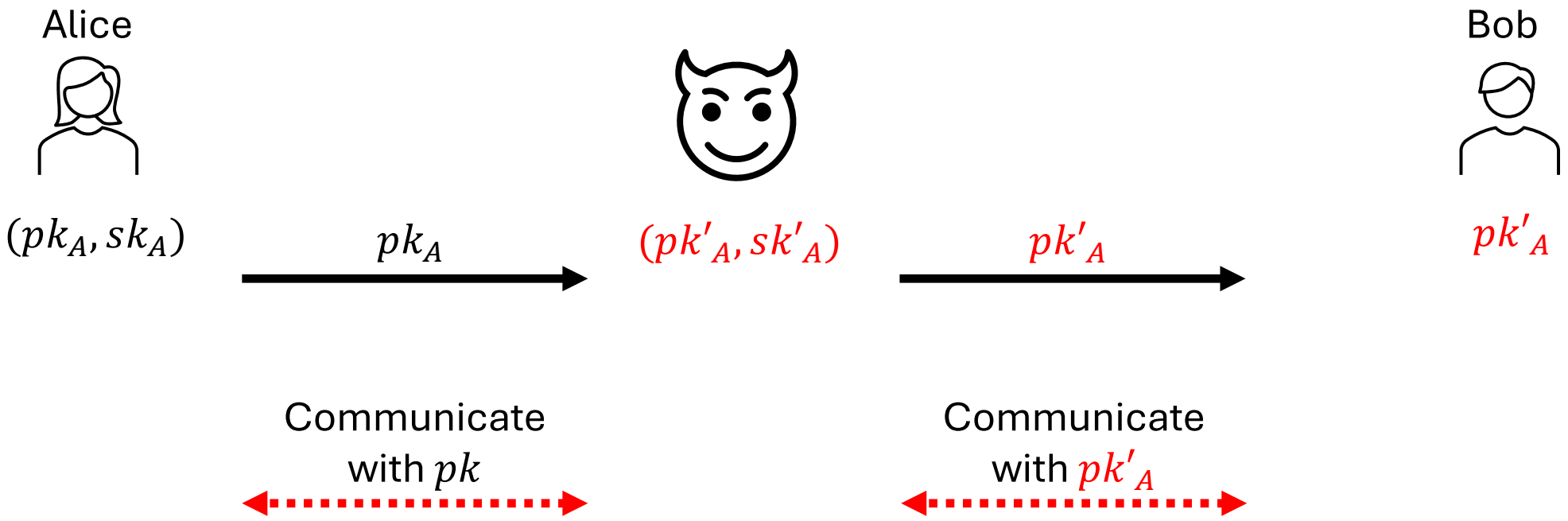
MitM attacks on DHKE

- Diffie-Hellman Key Exchange



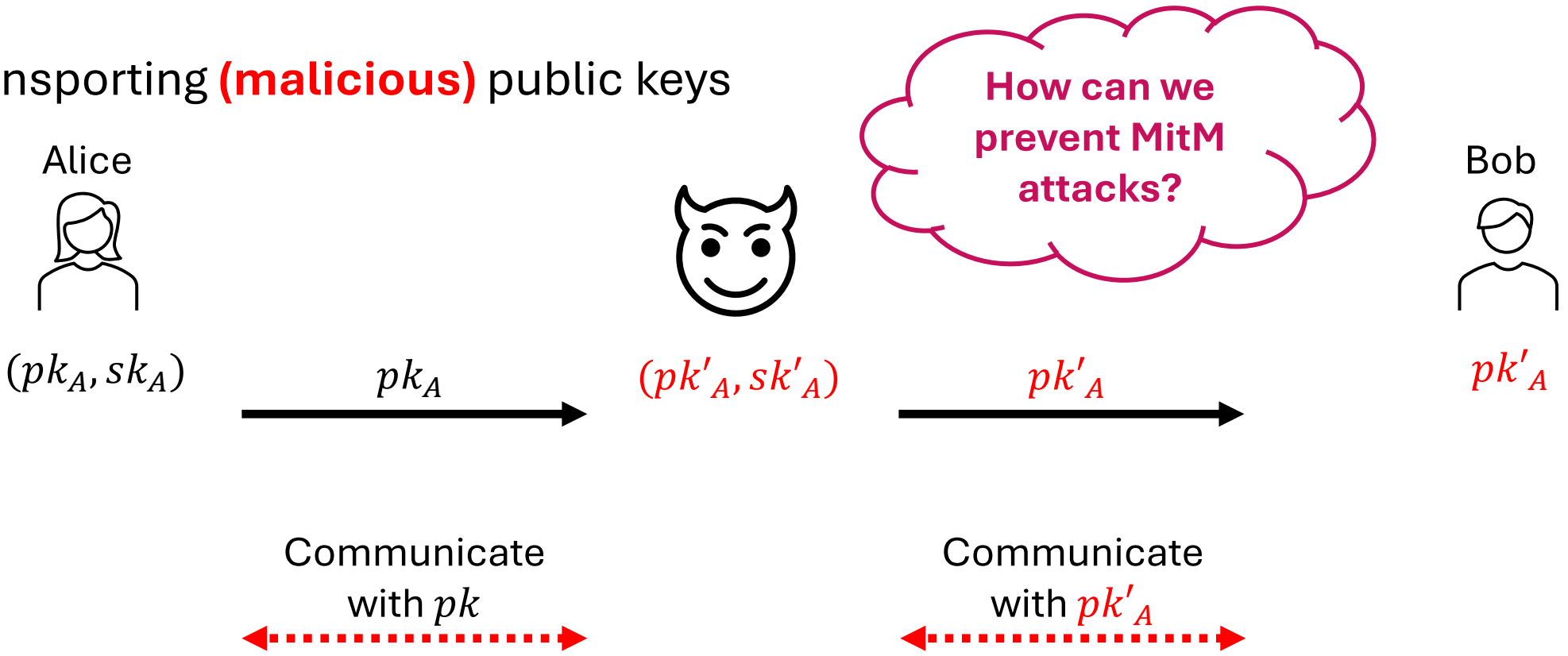
MitM attacks (in General)

