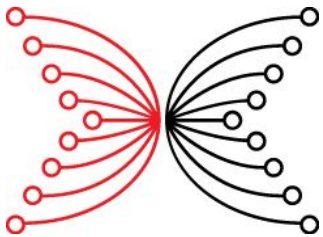
Lesson 5
Key exchange

A. Sorokin (Input | Output)

Self paced course

July 31, 2025



Motivation

Secure communication relies on encryption, which requires a **shared secret key**. However, if two parties (e.g., Alice and Bob) have never met, how can they agree on a key without exposing it to eavesdroppers (Eve)?

(i) **Problem:** Sending a key in plaintext allows interception.

(ii) **Solution:** *Key exchange protocols* enable secure key establishment over insecure channels.

Historical Context

Before public-key cryptography, key distribution required:

(i) physical key delivery (e.g., couriers with codebooks)

(ii) pre-shared keys.

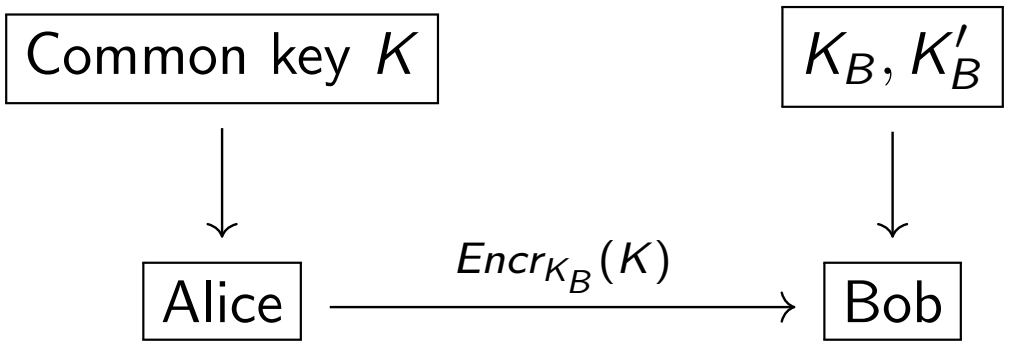
Such approach was expensive and impractical for large-scale systems

Public-key cryptography revolutionized this by allowing secure key exchange *without prior secrets*.

Key Exchange in a public-key cryptosystem

There are three popular ways for Alice and Bob to establish a **common key**.

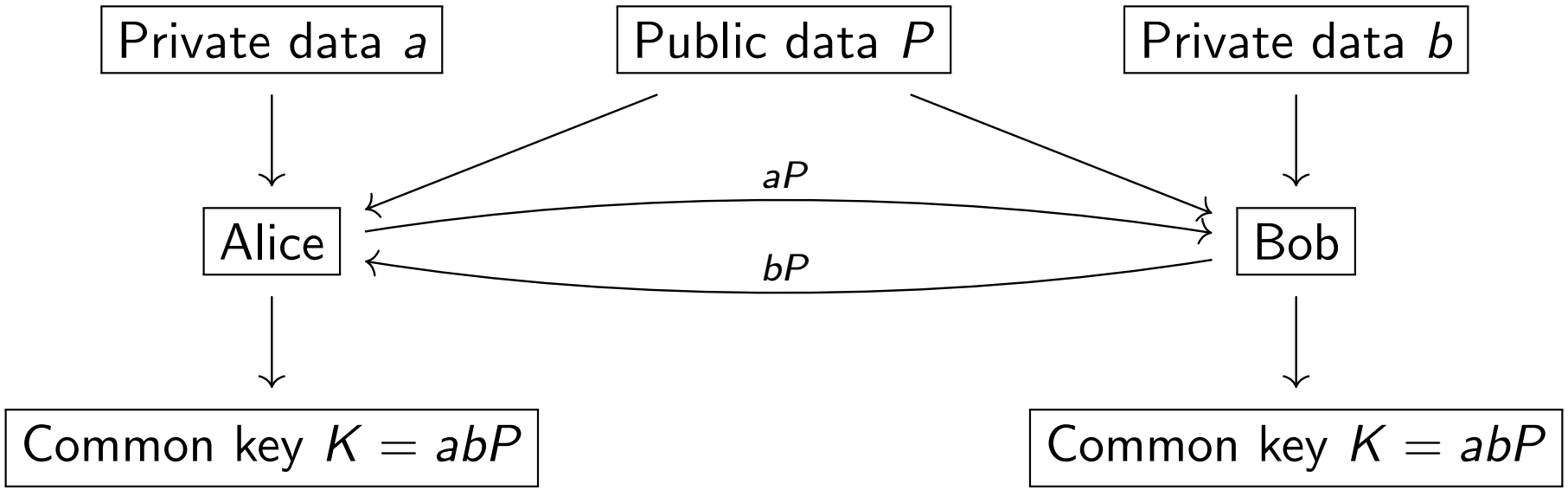
(i) Assuming that Alice and Bob are already connected via public-key cryptosystem, Alice makes a choice of a *common key* K and sends it to Bob.



