# Asymmetric cryptography

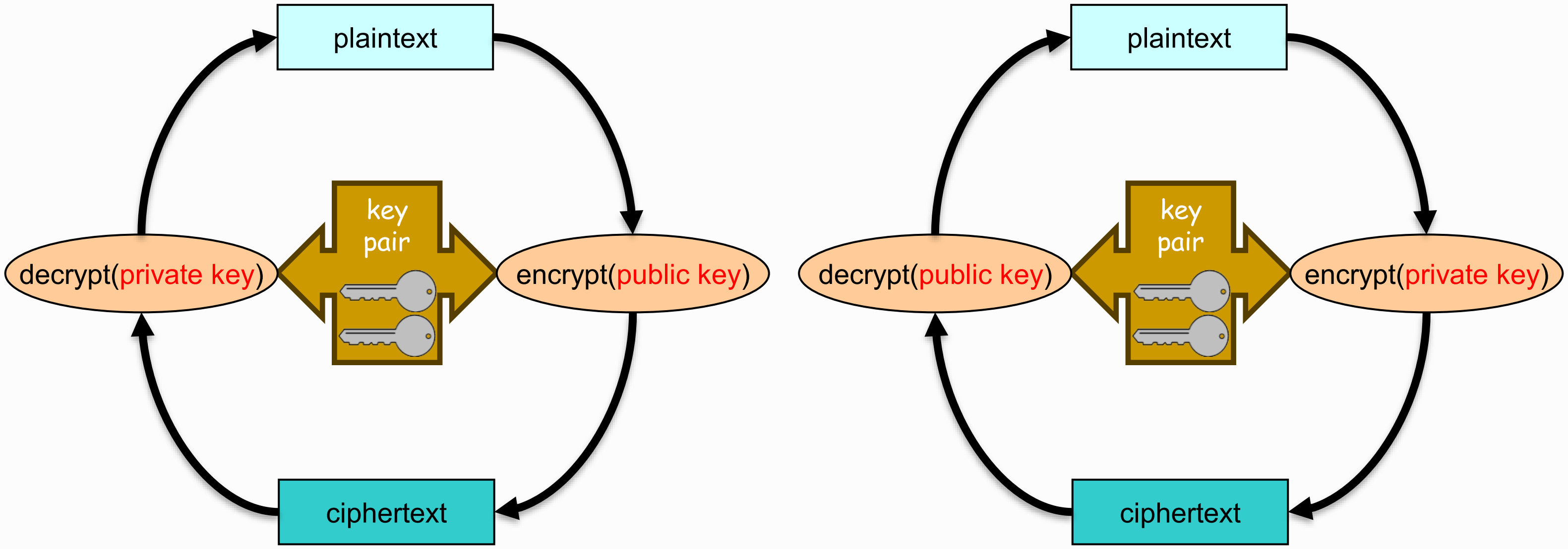
# Asymmetric (Block) Ciphers

- Use key pairs
  - One private key (personal, not transmittable)
  - One public key, available to all

- Allow
  - Confidentiality without any previous exchange of secrets
  - Authentication
    - Of contents (data integrity)
    - Of origin (source authentication, or digital signature)

# Operations of an asymmetric cipher

- Confidentiality
- Authentication (signature)

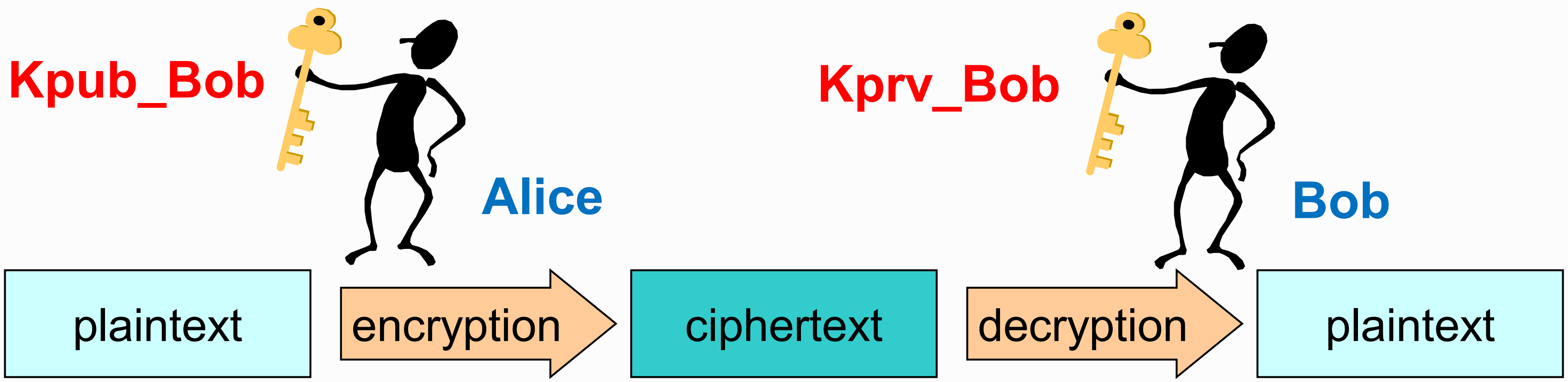


# Use cases: secure communication

- Secure communication with a target (Bob)
  - Alice encrypts plaintext **P** with Bob's public key **Kpub_Bob**

