Cryptography and Computer Security (CSS)

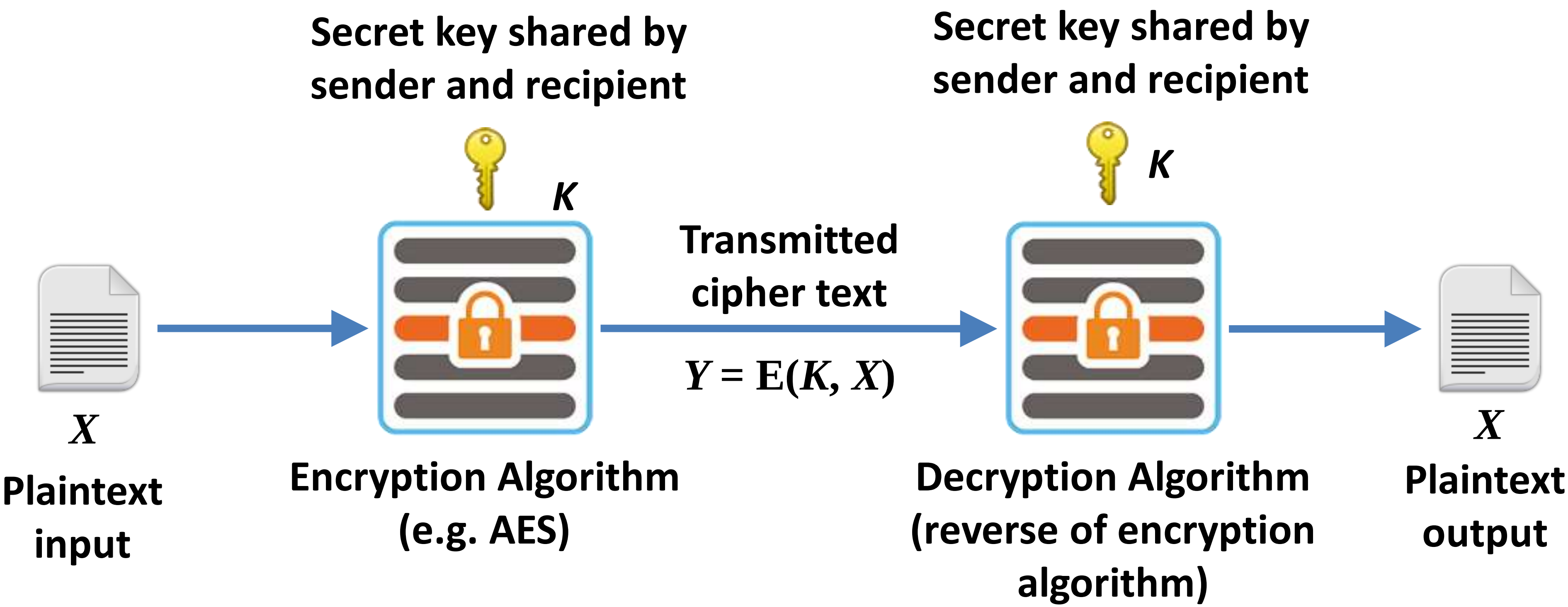
Lecture # 4

INTRODUCTION
TO
CSS COURSE
(CS401)

