Diffie Hellman Key Exchange

Public-key algorithms

- Public-key algorithms are based on mathematical functions rather than on substitution and permutation.

- More important, public-key cryptography is **asymmetric**, involving the use of **two separate keys**, in contrast to symmetric encryption which uses only one key.

- The use of two keys has profound consequences in the areas of **confidentiality, key distribution, and authentication**

- A cryptographic algorithm that uses two related keys, a public key and a private key.

- The two keys have the property that deriving the private key from the public key is computationally infeasible.

Principles of public-key cryptosystems

- A **public-key cryptosystem** is based on a pair of keys:
- **Public key:** Shared openly for encryption or verification.
- **Private key:** Kept secret for decryption or signing.
- The security relies on hard mathematical problems (like factorization or **discrete logarithms**).

- **Two-key Pair (Key Asymmetry)**
  - Each user has a **pair of keys**:
    - Public key (Kpub) is known to everyone.
    - Private key (Kpriv) is known only to the owner.
  - Keys are **mathematically related** but it's **computationally infeasible** to derive the private key from the public key.
- **Encryption/Decryption**
  - **Encryption with Public Key:**
    Anyone can encrypt a message using the recipient's public key.

$$C=E_{Kpub}(M)$$

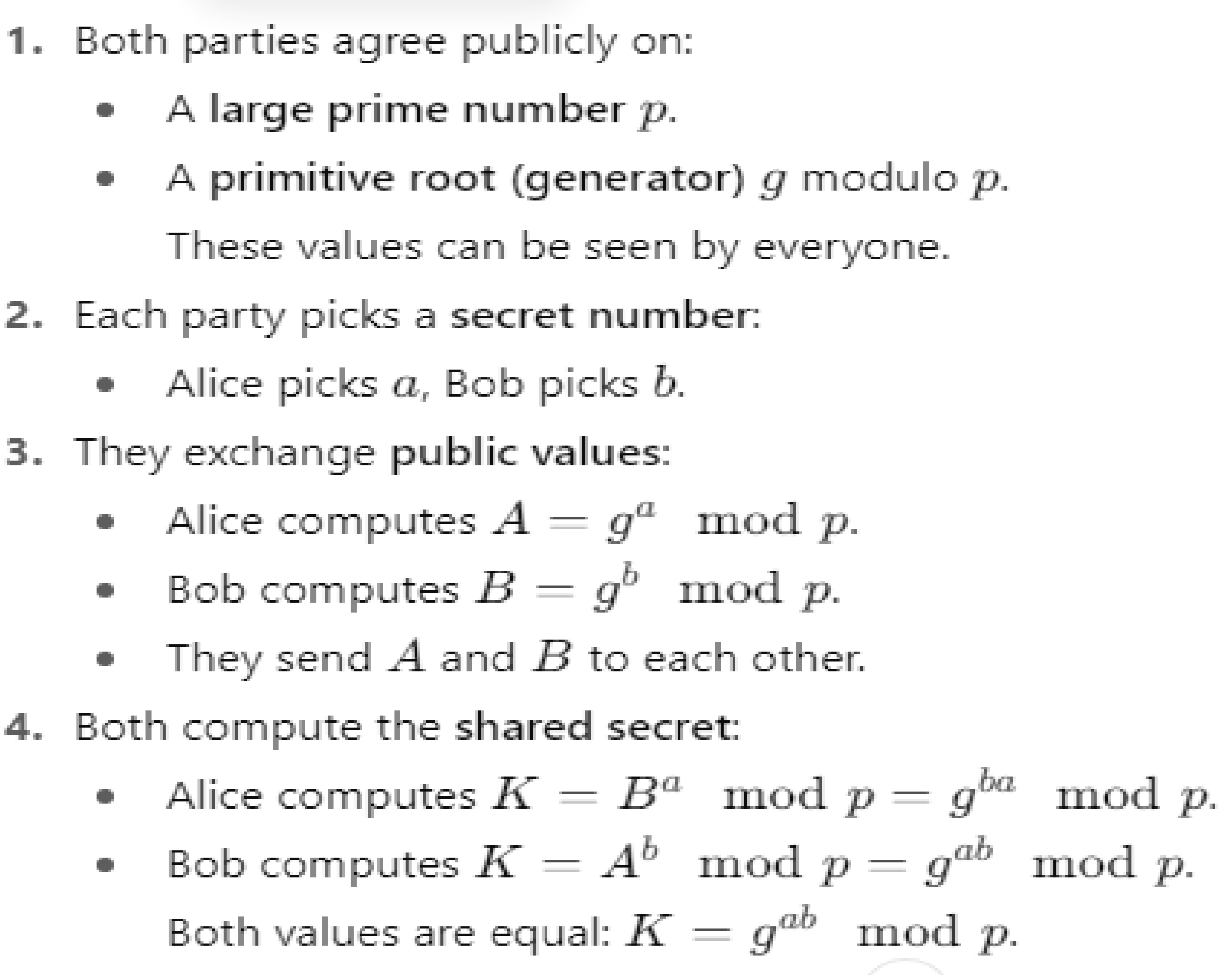
  - **Decryption with Private Key:**
    Only the private key holder can decrypt.