- Transporting **(malicious)** public keys



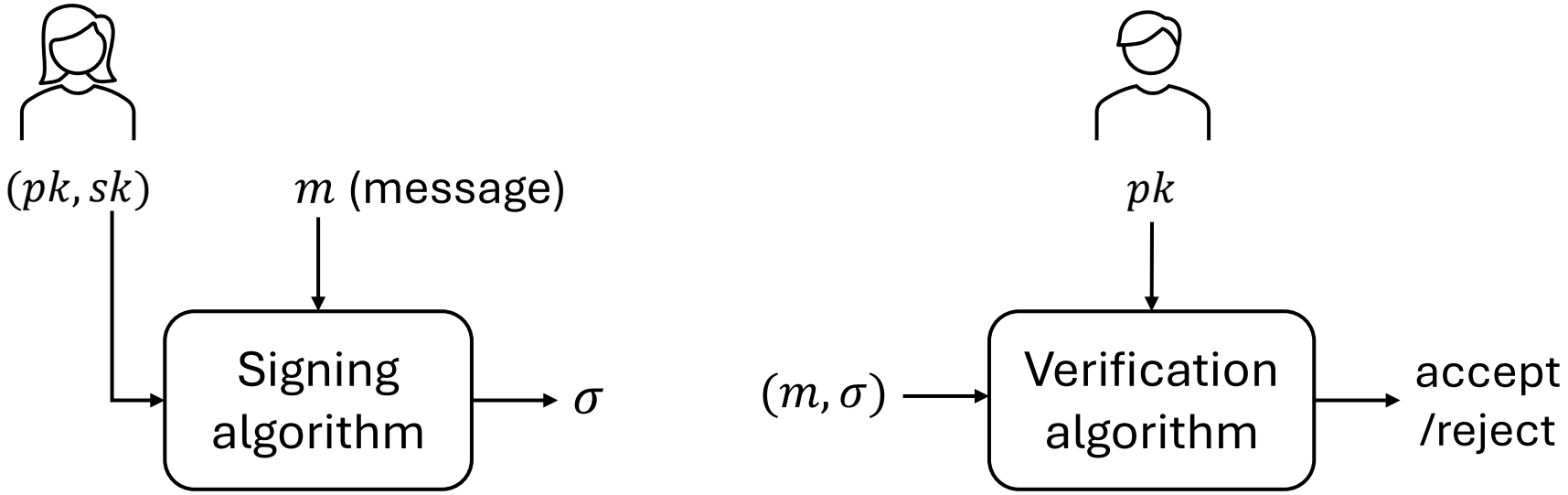
MitM attacks (in General)