Bob decrypts Alice's ciphertext and gets a *common key* K.

Diffie-Hellman key exchange (DH)

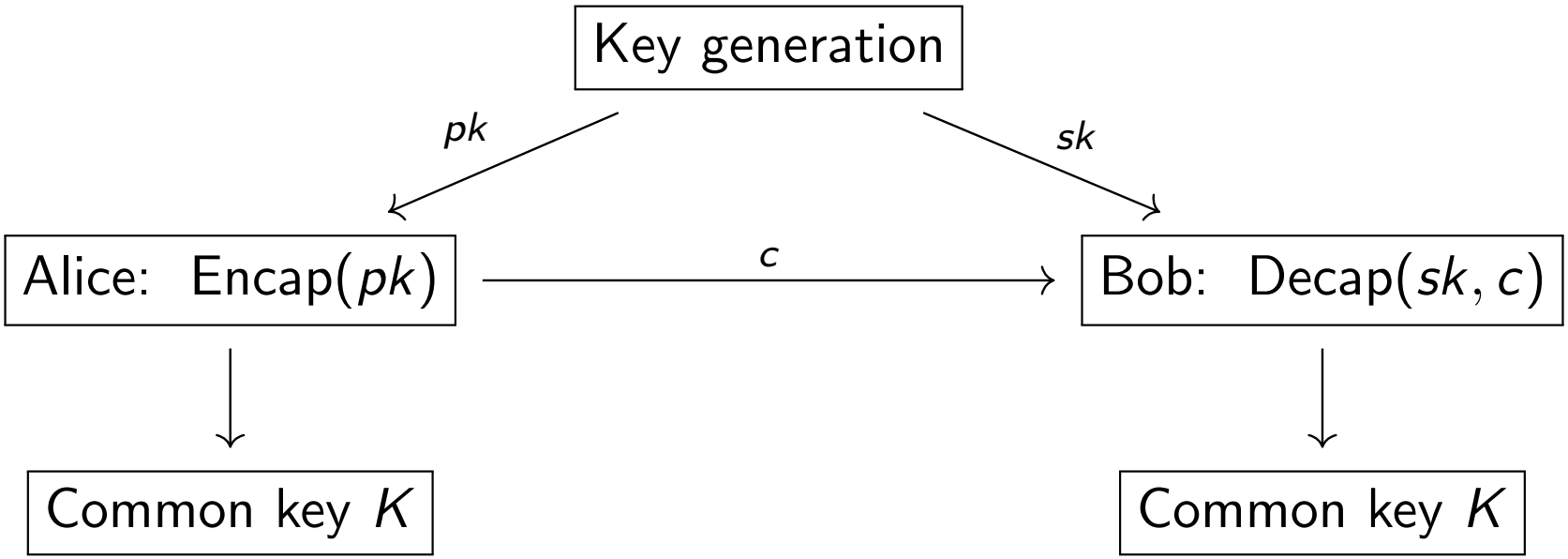
(ii) Alice and Bob agrees about some public data, then pick up their private keys and create their public keys.



Once they exchange their public keys, they derive a *common key* simultaneously.

Key Encapsulation Mechanism (KEM)

(iii) A KEM chooses a *common key* at random for Alice together with its *encapsulation*, which Alice sends to Bob. Bob uses KEM's *decapsulation* to get the *common key*.



Benefits and challenges of KEM

(i) **Non-interactive**: sender can encapculate a key without recipient being online.

(ii) **Simpler security proofs**: KEMs are easier to analyze than interactive protocols.

(iii) **Performance:** some KEMs are slower than some DH key exchange schemes.

Benefits and applications

(i) **No need for pre-shared secrets**: key exchange enables secure communication between strangers (HTTPS, VPNs, messaging apps).

(ii) **Efficiency**: symmetric encryption is faster than asymmetric, so key exchange enables efficient bulk encryption.

(iii) **Flexibility**: key exchange works over untrusted networks (public Wi-Fi).

Challenges

(i) **Man-in-the-middle (MITM) attack**: an attacker intercepts and alters exchanged keys to decrypt traffic via establishing separate keys with both parties.

A possible solution is to use *authentication* via digital signatures and certificates prevents MITM attack (used in TLS (HTTPS), SSH).

(ii) **Implementation flaws**: weak random number generators or misconfigured parameters.

Challenges

(iii) **Performance overhead**: key exchange adds latency (especially in RSA-based methods).

(iv) **Dependency on trusted third parties**: certificate authorities (CAs) must be trusted, as compromises undermine security.