$$M=D_{Kpriv}(C)$$

Diffie–Hellman Key Exchange

- Introduced in **1976** by **Whitfield Diffie** and **Martin Hellman**.
- It's a **cryptographic protocol** that allows **two parties to establish a shared secret key** over a public (insecure) channel.
- The shared key can then be used for **symmetric encryption**.
- The magic is that **no secret key is sent over the network**—yet both parties end up with the same key.
- Security relies on the **difficulty of solving the discrete logarithm problem**.

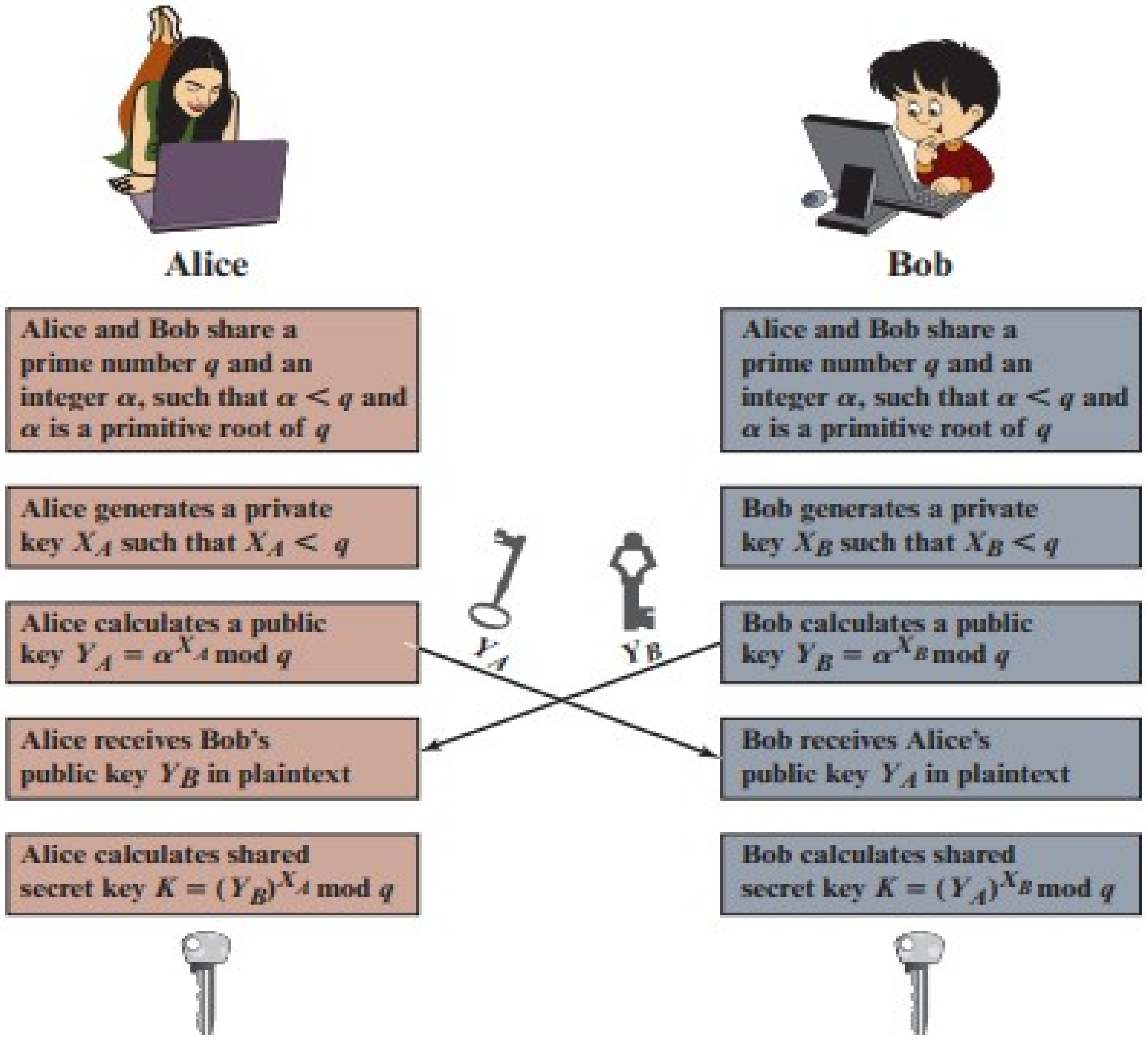
How it Works



1. Both parties agree publicly on:
   - A **large prime number** p.
   - A **primitive root (generator)** g modulo p.

   These values can be seen by everyone.
2. Each party picks a **secret number**:
   - Alice picks a, Bob picks b.
3. They exchange **public values**:
   - Alice computes $A = g^a \mod p$.
   - Bob computes $B = g^b \mod p$.
   - They send A and B to each other.
4. Both compute the **shared secret**:
   - Alice computes $K = B^a \mod p = g^{ba} \mod p$.
   - Bob computes $K = A^b \mod p = g^{ab} \mod p$.

   Both values are equal: $K = g^{ab} \mod p$.



Alice and Bob share a prime number q and an integer α, such that $\alpha < q$ and α is a primitive root of q