- Transporting **(malicious)** public keys



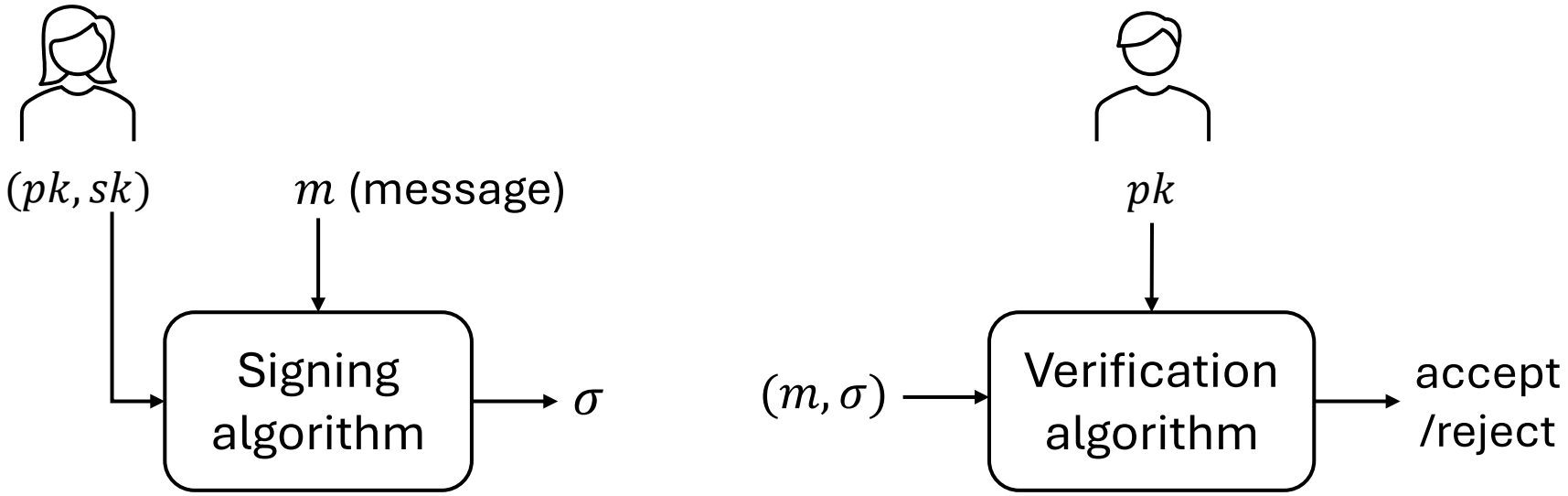
Digital Signature

- Signature Schemes



Digital Signature

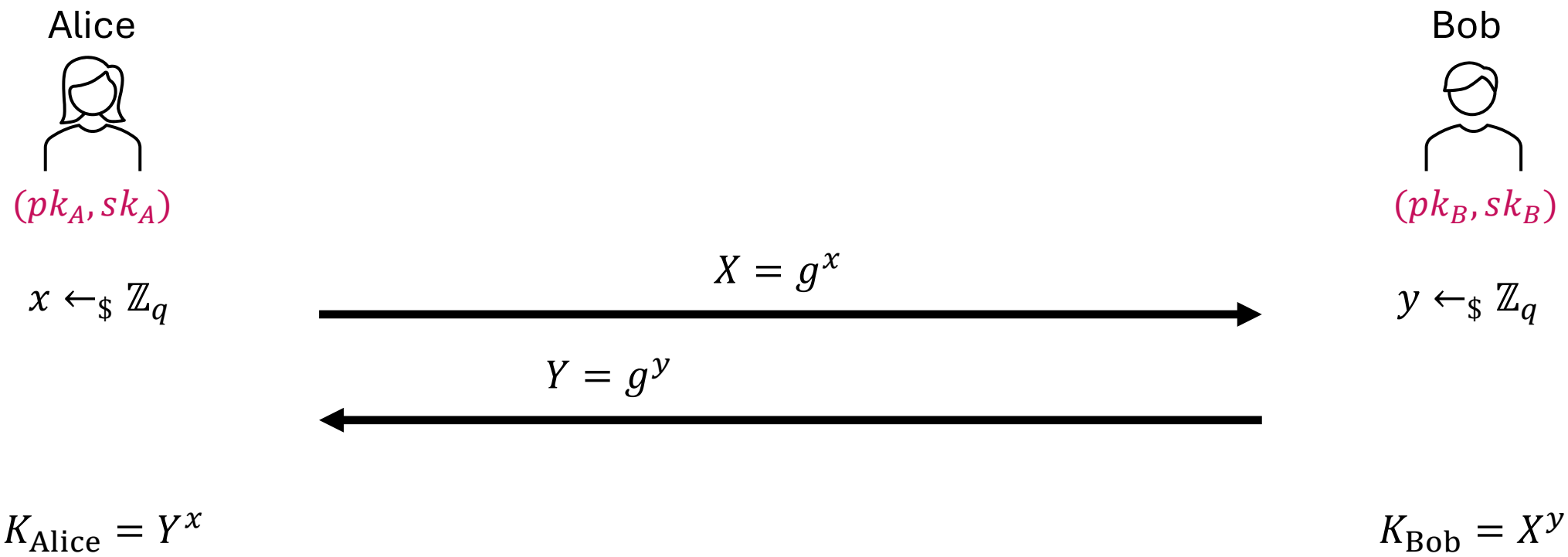
- Signature Schemes



- Security: **Unforgeability**
  - **Unable to forge** a valid signature on any message without sk

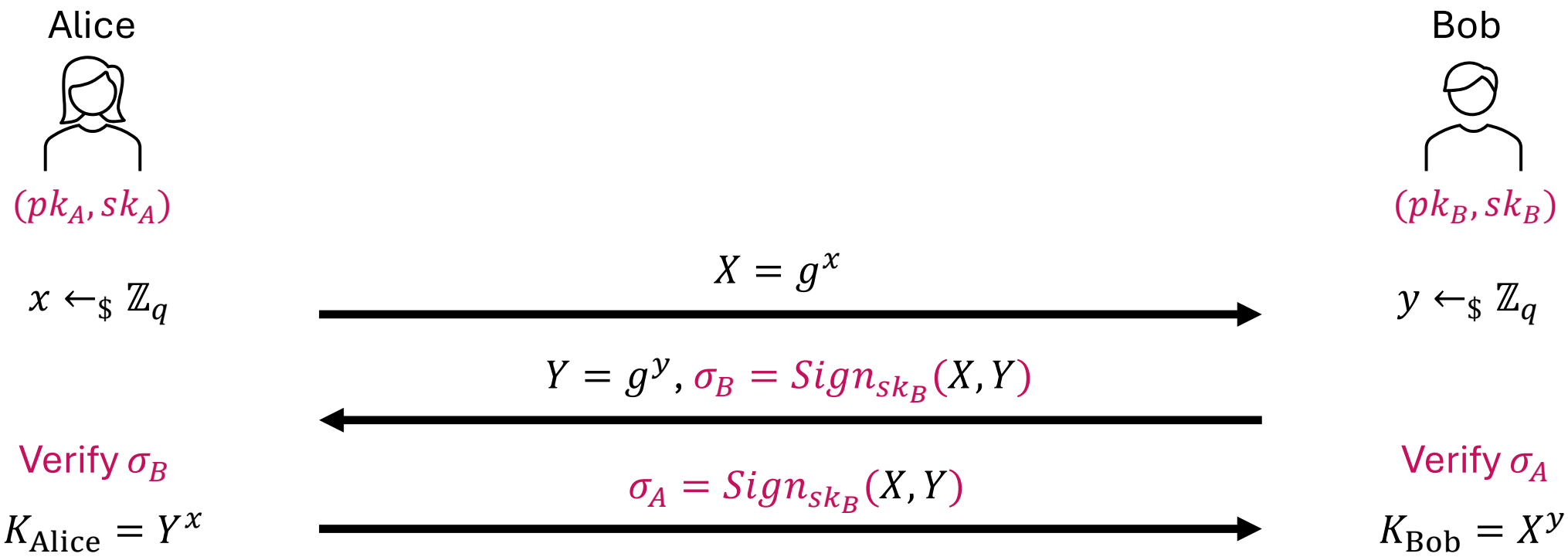
Signed DH Key Exchange (Next Lecture)

- Use signature to avoid MitM attacks on DHKE:



Signed DH Key Exchange (Next Lecture)

- Use signature to avoid MitM attacks on DHKE:



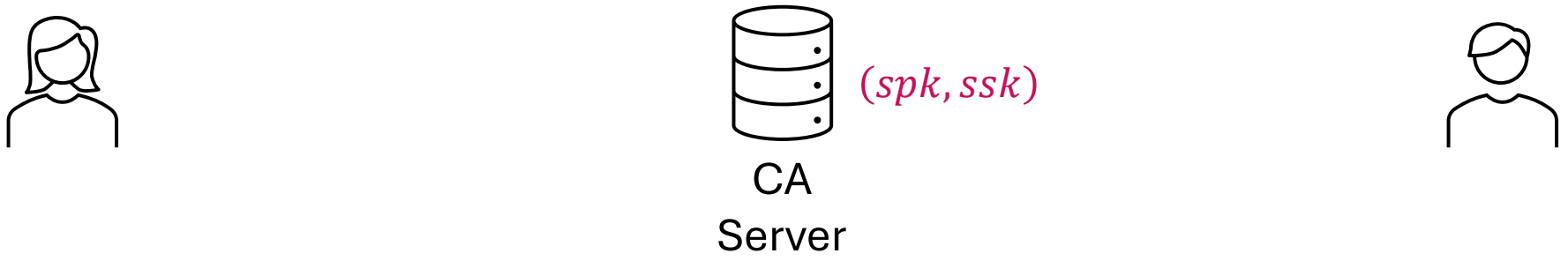
Digital Signature

- Other standard properties of Digital Signature:
  - Authentication // Verify the identity…
  - Publicly verifiable // Everyone with pk can verify the signature…
  - Non-repudiation // A party cannot deny having sent or signed a message…
  - …

