# Lecture 12

# Public-key Cryptography

- **Benefit:** No longer need to assume that Alice and Bob already share a secret
- **Drawback:** Much slower than symmetric-key cryptography
  - Number theory calculations are much slower than XORs and bit-shifts

# Reading materials

- Encryption: Strengths and Weaknesses of Public-key Cryptography
- Public-key cryptography is a public invention due to Whitfield Diffie & Martin Hellman at Stanford Uni in 1976

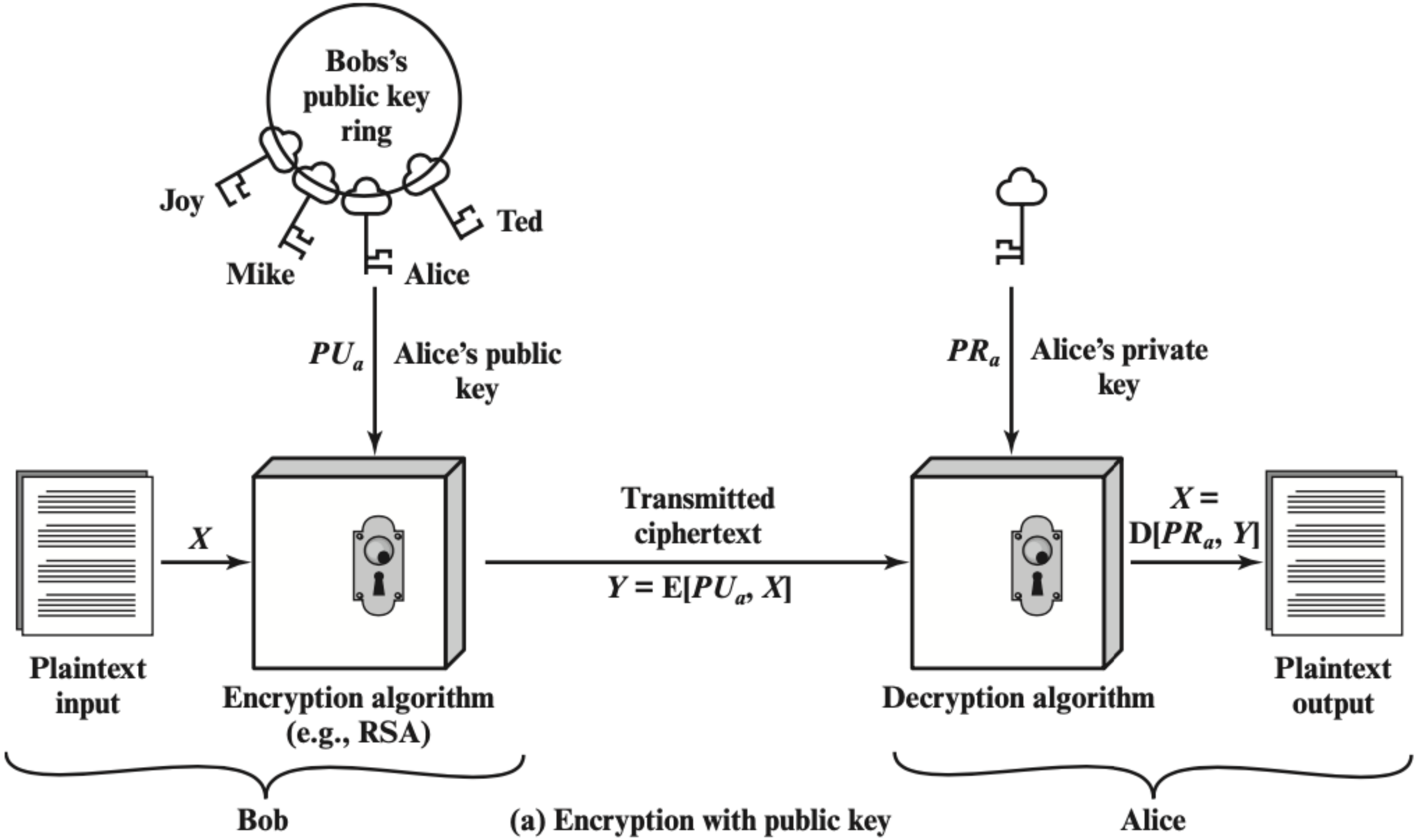
# Public-key cryptography

- **public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
  - a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
  - a **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**
- is **asymmetric** because
  - Not the same key
  - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures

# Public-Key Encryption

- Everybody can encrypt with the public key
- Only the recipient can decrypt with the private key