Unit II
Public Key Cryptography

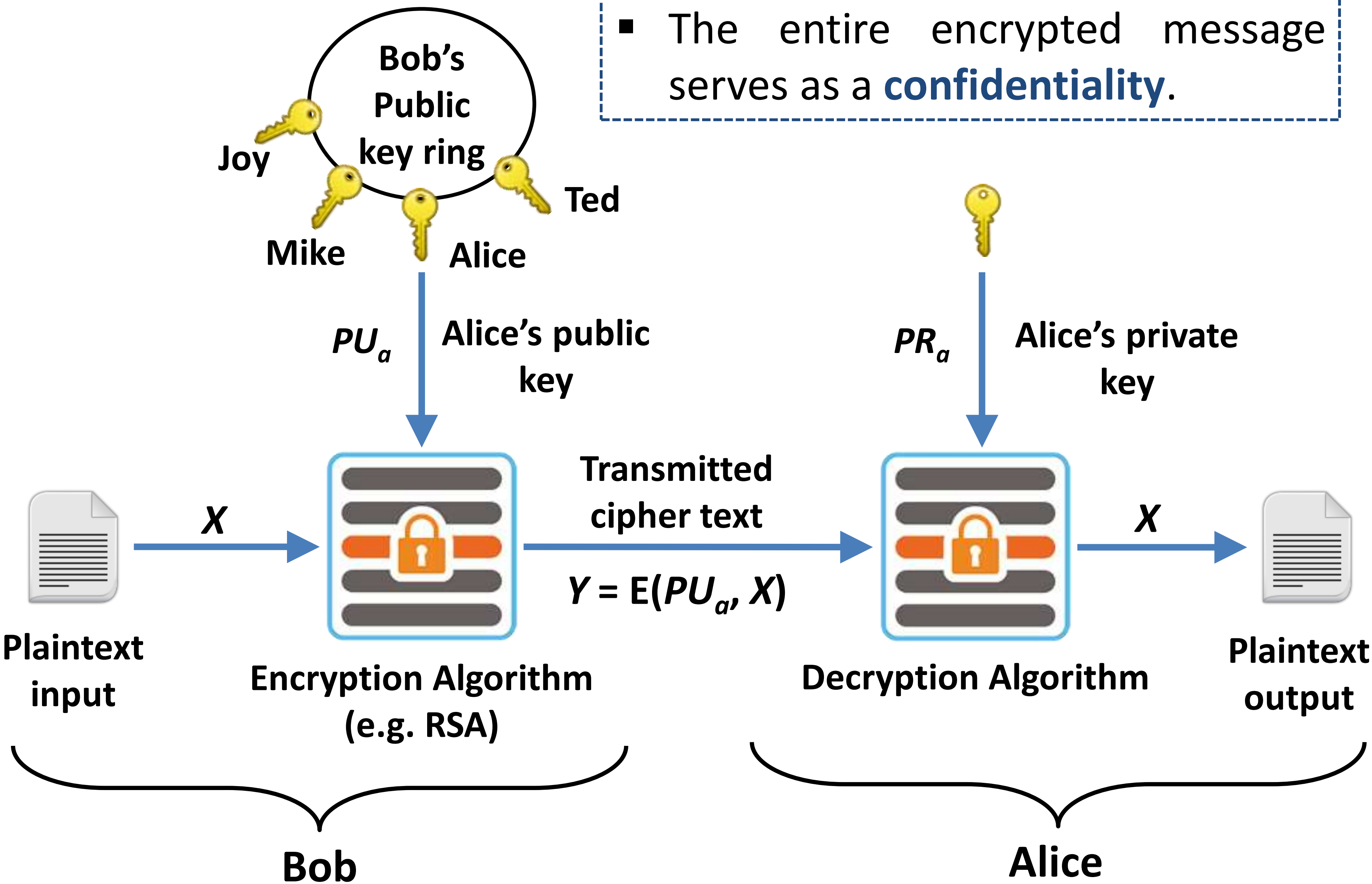
- Public Key Cryptosystems with Applications
- Requirements and Cryptanalysis
- RSA algorithm
- RSA computational aspects and security
- Diffie-Hillman Key Exchange algorithm
- Man-in-Middle attack

Symmetric key Encryption



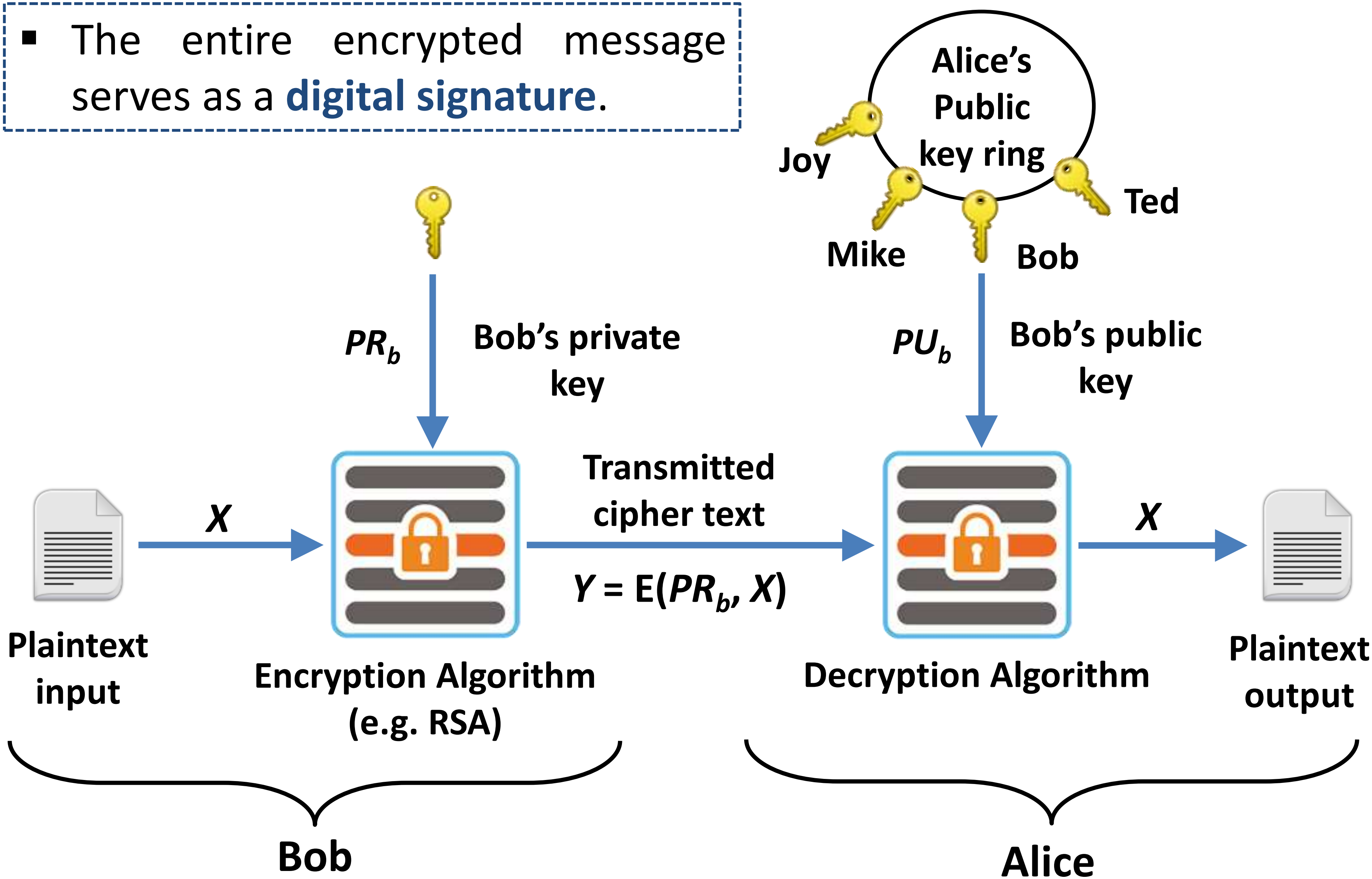
Asymmetric key Encryption with Public Key