- One of the most important applications: **Digital Certificate**

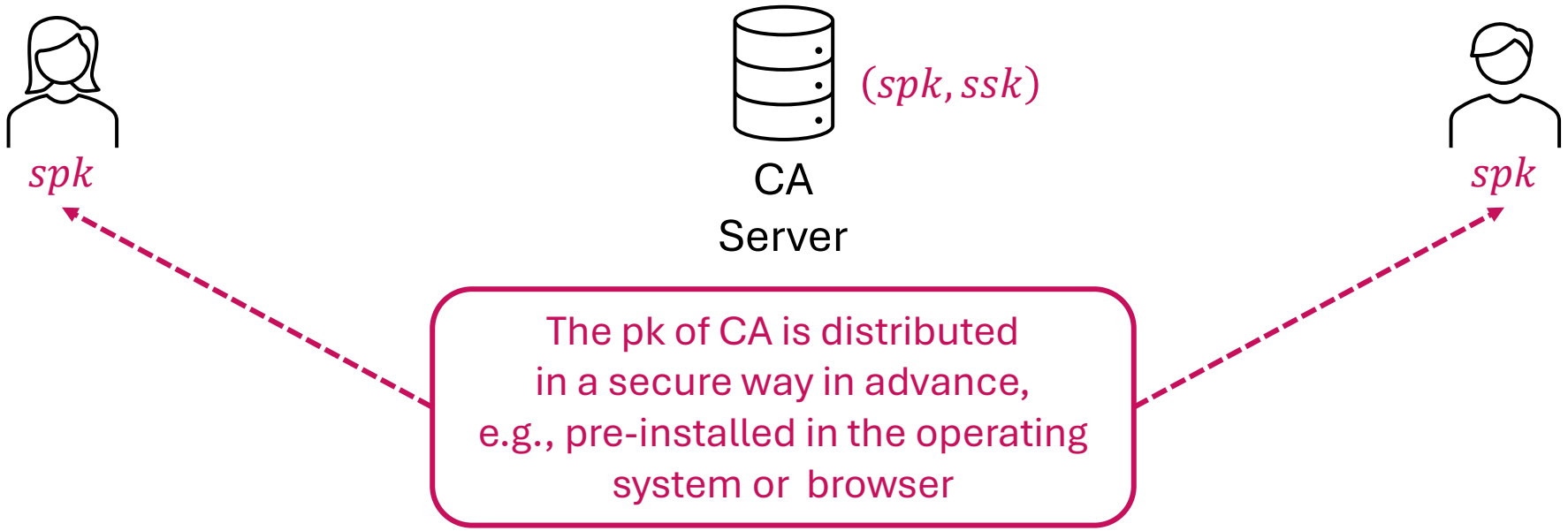
Digital Certificate

- Certificate: A signature generated by a trusted party (In short)
  - Verifies an ID and binds it to a public key
  - Securely distribute public keys
  - Issued by **CA** (Certificate Authority)



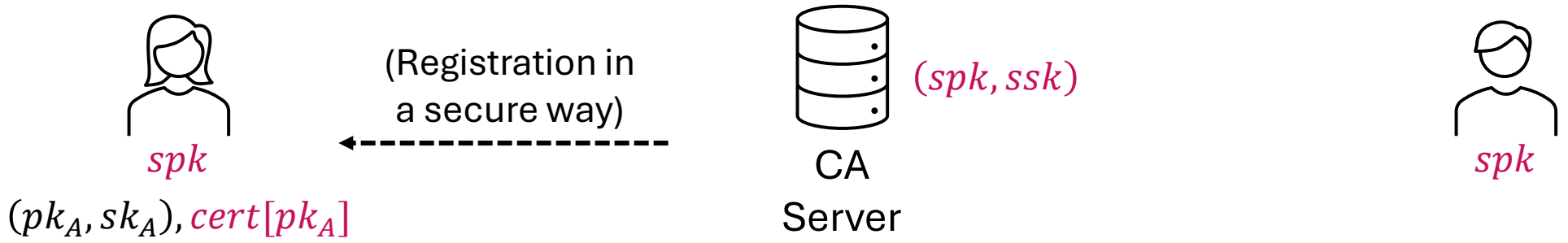
Digital Certificate

- Certificate: A signature generated by a trusted party (In short)
  - Verifies an ID and binds it to a public key
  - Securely distribute public keys
  - Issued by **CA** (Certificate Authority)



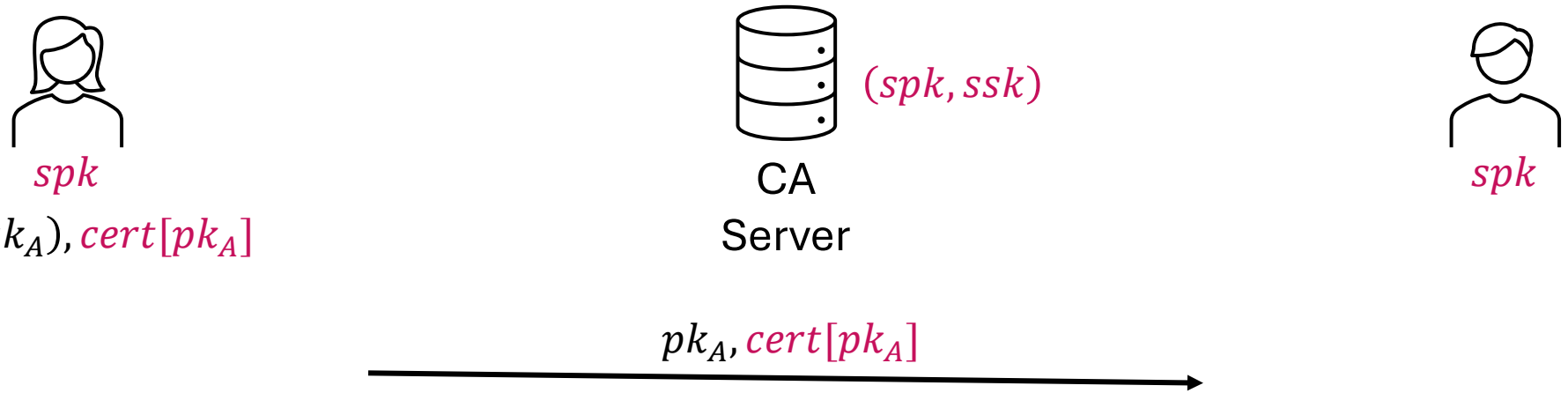
Digital Certificate

- Certificate: A signature generated by a trusted party (In short)
  - Verifies an ID and binds it to a public key
  - Securely distribute public keys
  - Issued by **CA** (Certificate Authority)