# Public-Key Cryptography - Encryption



(a) Encryption with public key

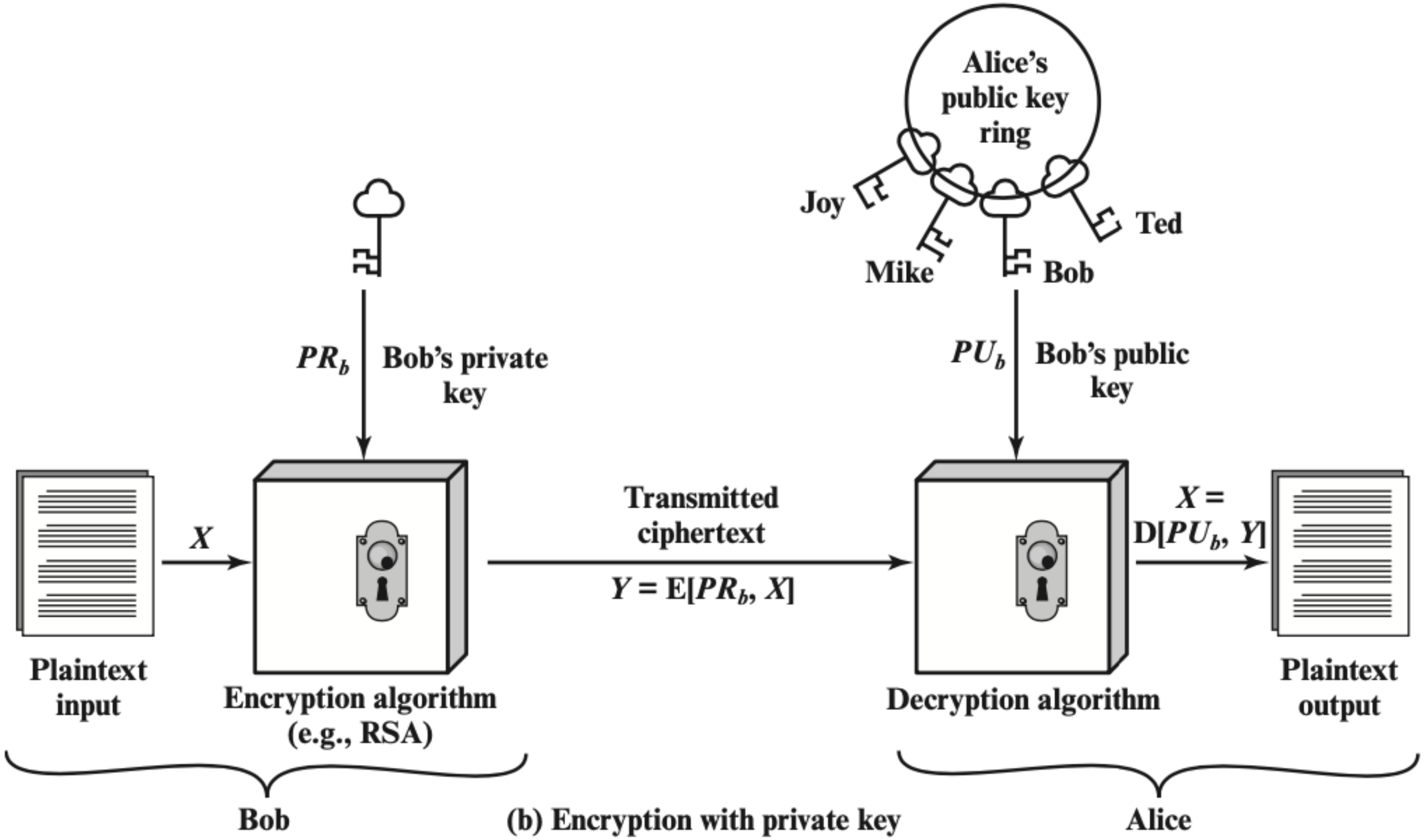
# Encryption steps

- step1: generate a pair of keys
- step2: keep the private key / secret key (SK) and distribute the public key (PK) – place PK in a public register or other accessible file
- step3: Bob encrypts the message with Alice’s PK
- step4: upon receiving the ciphertext (CT), Alice decrypt CT with SK