- The entire encrypted message serves as a **confidentiality**.

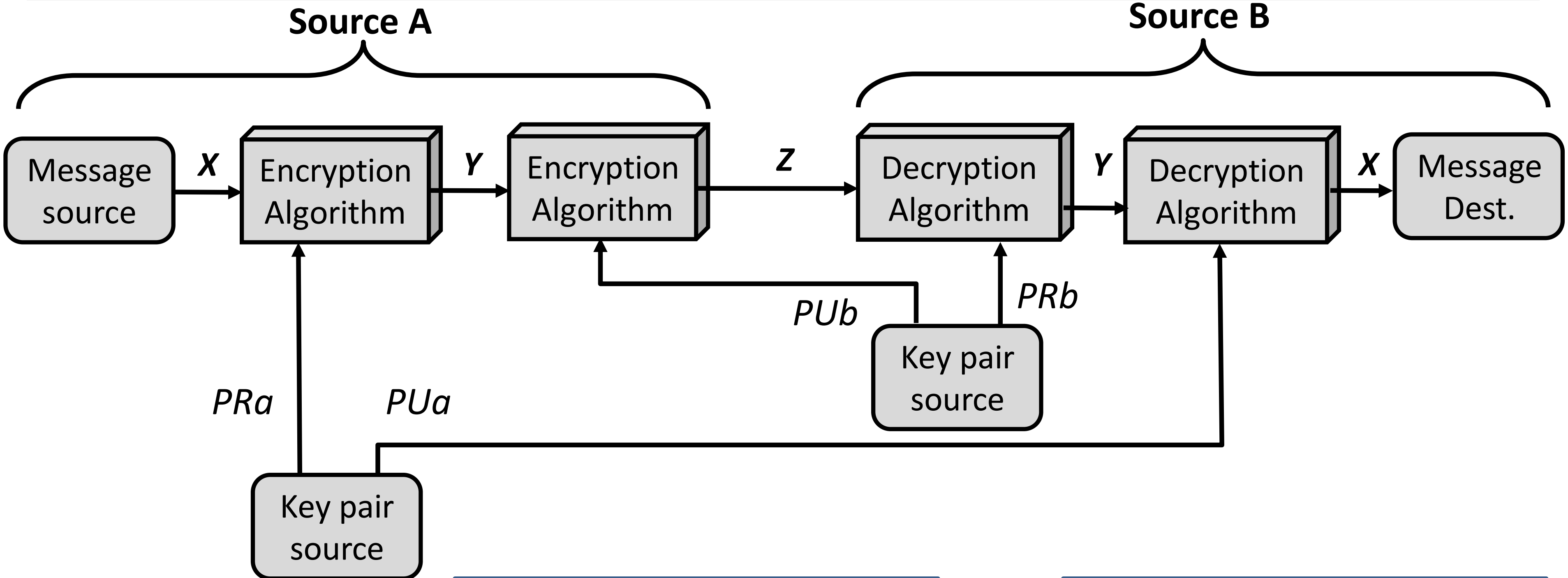


Asymmetric key Encryption with Private Key

- The entire encrypted message serves as a **digital signature**.



Authentication and Confidentiality



$Z = E(PU_b, E(PR_a, X))$

$X = D(PU_a, D(PR_B, Z))$

Applications for Public-Key Cryptosystems

- **Encryption/decryption:** The sender encrypts a message with the recipient's public key.
- **Digital signature:** The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.
- **Key exchange:** Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties.

RSA Algorithm

- **RSA** is a block cipher in which the Plaintext and Ciphertext are represented as integers between **0** and **n-1** for some n.
- Large messages can be broken up into a number of blocks.
- Each block would then be represented by an integer.