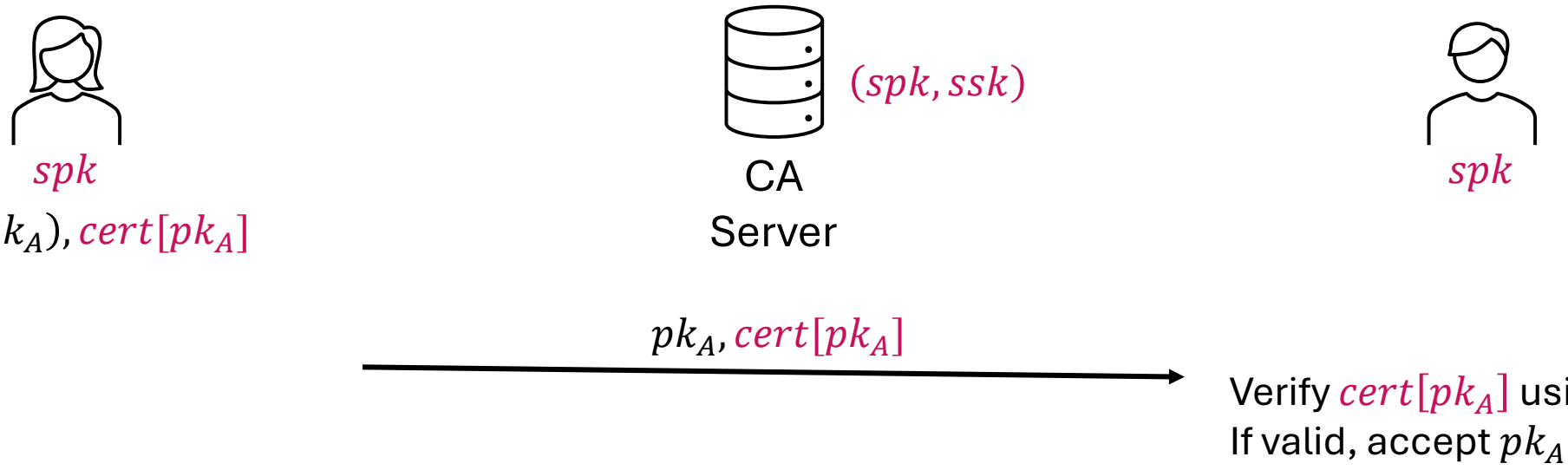
Digital Certificate

- Certificate: A signature generated by a trusted party (In short)
  - Verifies an ID and binds it to a public key
  - Securely distribute public keys
  - Issued by **CA** (Certificate Authority)



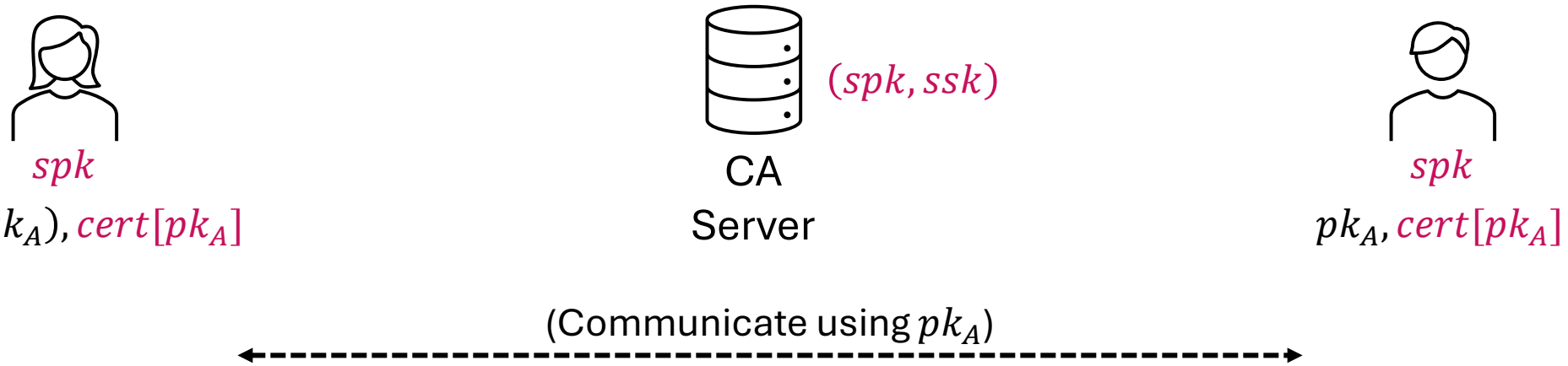
Digital Certificate

- Certificate: A signature generated by a trusted party (In short)
  - Verifies an ID and binds it to a public key
  - Securely distribute public keys
  - Issued by **CA** (Certificate Authority)