    **Alice: $C = \{P\}_{kpub\_Bob}$**
  - Bob decrypts cyphertext **C** with his private key **Kprv_Bob**

    **Bob: $P' = \{C\}_{kprv\_Bob}$**
  - **P'** should be equal to **P** (requires checking)
  - **Kpub_Bob** needs to be **known by Alice**

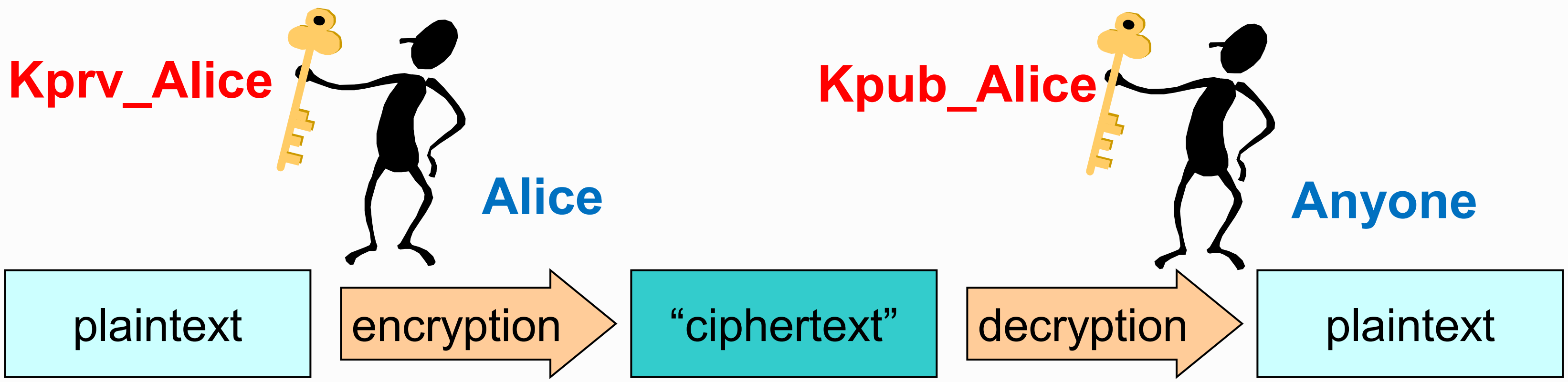


# Use cases: signature

- Data signature by Alice
  - Alice encrypts plaintext **P** with her private key **Kprv_Alice**

    **Alice: C = $\{P\}_{kprv\_Alice}$**
  - Anyone can decrypt cyphertext **C** with Alice's public key **Kpub_Alice**

    **Anyone: P'= $\{C\}_{kpub\_Bob}$**
  - If **P'** = **P**, then **C** is **Alice's signature** of **P**
  - **Kpub_Alice** needs to be **known by signature verifiers**



# Asymmetric ciphers

- Advantages
  - They are a fundamental authentication mechanism
  - They allow to explore features that are not possible with asymmetric ciphers
- Disadvantages
  - Performance
  - Usually are very inefficient and memory consuming
- Problems
  - Trustworthy distribution of public keys
  - Lifetime of key pairs

# Asymmetric ciphers

- Approaches: complex mathematic problems
  - Discrete logarithms of large numbers
  - Integer factorization of large numbers
- Most common algorithms
  - RSA
  - ElGamal
  - Elliptic curves (ECC)
- Other techniques with asymmetric key pairs
  - Diffie-Hellman (key agreement)

# RSA (Rivest, Shamir, Adelman, 1978)

- Keys
  - Private: (d, n)
  - Public: (e, n)

- Public key encryption (confidentiality)
  - $C = P^e \bmod n$
  - $P = C^d \bmod n$

P, C are big numbers!

$0 \leq P, C < n$

- Private key encryption (signature)
  - $C = P^d \bmod n$
  - $P = C^e \bmod n$

# Hybrid encryption

- Combines symmetric with asymmetric cryptography
  - Use the best of both worlds, while avoiding problems
  - Asymmetric cipher: Uses public keys (but it is slow)
  - Symmetric cipher: Fast (but with weak key exchange methods)