Step-1: Generate Public key and Private key

Step-2: Encrypt message using Public key

Step-3: Decrypt message using Private key

Step-1: Generate Public key and Private key

- Select two large prime numbers: **p** and **q**
- Calculate modulus : **$n = p * q$**
- Calculate Euler's totient function : **$\phi(n) = (p-1) * (q-1)$**
- Select **e** such that **e** is **relatively prime** to **$\phi(n)$** and **$1 < e < \phi(n)$**

Two numbers are relatively prime if they have no common factors other than 1.

- Determine **d** such that **$d * e \equiv 1 \pmod{\phi(n)}$**
- Publickey : **PU = { e, n }**
- Privatekey : **PR = { d, n }**

Step-1: Generate Public key and Private key

- Select two large prime numbers: **$p = 3$** and **$q = 11$**
- Calculate modulus : **$n = p * q$, $n = 33$**
- Calculate Euler's totient function : **$\phi(n) = (p-1) * (q-1)$**

  $\phi(n) = (3 – 1) * (11 – 1) = 20$
- Select **e** such that **e** is **relatively prime** to **$\phi(n)$** and **$1 < e < \phi(n)$**
- We have several choices for **e : 7, 11, 13, 17, 19** Let's take **e = 7**
- Determine **d** such that **$d * e \equiv 1 \pmod{\phi(n)}$**
- **$? * 7 \equiv 1 \pmod{20}$, $3 * 7 \equiv 1 \pmod{20}$**

- Public key : **PU = { e, n } , PU = { 7, 33 }**
- Private key : **PR = { d, n }, PR = { 3, 33 }**

- This is equivalent to finding d which satisfies $de = 1 + j.\phi(n)$ where j is any integer.
- We can rewrite this as $d = (1 + j. \phi(n)) / e$

Step-2 : Encrypt Message

- Encryption Using Public key: $C = M^e \bmod n$

Ciphertext

Input Message

Publickey

$PU = \{ e, n \}$, $PU = \{ 7, 33 \}$

For message $M = 14$

$C = 14^7 \bmod 33$

$C = [(14^1 \bmod 33) \times (14^2 \bmod 33) \times (14^4 \bmod 33)] \bmod 33$

$C = (14 \times 31 \times 4) \bmod 33 = 1736 \bmod 33$

$C = 20$

Step-3 : Decrypt Message

- Encryption Using Public key: $M = C^d \bmod n$

Plaintext Message

Cipher Message

Privatekey

PR = { d, n } , PR = { 3, 33 }

For Ciphertext C = 20

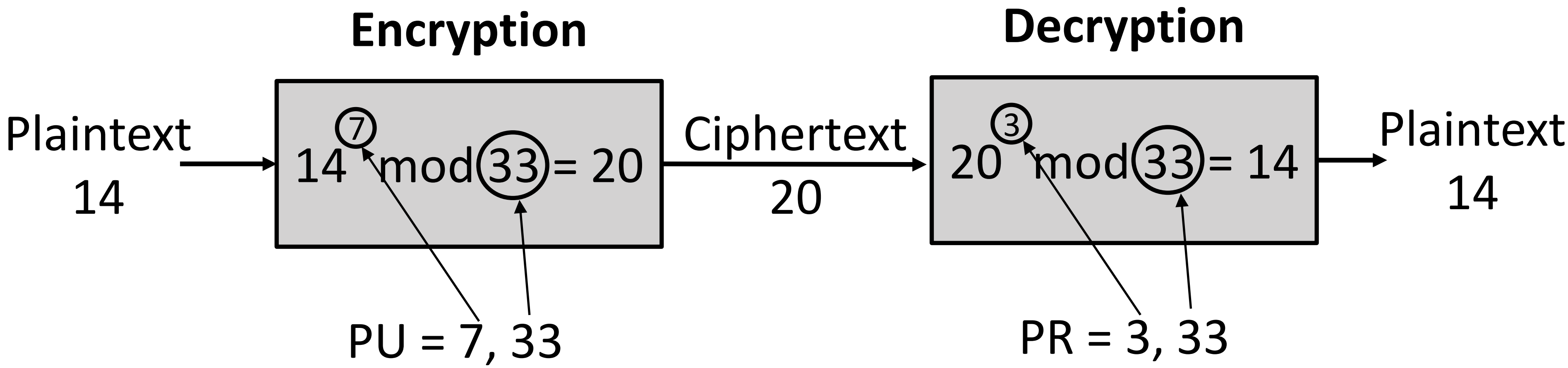
$M = 20^3 \bmod 33$

$M = [(20^1 \bmod 33) \text{ X } (20^2 \bmod 33)] \bmod 33$

$M = (20 \text{ X } 4) \bmod 33 = 80 \bmod 33$

$M = 14$

Example RSA Algorithm



RSA Example

- Find n, $\phi(n)$, e, d for p=7 and q= 19 then demonstrate encryption and decryption for M = 6

$n = p * q = 7 * 19 = 133$

$\phi(n) = (p - 1) * (q - 1) = 108$

Finding e relatively prime to 108

e = 2 => GCD(2, 108) = 2 (no)

e = 3 => GCD(3, 108) = 3 (no)

e = 5 => GCD(5, 108) = 1 (Yes)