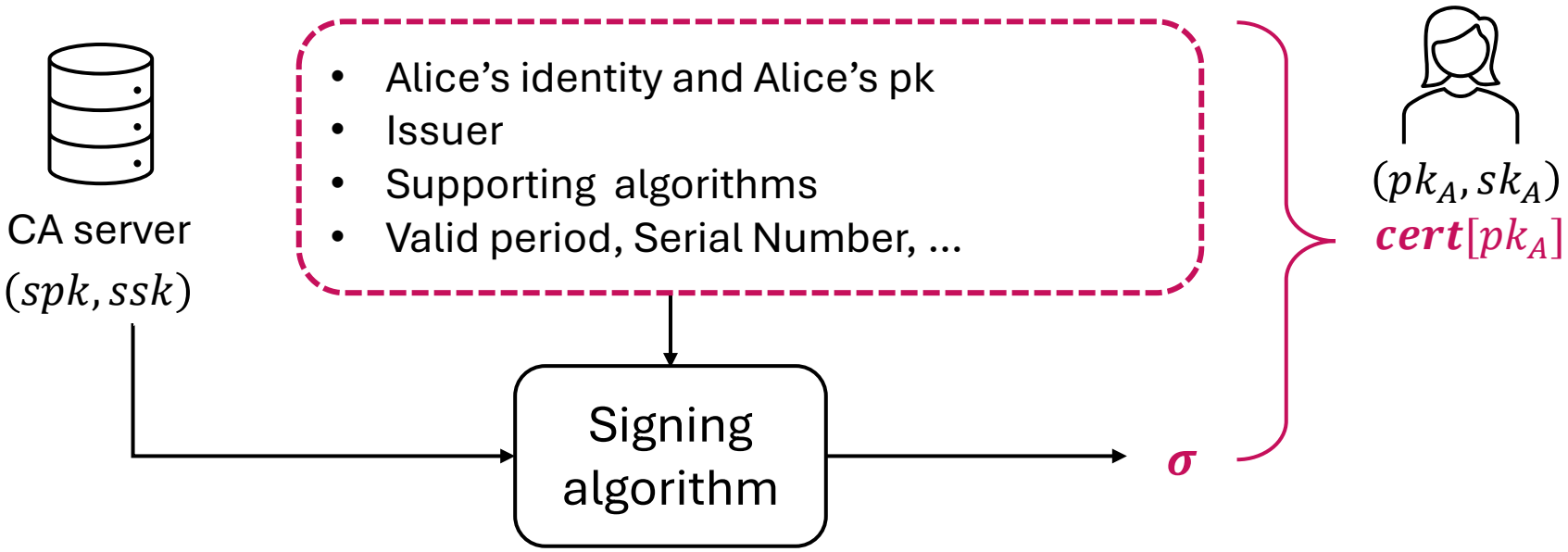
Digital Certificate

- Certificate: A signature generated by a trusted party (In short)
  - Verifies an ID and binds it to a public key
  - Securely distribute public keys
  - Issued by **CA** (Certificate Authority)



Digital Certificate

- What information does a certificate include?
  - X.509 standard: defines the format of public key certificates.

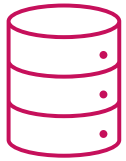


Digital Certificate

- Root Certificate and Certificate Chains
  - Hierarchical sequence of certificates
  - Trace the authenticity of a certificate back to a trusted **Root CA**
  - Only **root certificates** need to be pre-installed…

Digital Certificate

- Root Certificate and Certificate Chains
  - Hierarchical sequence of certificates
  - Trace the authenticity of a certificate back to a trusted **Root CA**
  - Only **root certificates** need to be pre-installed…



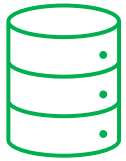
Root CA

(rpk, rsk)

$cert[rpk]$

Intermediate CA 1

(pk_1, sk_1)



Intermediate CA 2

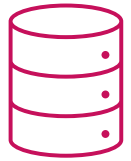
(pk_2, sk_2)

End server

(pk_3, sk_3)

Digital Certificate

- Root Certificate and Certificate Chains
  - Hierarchical sequence of certificates
  - Trace the authenticity of a certificate back to a trusted **Root CA**
  - Only **root certificates** need to be pre-installed…



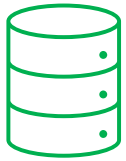
Root CA

(rpk, rsk)

$cert[rpk]$

Intermediate CA 1

(pk_1, sk_1)



Intermediate CA 2

(pk_2, sk_2)

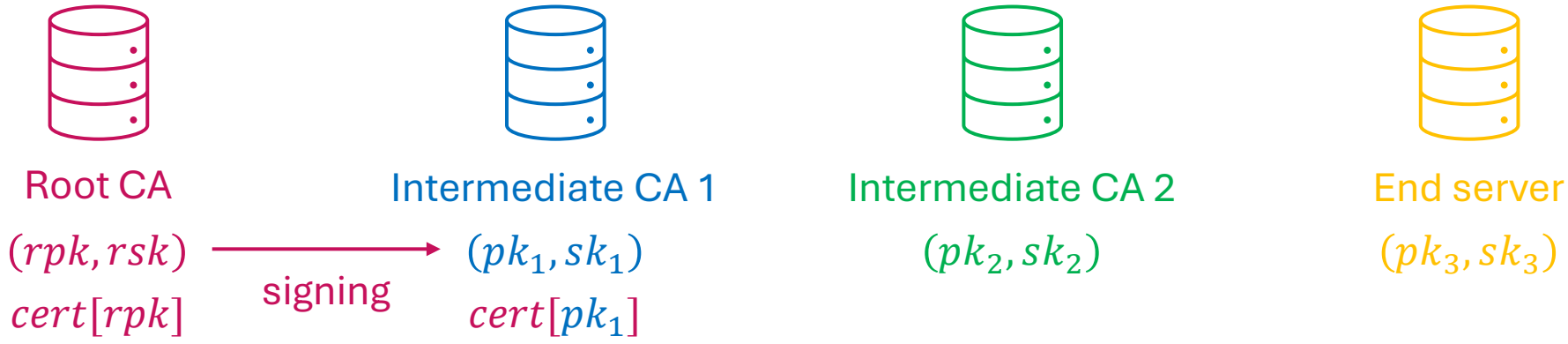
End server

(pk_3, sk_3)

- Pre-installed pk
- Self-signed cert

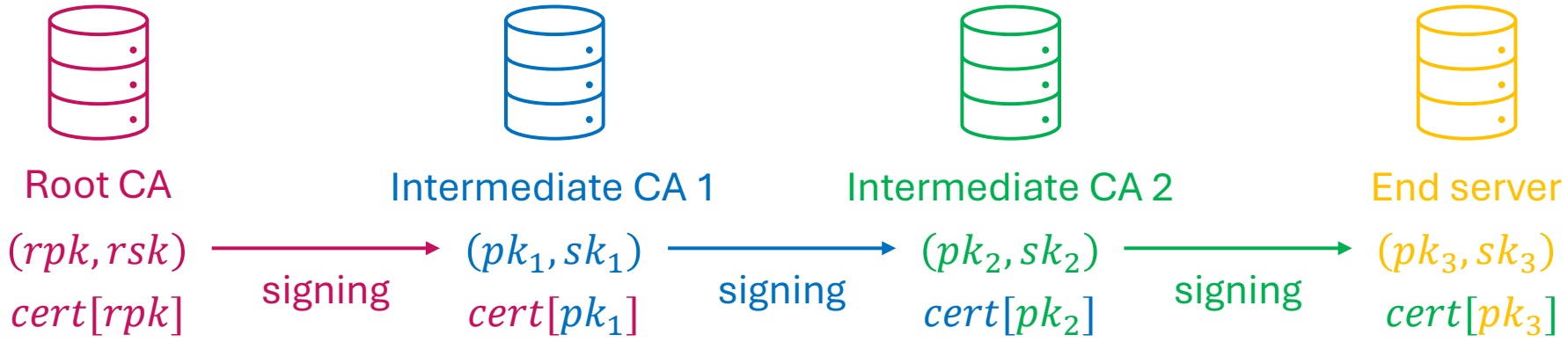
Digital Certificate

- Root Certificate and Certificate Chains
  - Hierarchical sequence of certificates
  - Trace the authenticity of a certificate back to a trusted **Root CA**
  - Only **root certificates** need to be pre-installed…



