Unit III. Asymmetric Ciphers
Diffie-Hellman, Elgamal

Er. Kobid Karkee
Himalaya College of Engineering

Diffie-Hellman Key Exchange

Diffie-Hellman Key Exchange

- first public-key type scheme proposed
- by Diffie & Hellman in 1976 along with the exposition of public key concepts
  - note: now know that James Ellis (UK CESG) secretly proposed the concept in 1970
- is a practical method for public exchange of a secret key
- used in several commercial products

Diffie-Hellman Key Exchange

- a public-key distribution scheme
  - cannot be used to exchange an arbitrary message
  - rather it can establish a common key
  - known only to the two participants
- value of key depends on the participants (and their private and public key information)
- based on exponentiation in a finite (Galois) field (modulo a prime or a polynomial) - easy
- security relies on the difficulty of computing discrete logarithms (similar to factoring) – hard

Diffie-Hellman Setup

- all users agree on global parameters:
  - large prime integer or polynomial q
  - α a **primitive root** mod q
- each user (e.g. A) generates their key
  - chooses a secret key (number): $x_A < q$
  - compute their **public key**: $y_A = \alpha^{x_A} \bmod q$
- each user makes public that key y_A
- shared session key for users A & B is K_{AB}:

$K_{AB} = \alpha^{x_A \cdot x_B} \bmod q$

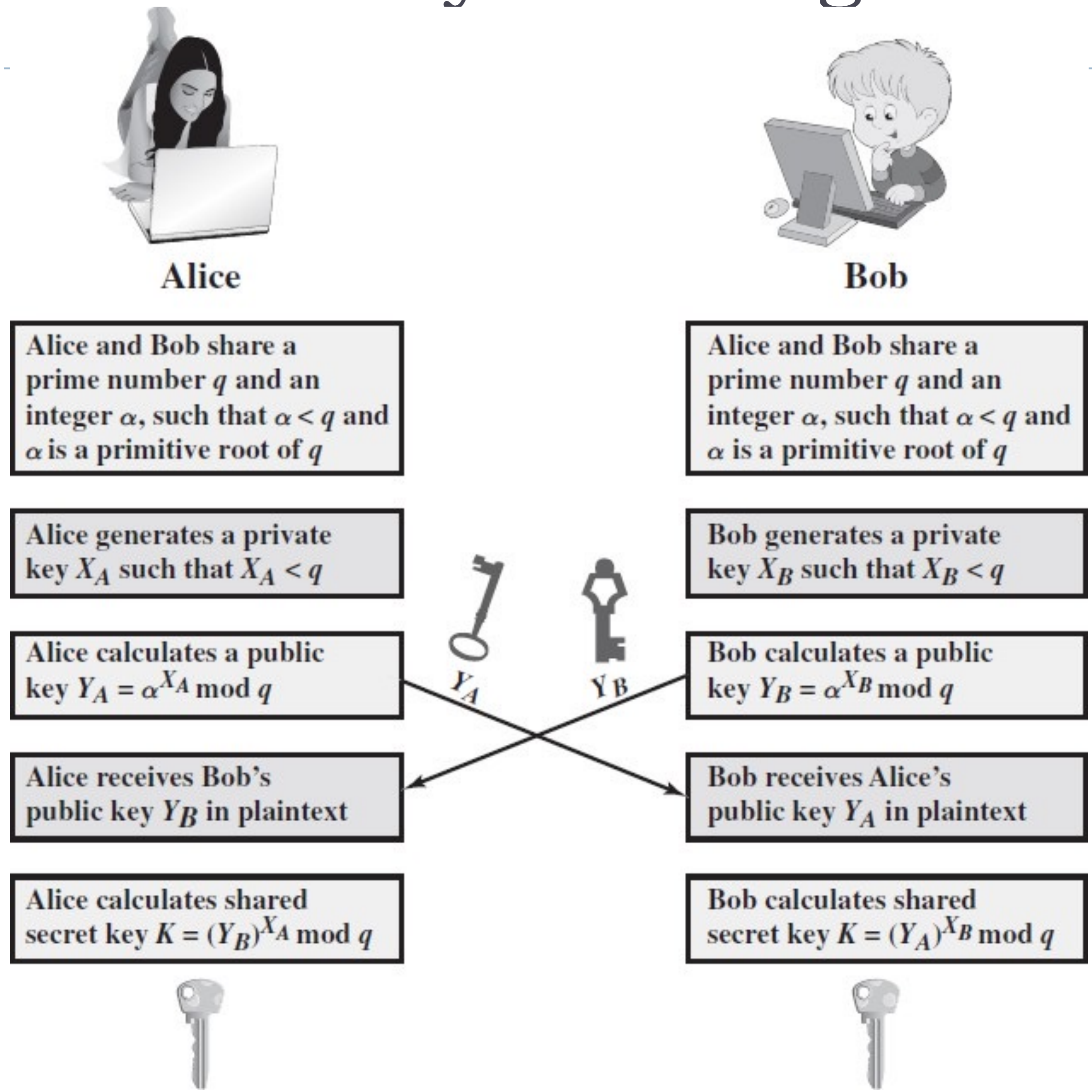
$= y_A^{x_B} \bmod q$ (which **B** can compute)