- Finding d such that (d * e) mod $\phi(n)$ = 1
- We can rewrite this as d = (1 + j . $\phi(n)$) / e

j = 0 => d = 1 / 5 = 0.2 ← integer ? (no)

j = 1 => d = 109 / 5 = 21.8 ← integer ? (no)

j = 2 => d = 217 / 5 = 43.4 ← integer ? (no)

j = 3 => d = 325 / 5 = 65 integer ? (yes)

Public key :

PU = { e, n } = {5, 133}

Private key :

PR = { d, n } = {65, 133}

RSA Example – cont…

- Encryption:

$C = M^e \bmod n$

$PU = \{ e, n \}$, $PU = \{ 5, 133 \}$

For message $M = 6$

$C = 6^5 \bmod 133$

$C = 7776 \bmod 33$

$C = 62$

- Decryption:

$M = C^d \bmod n$

$PR = \{ d, n \}$, $PU = \{ 65, 133 \}$

For $C = 62$

$M = 62^{65} \bmod 133$

$M = 2666 \bmod 33$

$M = 6$

RSA Example

- P and Q are two prime numbers. P=7, and Q=17. Take public key E=5. If plain text value is 10, then what will be cipher text value according to RSA algorithm?
- n = 119
- $\phi(n) = 96$
- e = 5
- d = 77
- PU = { 5, 119 }
- PR = {77, 119}
- $C = 10^5 \bmod 119 \Rightarrow C = 40$

Diffie-Hellman key Exchange

- The purpose of the **Diffie-Hellman** algorithm is to enable two users to securely exchange a key that can be used for subsequent encryption of message.
- This algorithm depends for its effectiveness on the difficulty of computing **discrete logarithms**.

Primitive root

- Let $\boldsymbol{p}$ be a prime number
- Then $\boldsymbol{a}$is a primitive root for $\boldsymbol{p}$,if the powers of $\boldsymbol{a}$ modulo $\boldsymbol{p}$ generates all integers from **1 to $\boldsymbol{p}$ – 1** in some permutation.

$$a \bmod p, a^2 \bmod p, \ldots, a^{p-1} \bmod p$$

- Example: p = 7 then primitive root is 3 because powers of 3 mod 7 generates all the integers from 1 to 6

$$
\begin{aligned}
3^1 &= 3 \equiv 3 \ (mod\ 7) \\
3^2 &= 9 \equiv 2 \ (mod\ 7) \\
3^3 &= 27 \equiv 6 \ (mod\ 7) \\
3^4 &= 81 \equiv 4 \ (mod\ 7) \\
3^5 &= 243 \equiv 5 \ (mod\ 7) \\
3^6 &= 729 \equiv 1 \ (mod\ 7)
\end{aligned}
$$

Discrete Logarithm

- For any integer $\boldsymbol{b}$ and a primitive root $\boldsymbol{a}$ of prime number $\boldsymbol{p}$, we can find a unique exponent $\boldsymbol{i}$ such that

$$b = \ a^i(mod\ p) \ \ where\ 0 \leq i \leq (p-1)$$

- The exponent $\boldsymbol{i}$ is referred as the discrete logarithm of $\boldsymbol{b}$ for the base $\boldsymbol{a}$, mod $\boldsymbol{p}$. It expressed as below.

$$b\mathrm{dlog}_{a,p}(b)$$

Diffie-Hellman Key Exchange – Cont…

- User A and User B agree on two large prime numbers q and α. User A and User B can use insecure channel to agree on them.
- User A selects a random integer $X_A < q$ and calculates Y_A