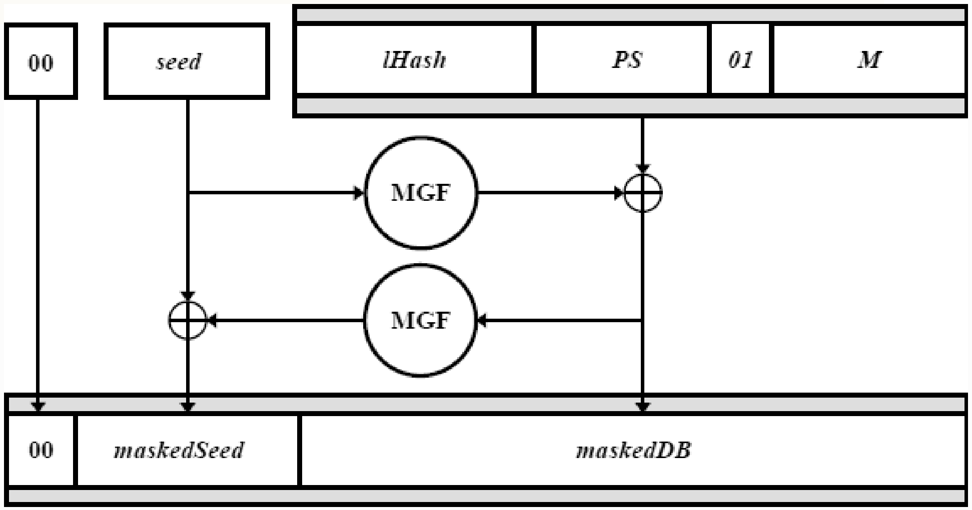
- Method:
  - Obtain $K_{pub}$ from the receiver
  - Generate a random $K_{sym}$
  - Calculate $C1 = E_{sym}( K_{sym}, P )$
  - Calculate $C2 = E_{asym}( K_{pub}, K_{sym} )$
  - Send C1 and C2
    - C1 = Text encrypted with symmetric key
    - C2 = Symmetric key encrypted with the receiver public key
      - May also contain the IV

# Randomization of asymmetric encryptions

- Non-deterministic (unpredictable) result of asymmetric encryptions
  - N encryptions of the **same value**, with the **same key**, should yield N **different results**
  - Goal:
    - Prevent the trial & error discovery of values encrypted w/ public keys
    - Prevent the cryptanalysis of the private key (noise)

- Approaches
  - Concatenation of the value to encrypt with two other values:
    - A fixed one (for integrity control)
    - A random one (for randomization)

# Randomization of public key encryptions: OAEP (Optimal Asymmetric Encryption Padding)




- ▷ lHash: digest over Label
- ▷ seed: random
- ▷ PS: zeros
- ▷ M: plaintext
- ▷ MGF: Mask Generation Function
  - Similar to Hash, but with variable size

# Diffie-Hellman key agreement (1976)

q (large prime)
$\alpha$ (primitive root mod q)

| Alice | | Bob |
|---|---|---|
| $a$ = random | | $b$ = random |
| $Y_a = \alpha^a \bmod q$ | $Y_a \rightarrow$ | |
| | $\leftarrow Y_b$ | $Y_b = \alpha^b \bmod q$ |
| $K_{ab} = Y_b^{\,a} \bmod q$ | | $K_{ba} = Y_a^{\,b} \bmod q$ |

$$K_{ab} = K_{ba}$$

# DH key agreement: MitM attack

$a$ = random

$Y_a = \alpha^a \bmod q$

$K_{ac} = Y_c^{\,a} \bmod q$

$Y_a \rightarrow$

$\leftarrow Y_c$

$c$ = random

$Y_c = \alpha^c \bmod q$

$K_{ca} = Y_a^{\,c} \bmod q$

$K_{cb} = Y_b^{\,c} \bmod q$

$\leftarrow Y_b$

$Y_c \rightarrow$

$b$ = random

$Y_b = \alpha^b \bmod q$

$K_{bc} = Y_c^{\,b} \bmod q$

# Elliptic Curve Cryptography (ECC)

- Elliptic curves are specific functions
  - They have a generator (G)
  - A private key $K_{prv}$ is an **integer** with a maximum of bits allowed by the curve
  - A public key $K_{pub}$ is a **point** $(x,y) = K_{prv} \times G$
  - Given $K_{pub}$, it should be hard to guess $K_{prv}$
- Curves
  - NIST curves (15)
    - P-192, P-224, P-256, P-384, P-521
    - B-163, B-233, B-283, B-409, B-571
    - K-163, K-233, K-283, K-409, K-571
  - Other curves
    - Curve25519 (256 bits)
    - Curve448 (448 bits)

# ECDH: DH with ECC

ECC curve → G

$a$ = random

$Y_a = aG$

$b$ = random

$Y_b = bG$

$Y_a$

$Y_b$

$K_{ab} = aY_b = abG$

$K_{ba} = bY_a = baG$

$K_{ab} = K_{ba}$

# ECC public key encryption

- Combines hybrid encryption with ECDH
- Method:
  - Obtain $K_{pub\_recv}$ from the receiver
  - Generate a random $K_{prv\_send}$ and the corresponding $K_{pub\_send}$
  - Calculate $K_{sym} = K_{prv\_send}\ K_{pub\_recv}$
  - $C = E(\ P, K_{sym}\ )$
  - Send C and $K_{pub\_send}$

  - Receiver calculates $K_{sym} = K_{pub\_send}\ K_{prv\_recv}$
  - $P = D(\ C, K_{sym}\ )$