Alice generates a private key X_A such that $X_A < q$

Alice calculates a public key $Y_A = \alpha^{X_A} \bmod q$

Alice receives Bob's public key Y_B in plaintext

Alice calculates shared secret key $K = (Y_B)^{X_A} \bmod q$

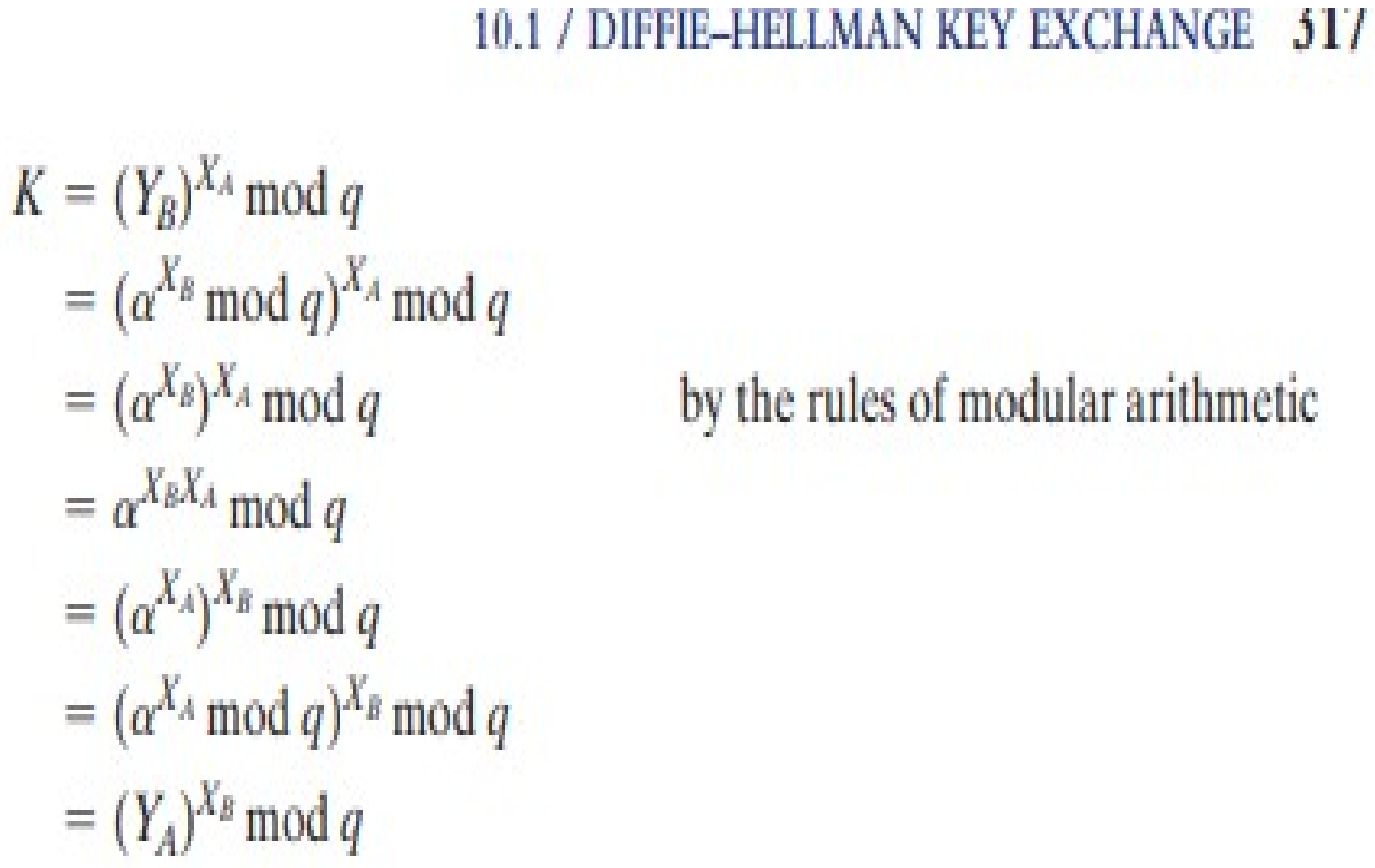
Alice and Bob share a prime number q and an integer α, such that $\alpha < q$ and α is a primitive root of q