Diffie-Hellman Key Exchange – Cont…

Global Public Elements

q prime number

$\alpha\alpha$<q and α is primitive root of q

User A Key Generation

Select private $X_A X_A < q$

Calculate public $Y_A Y_A = \alpha^{XA} \bmod q$

User B Key Generation

Select private $X_B X_B < q$

Calculate public $Y_B Y_B = \alpha^{XB} \bmod q$

Diffie-Hellman Key Exchange – Cont…

User A Key Generation

Select private $X_A X_A < q$

Calculate public $Y_A Y_A = \alpha^{XA} \bmod q$

User B Key Generation

Select private $X_B X_B < q$

Calculate public $Y_B Y_B = \alpha^{XB} \bmod q$

Calculation of Secret Key by User A

$$= (Y_B)^{XA} \bmod q$$

Calculation of Secret Key by User b

$$= (Y_A)^{XB} \bmod q$$

Diffie-Hellman Key Exchange – Cont…

User A Key Generation

Private $X_A X_A < q$, Public Y_A $Y_A = \alpha^{XA} \bmod q$

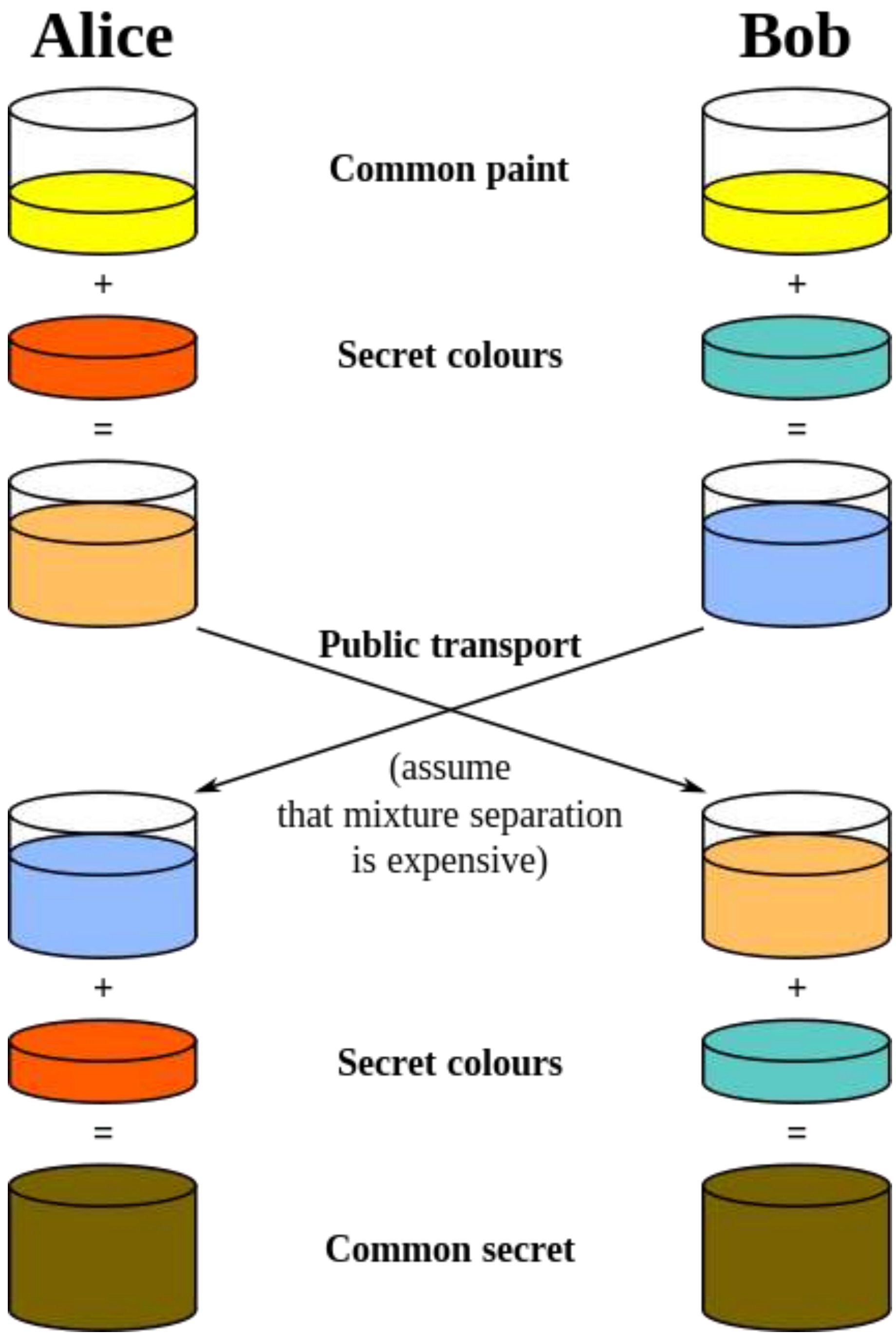
User B Key Generation

Private $X_B X_B < q$, Public $Y_B Y_B = \alpha^{XB} \bmod q$

Secret Key by User A : $K = (Y_B)^{XA} \bmod q$

Secret Key by User B : $K = (Y_A)^{XB} \bmod q$

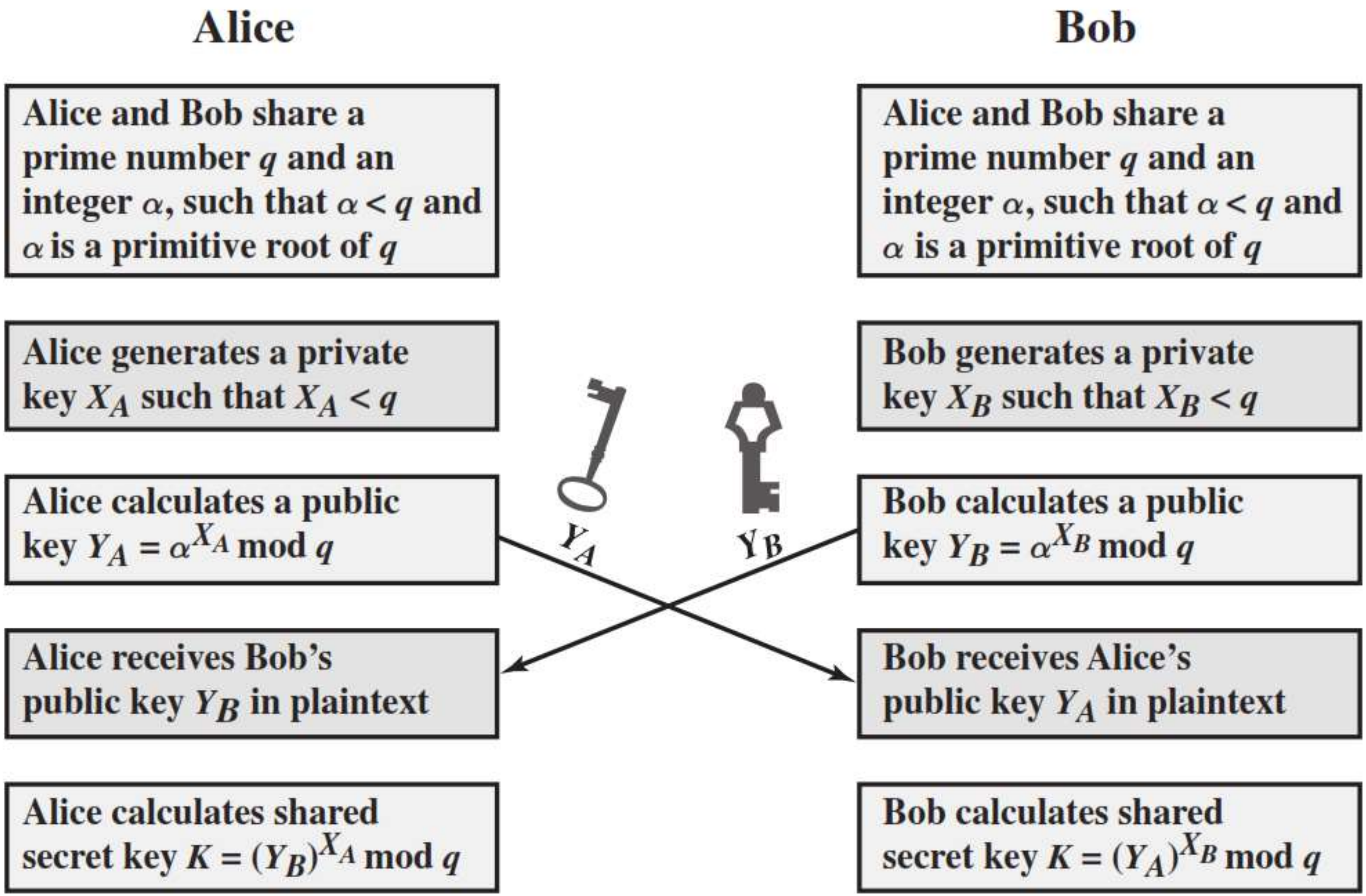