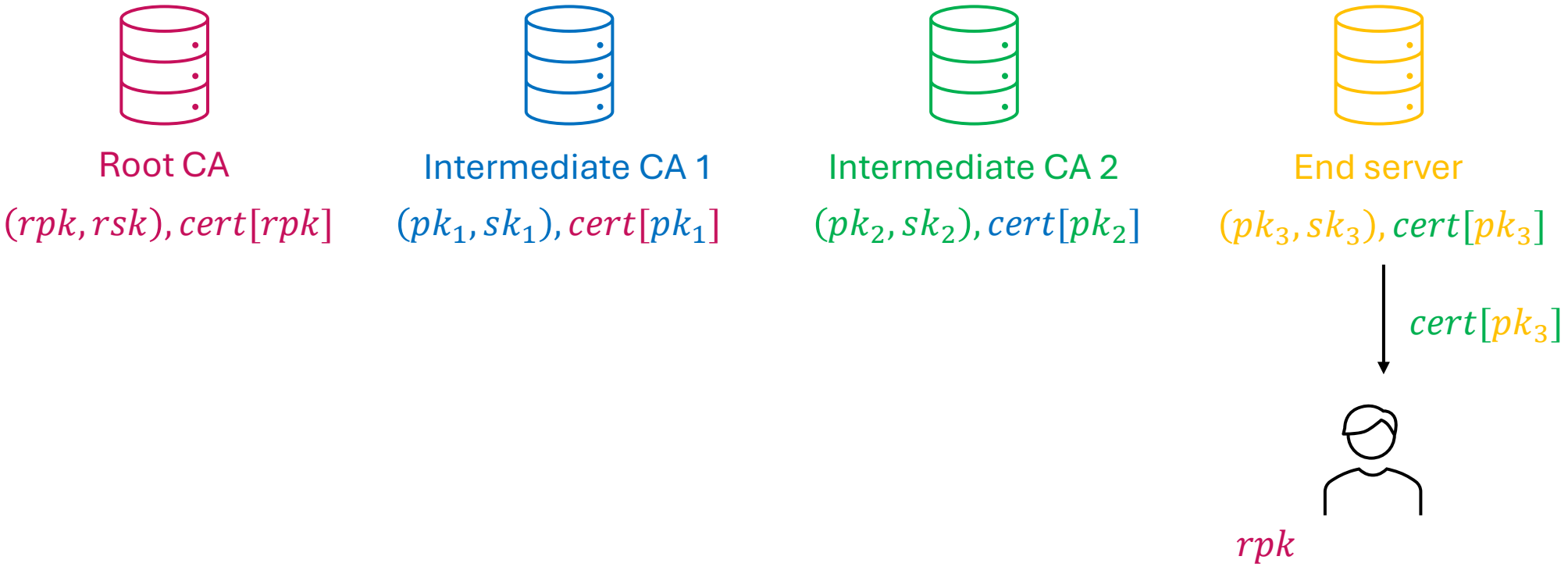
Digital Certificate

- Root Certificate and Certificate Chains
  - Hierarchical sequence of certificates
  - Trace the authenticity of a certificate back to a trusted **Root CA**
  - Only **root certificates** need to be pre-installed…