Bob generates a private key X_B such that $X_B < q$

Bob calculates a public key $Y_B = \alpha^{X_B} \bmod q$

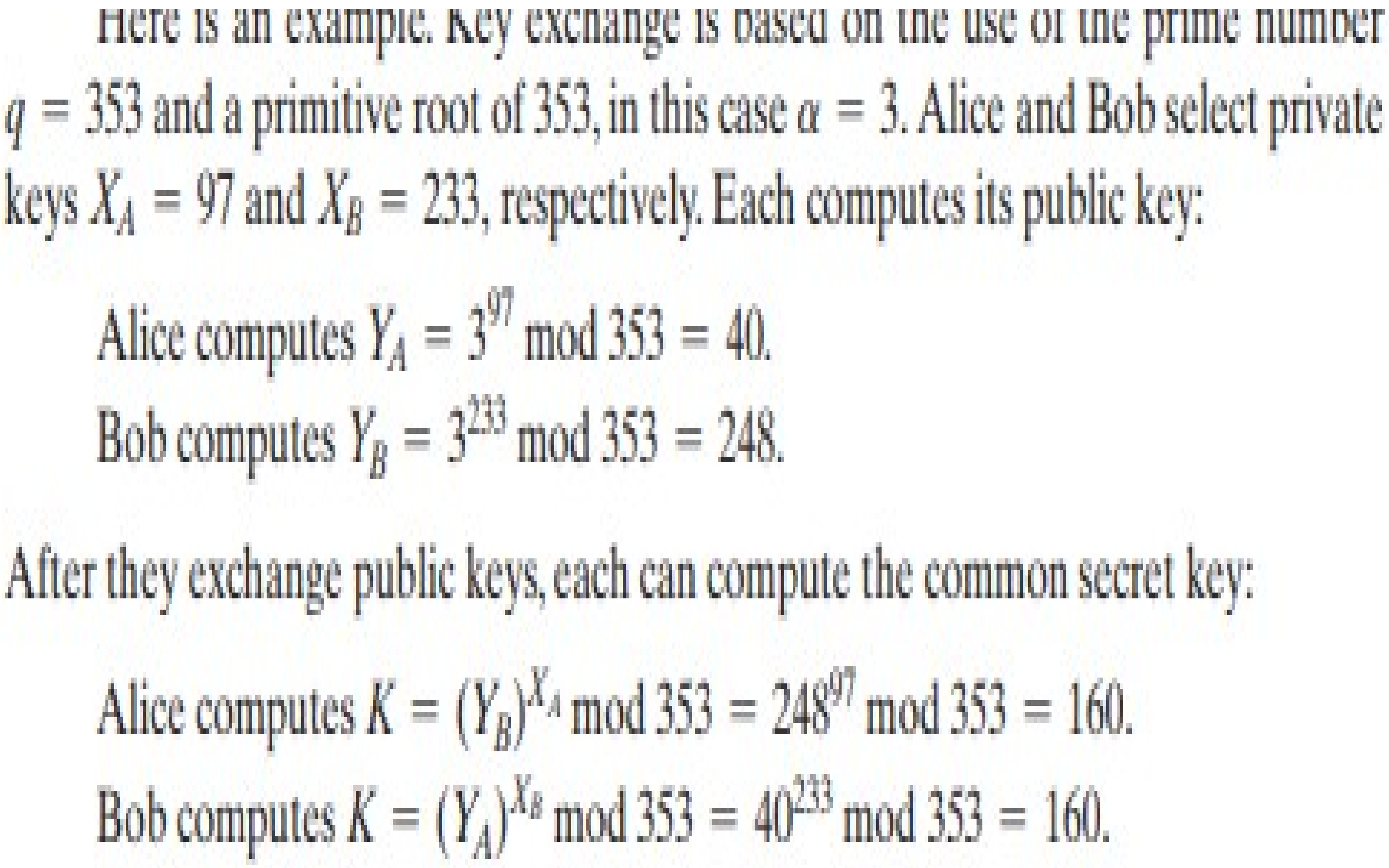
Bob receives Alice's public key Y_A in plaintext

Bob calculates shared secret key $K = (Y_A)^{X_B} \bmod q$



10.1 / DIFFIE–HELLMAN KEY EXCHANGE 317

$$
\begin{aligned}
K &= (Y_B)^{X_A} \bmod q \\
&= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\
&= (\alpha^{X_B})^{X_A} \bmod q && \text{by the rules of modular arithmetic} \\
&= \alpha^{X_B X_A} \bmod q \\
&= (\alpha^{X_A})^{X_B} \bmod q \\
&= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\
&= (Y_A)^{X_B} \bmod q
\end{aligned}
$$



Here is an example. Key exchange is based on the use of the prime number $q = 353$ and a primitive root of 353, in this case $\alpha = 3$. Alice and Bob select private keys $X_A = 97$ and $X_B = 233$, respectively. Each computes its public key:

Alice computes $Y_A = 3^{97} \bmod 353 = 40$.

Bob computes $Y_B = 3^{233} \bmod 353 = 248$.

After they exchange public keys, each can compute the common secret key:

Alice computes $K = (Y_B)^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160$.

Bob computes $K = (Y_A)^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160$.

Man-in-the-Middle Attack

The protocol depicted in Figure 10.1 is insecure against a **man-in-the-middle attack**. Suppose Alice and Bob wish to exchange keys, and Darth is the adversary. The attack proceeds as follows (Figure 10.2).