$$K = (Y_B)^{XA} \bmod q$$
$$K = (\alpha^{XB} \bmod q)^{XA} \bmod q$$
$$K = (\alpha^{XB})^{XA} \bmod q$$
$$K = \alpha^{XB\,XA} \bmod q$$
$$K = (\alpha^{XA})^{XB} \bmod q$$
$$K = (\alpha^{XA} \bmod q)^{XB} \bmod q$$
$$K = (Y_A)^{XB} \bmod q$$


Alice
Bob
Common paint
+
Secret colours
=
Public transport
(assume
that mixture separation
is expensive)
+
Secret colours
=
Common secret

Diffie-Hellman Key Exchange Example

- Alice and bob agrees on a prime number $\boldsymbol{q = 23}$
- $\boldsymbol{\alpha = 5}$ as primitive root of q
- Alice selects a private integer $\boldsymbol{X_A = 6}$
- Alice computes $Y_A = \alpha^{XA} \bmod q$ =>$\boldsymbol{Y_A = 5^6 \bmod 23 = 8}$
- Bob selects a private integer $\boldsymbol{X_B = 15}$
- Bob computes $Y_B = \alpha^{XB} \bmod q$ =>$\boldsymbol{Y_B = 5^{15} \bmod 23 = 19}$
- Alice sends Y_A to Bob and Bob sends Y_B to Alice
- Alice computes key $K = (Y_B)^{XA} \bmod q$ =>$\boldsymbol{K = (19)^6\, mod\, 23}$
- $\boldsymbol{K = 2}$
- Bob computes key $K = (Y_A)^{XB} \bmod q$=>$\boldsymbol{K = (8)^{15}\, mod\, 23}$
- $\boldsymbol{K = 2}$