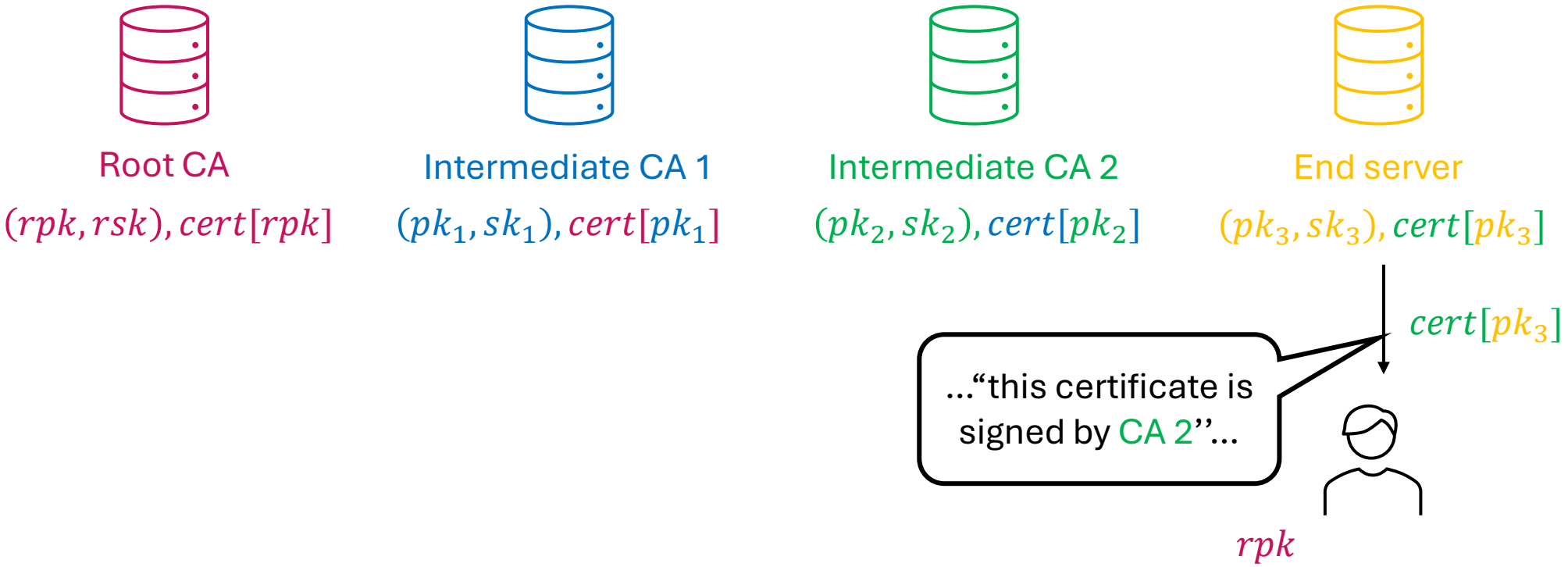
Digital Certificate

- Root Certificate and Certificate Chains



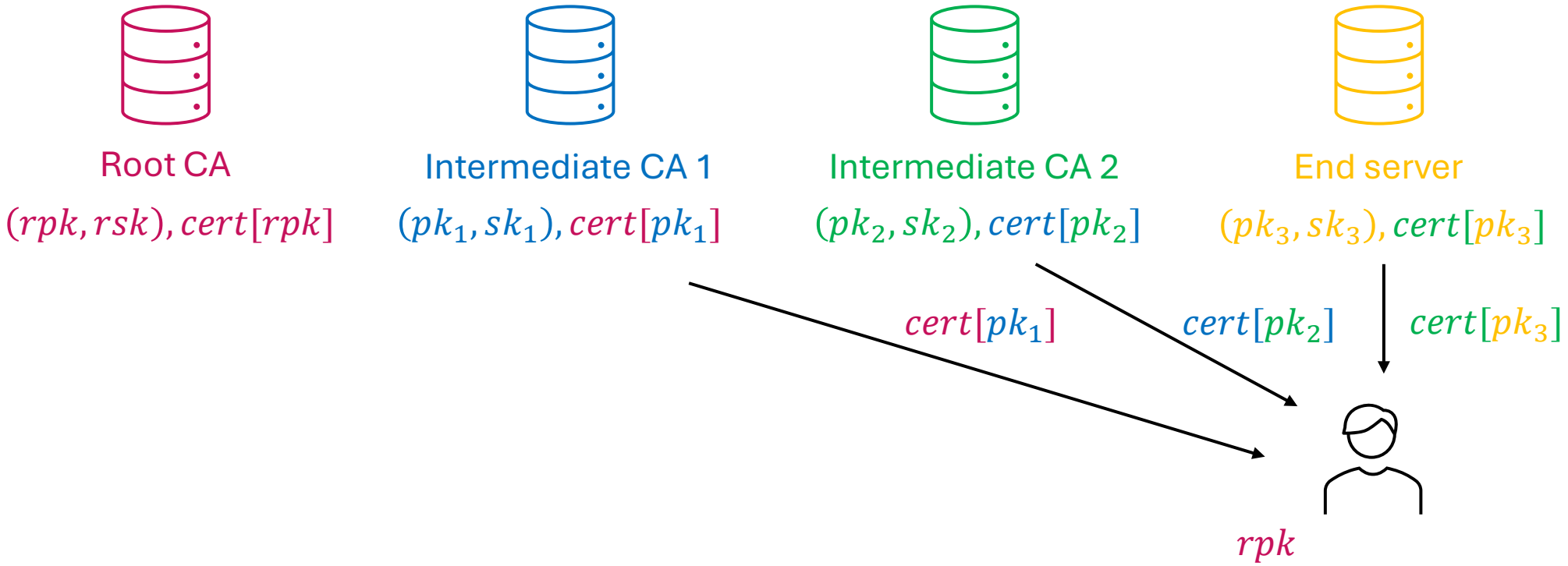
Digital Certificate

- Root Certificate and Certificate Chains



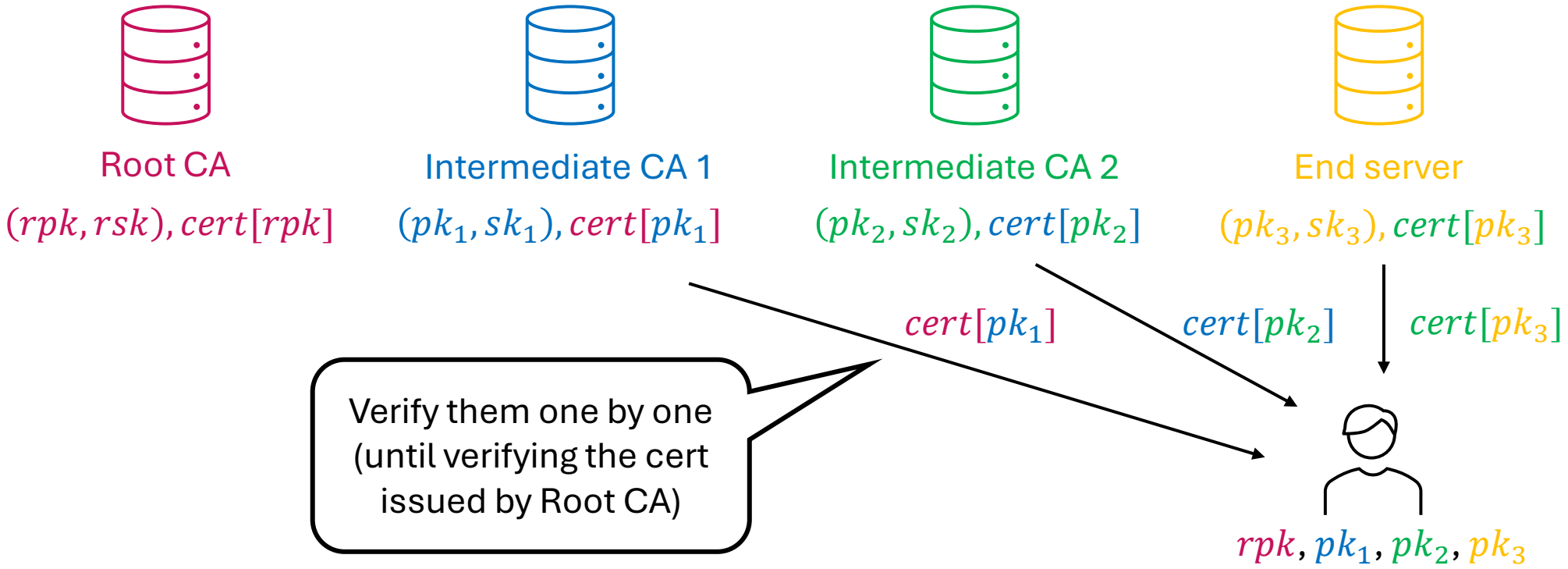
Digital Certificate

- Root Certificate and Certificate Chains



Digital Certificate

- Root Certificate and Certificate Chains



Exercise

1. Export a certificate from a website and write a simple program to read the certificate.
2. Find and export a pre-installed certificate on your laptop or PC (via browser) and use your program to read the certificate.
3. Implement the DH key exchange and derive a key using the shared DH secret.
4. Implement the man-in-the-middle attacks on the DH key exchange.

Further Reading

- DigiCert (one of the largest and most widely trusted CAs): https://www.digicert.com/
- Elliptic Curves: https://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/
- P-256 (secp256r1) curve: https://neuromancer.sk/std/nist/P-256
- The X.509 standard: https://en.wikipedia.org/wiki/X.509
- Public Key Infrastructure (PKI): https://en.wikipedia.org/wiki/Public_key_infrastructure