1. Darth prepares for the attack by generating two random private keys X_{D1} and X_{D2} and then computing the 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 $K2 = (Y_A)^{X_{D2}} \bmod q$.
4. Bob receives Y_{D1} and calculates $K1 = (Y_{D1})^{X_B} \bmod q$.
5. Bob transmits Y_B to Alice.
6. Darth intercepts Y_B and transmits Y_{D2} to Alice. Darth calculates $K1 = (Y_B)^{X_{D1}} \bmod q$.
7. Alice receives Y_{D2} and calculates $K2 = (Y_{D2})^{X_A} \bmod q$.

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key $K1$ and Alice and Darth share secret key $K2$. All future communication between Bob and Alice is compromised in the following way.

1. Alice sends an encrypted message M: E($K2$, M).
2. Darth intercepts the encrypted message and decrypts it to recover M.
3. Darth sends Bob E($K1$, M) or E($K1$, M'), where M' is any message. In the first case, Darth simply wants to eavesdrop on the communication without altering it. In the second case, Darth wants to modify the message going to Bob.

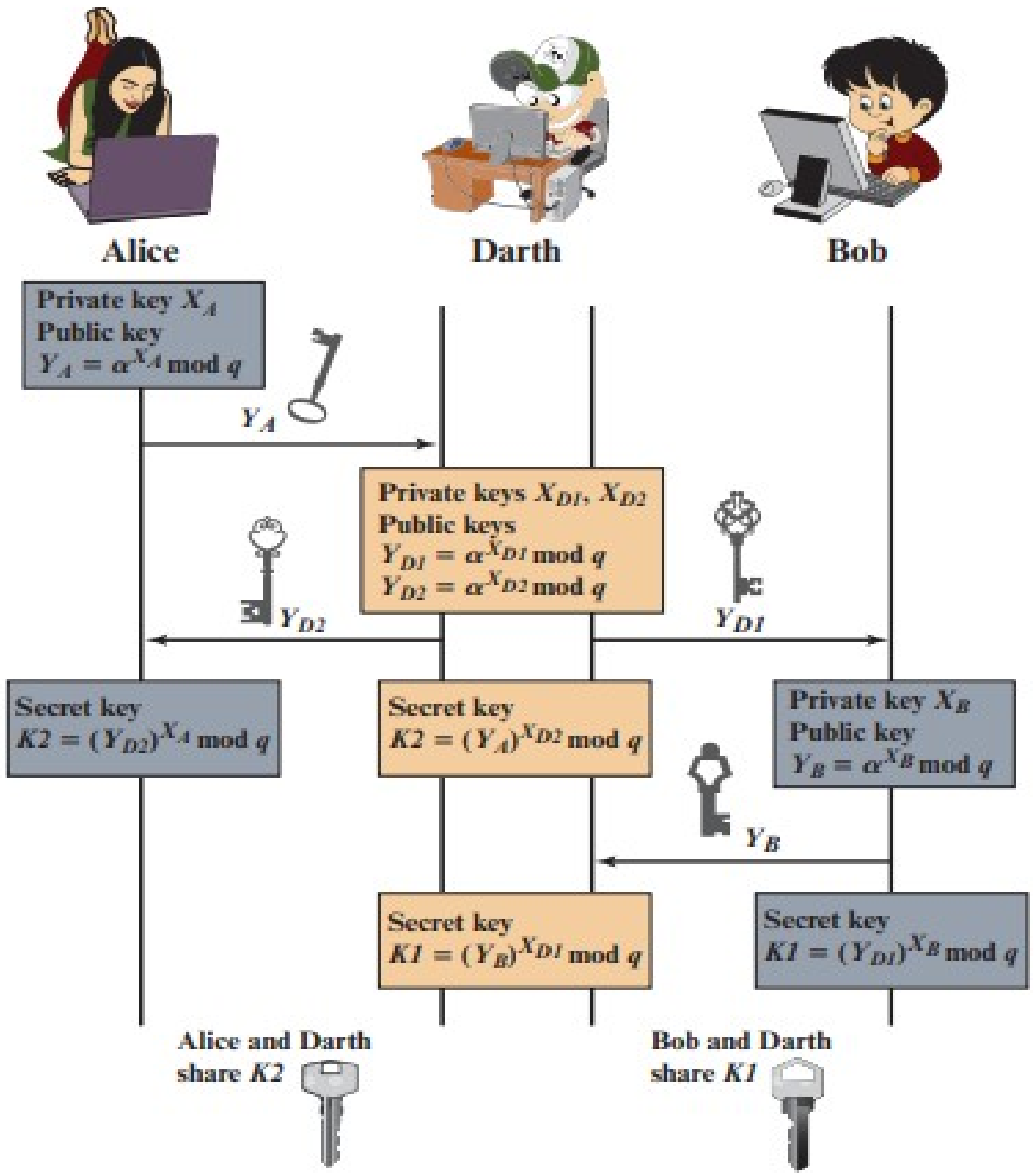
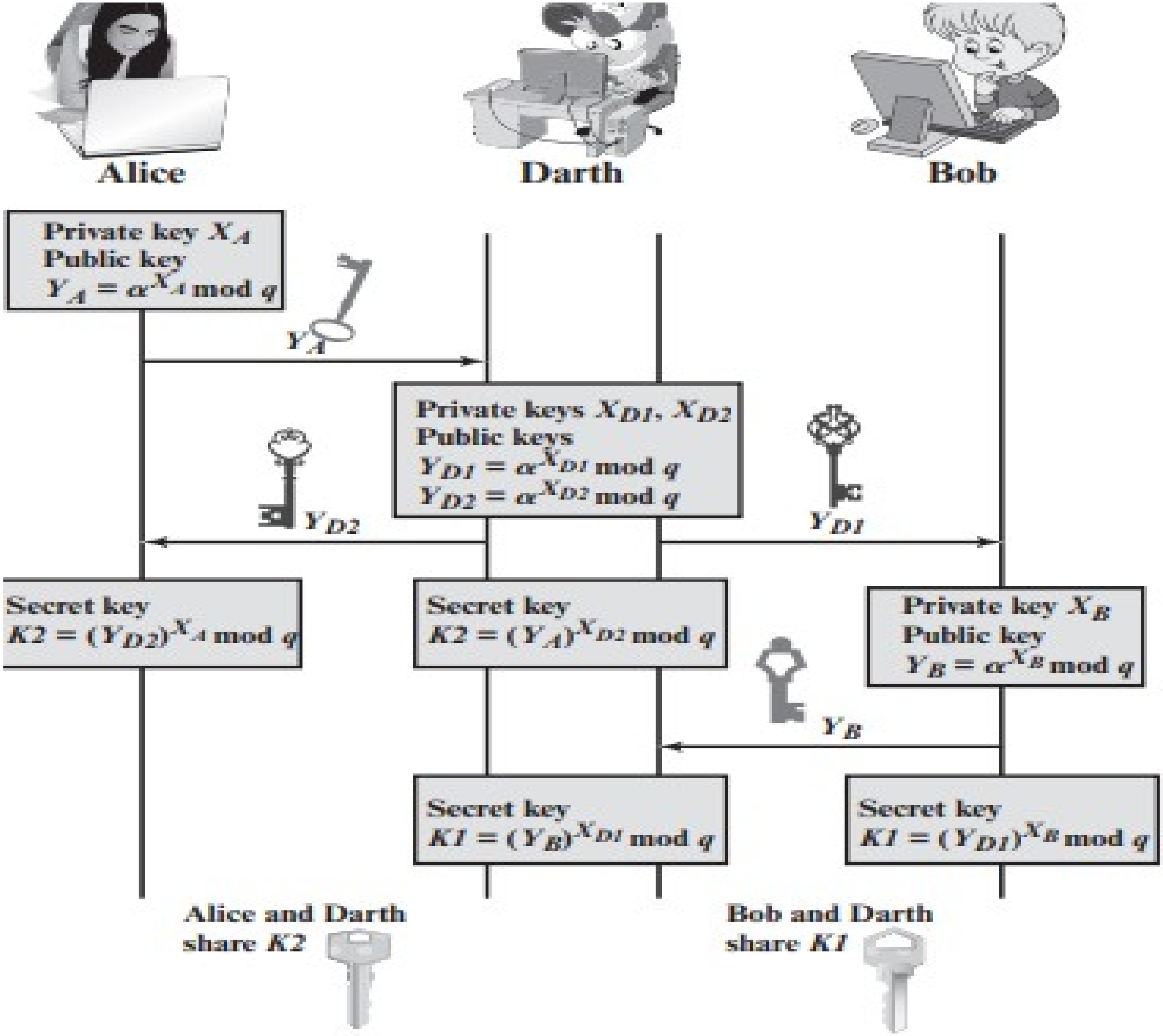