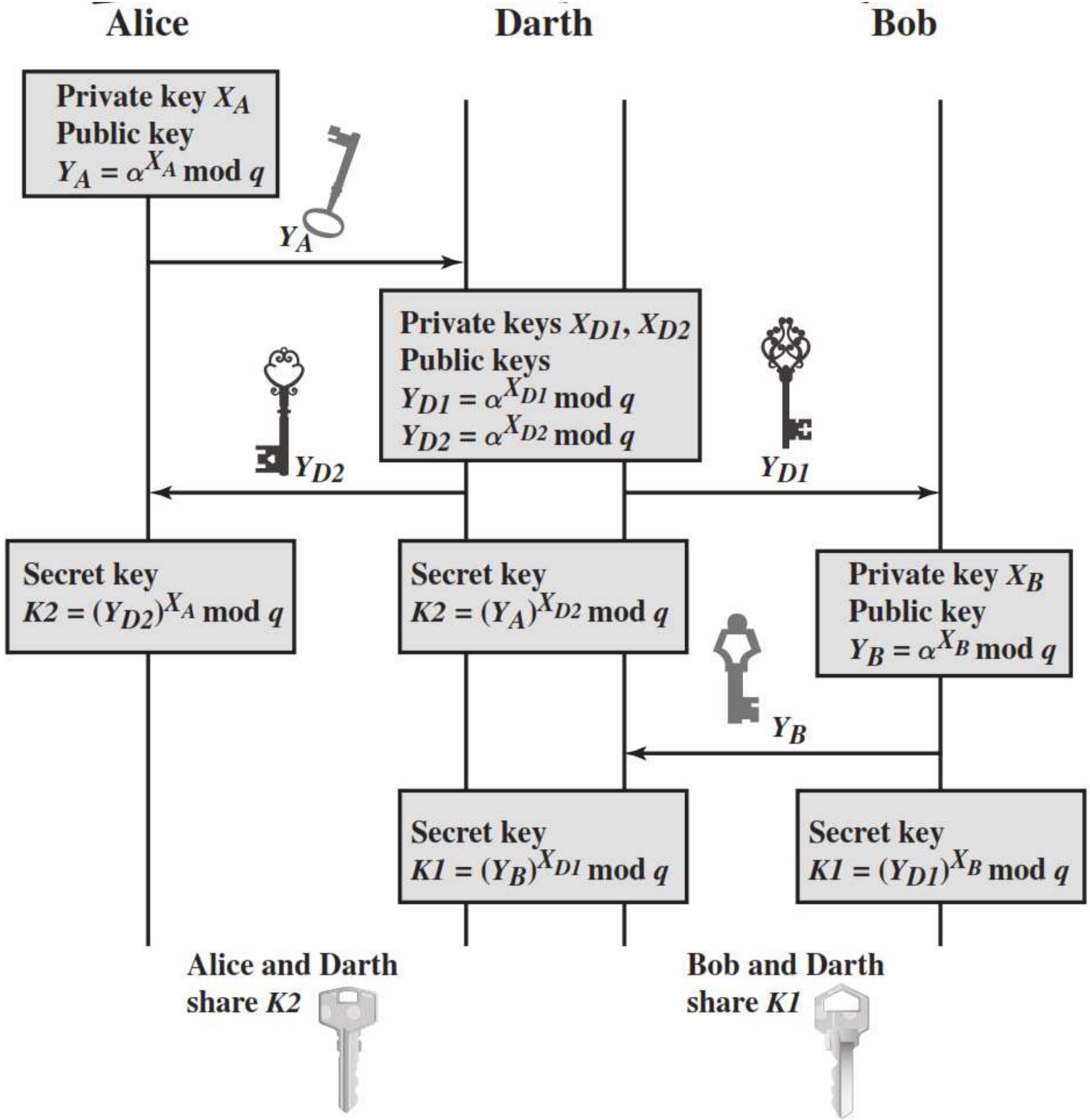
Diffie-Hellman Key Exchange Illustration



Man in the middle attack

- Suppose Alice and Bob wish to exchange keys, and Darth is the adversary.

1. Darth prepares for the attack by generating two random private keys X_{D1} and X_{D2} and then computes corresponding public keys Y_{D1} and Y_{D2}.
2. Alice transmits Y_A to Bob.
3. Darth intercepts Y_A and transmits Y_{D1} to Bob. Darth also calculates $K_2=(Y_A)^{XD2} \bmod q$.
4. Bob receives Y_{D1} and calculates $K_1 = (Y_{D1})^{XB} \bmod q$.
5. Bob transmits Y_B to Alice.
6. Darth intercepts Y_B and transmits Y_{D2} to Alice. Darth calculates $K_1=(Y_B)^{XD1} \bmod q$.
7. Alice receives Y_{D2} and calculates $K_2 = (Y_{D2})^{XA} \bmod q$.


Alice
Darth
Bob
Private key X_A
Public key
$Y_A = \alpha^{X_A} \bmod q$
Y_A
Private keys X_{D1}, X_{D2}
Public keys
$Y_{D1} = \alpha^{X_{D1}} \bmod q$
$Y_{D2} = \alpha^{X_{D2}} \bmod q$
Y_{D2}
Y_{D1}
Secret key
$K2 = (Y_{D2})^{X_A} \bmod q$
Secret key
$K2 = (Y_A)^{X_{D2}} \bmod q$
Private key X_B
Public key
$Y_B = \alpha^{X_B} \bmod q$
Y_B
Secret key
$K1 = (Y_B)^{X_{D1}} \bmod q$
Secret key
$K1 = (Y_{D1})^{X_B} \bmod q$
Alice and Darth share $K2$
Bob and Darth share $K1$