# Public-Key Encryption: Definition

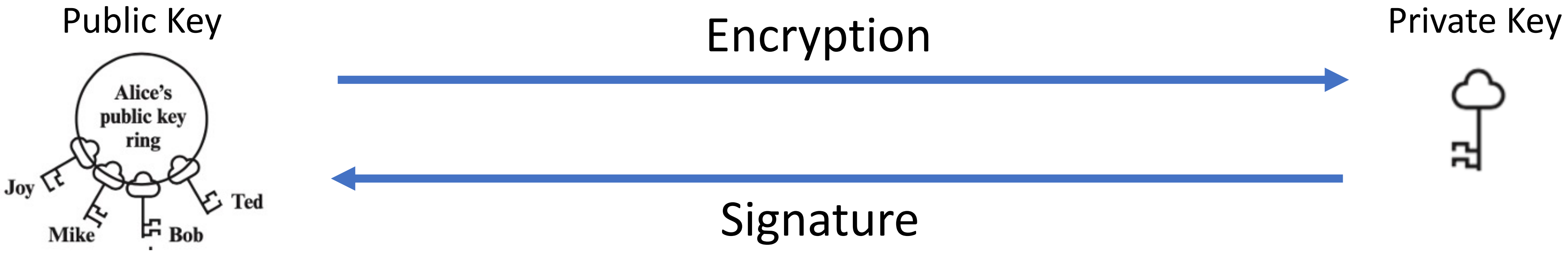
- Three parts:
  - KeyGen() → *PK*, *SK*: Generate a public/private keypair, where *PK* is the public key, and *SK* is the private (secret) key
  - Enc(*PK*, *M*) → *C*: Encrypt a plaintext *M* using public key *PK* to produce ciphertext *C*
  - Dec(*SK*, *C*) → *M*: Decrypt a ciphertext *C* using secret key *SK*
- Properties
  - **Correctness**: Decrypting a ciphertext should result in the message that was originally encrypted
    - Dec(*SK*, Enc(*PK*, *M*)) = *M* for all *PK*, *SK* ← KeyGen() and *M*
  - **Efficiency**: Encryption/decryption should be fast
  - **Security**: 1. Alice (the challenger) just gives Eve (the adversary) the public key, and Eve doesn't request encryptions. Eve cannot guess out anything; 2. computationally infeasible to recover M with PK and ciphertext

# Public-Key Cryptography - Signature



(b) Encryption with private key

# Review



Public Key

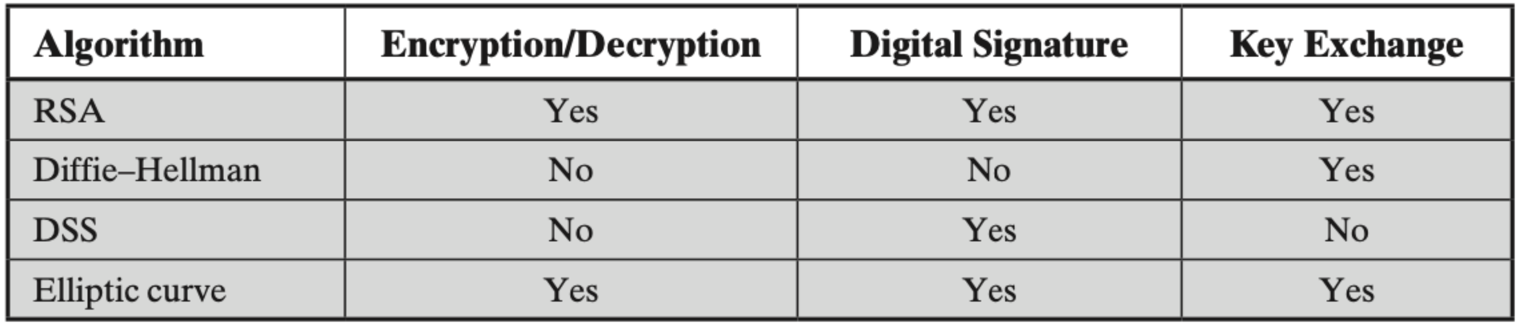
Encryption

Signature

Private Key

# Public-Key application

- can classify uses into 3 categories:
    - **encryption/decryption** (provide secrecy)
    - **digital signatures** (provide authentication)
    - **key exchange** (of session keys)
- some algorithms are suitable for all uses; others are specific to one
- Either of the two related keys can be used for encryption, with the other used for decryption



| Algorithm | Encryption/Decryption | Digital Signature | Key Exchange |
|---|---|---|---|
| RSA | Yes | Yes | Yes |
| Diffie–Hellman | No | No | Yes |
| DSS | No | Yes | No |
| Elliptic curve | Yes | Yes | Yes |