Alice
Darth
Bob
Private key X_A
Public key
$Y_A = \alpha^{X_A} \bmod q$
Y_A
Private keys X_{D1}, X_{D2}
Public keys
$Y_{D1} = \alpha^{X_{D1}} \bmod q$
$Y_{D2} = \alpha^{X_{D2}} \bmod q$
Y_{D2}
Y_{D1}
Secret key
$K2 = (Y_{D2})^{X_A} \bmod q$
Secret key
$K2 = (Y_A)^{X_{D2}} \bmod q$
Private key X_B
Public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

Figure 10.2 Man-in-the-Middle Attack


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 $K1$

igure 10.2 Man-in-the-Middle Attack

10.1 Alice and Bob use the Diffie–Hellman key exchange technique with a common prime $q = 157$ and a primitive root $\alpha = 5$.

a. If Alice has a private key $X_A = 15$, find her public key Y_A.
b. If Bob has a private key $X_B = 27$, find his public key Y_B.
c. What is the shared secret key between Alice and Bob?

10.2 Alice and Bob use the Diffie–Hellman key exchange technique with a common prime $q = 23$ and a primitive root $\alpha = 5$.

a. If Bob has a public key $Y_B = 10$, what is Bob's private key X_B?
b. If Alice has a public key $Y_A = 8$, what is the shared key K with Bob?
c. Show that 5 is a primitive root of 23.

10.3 In the Diffie–Hellman protocol, each participant selects a secret number x and sends the other participant $\alpha^x \bmod q$ for some public number α. What would happen if the participants sent each other x^α for some public number α instead? Give at least one method Alice and Bob could use to agree on a key. Can Darth break your system without finding the secret numbers? Can Darth find the secret numbers?

10.4 This problem illustrates the point that the Diffie–Hellman protocol is not secure without the step where you take the modulus; i.e. the "Indiscrete Log Problem" is not a hard problem! You are Darth and have captured Alice and Bob and imprisoned them. You overhear the following dialog.

Bob: Oh, let's not bother with the prime in the Diffie–Hellman protocol, it will make things easier.

Alice: Okay, but we still need a base α to raise things to. How about $\alpha = 3$?

Bob: All right, then my result is 27.

Alice: And mine is 243.

What is Bob's private key X_B and Alice's private key X_A? What is their secret combined key? (Don't forget to show your work.)

Global Public Elements

$E_q(a, b)$	elliptic curve with parameters a, b, and q, where q is a prime or an integer of the form 2^m
G	point on elliptic curve whose order is large value n

User A Key Generation

Select private n_A	$n_A < n$
Calculate public P_A	$P_A = n_A \times G$

User B Key Generation

Select private n_B	$n_B < n$
Calculate public P_B	$P_B = n_B \times G$

Calculation of Secret Key by User A

$K = n_A \times P_B$

Calculation of Secret Key by User B

$K = n_B \times P_A$

Figure 10.7 ECC Diffie–Hellman Key Exchange

- The cryptosystem parameters are E11(1, 6) and G = (2,7). B’s private key is nB = 7.
- Find B’s public key PB.
- A wishes to encrypt the message Pm = (10, 7) and chooses the random value k = 3. Determine the cipher text Cm.