$= y_B^{x_A} \bmod q$ (which **A** can compute)

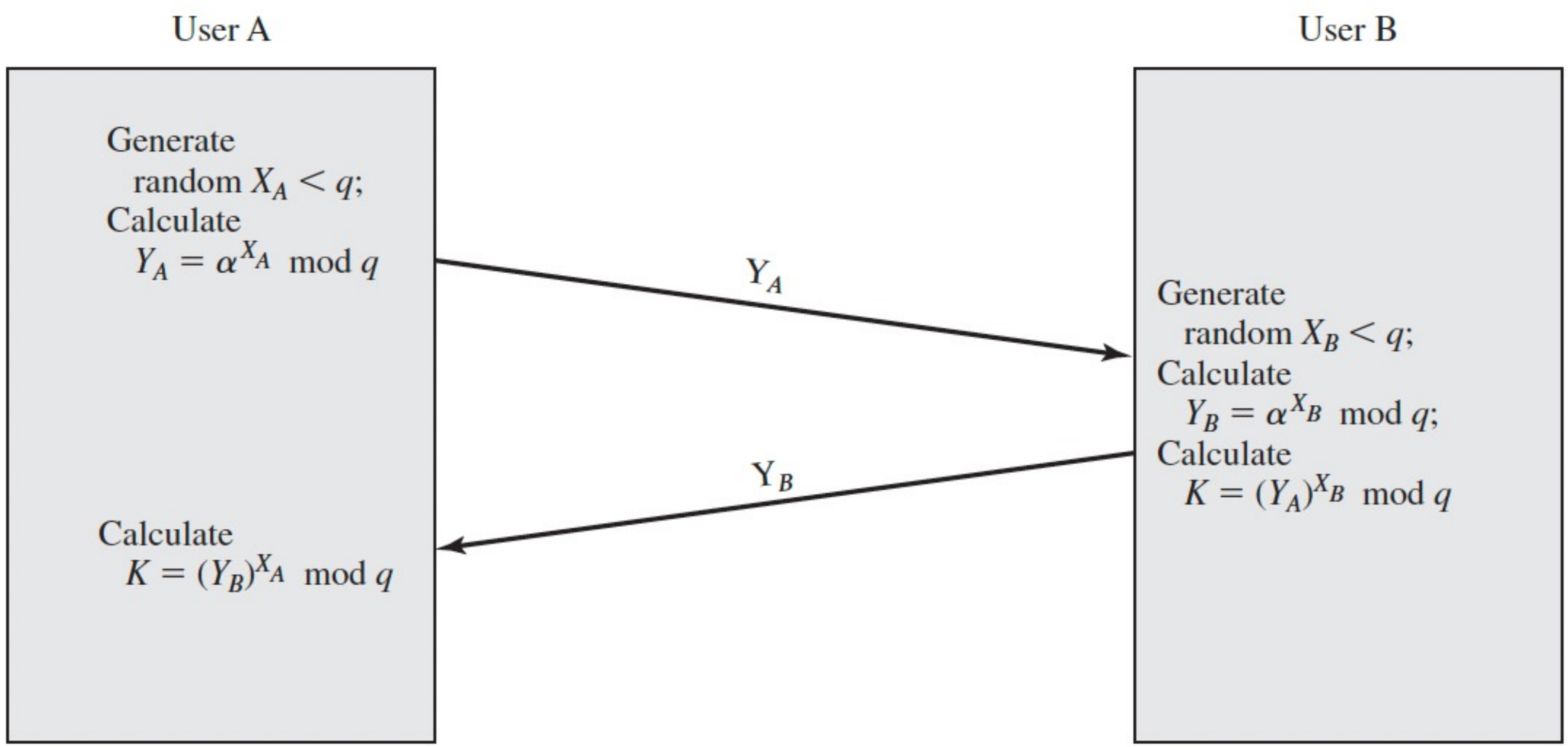
Diffie-Hellman Key Exchange

- K_{AB} is used as session key in private-key encryption scheme between Alice and Bob
- if Alice and Bob subsequently communicate, they will have the same key as before, unless they choose new public-keys
- attacker needs an x, must solve discrete log (which is hard)

Diffie-Hellman Key Exchange



Difie-Hellman Key Exchange



Diffie-Hellman Example

- users Alice & Bob who wish to swap keys:
- agree on prime $q=353$ and $\alpha=3$
- select random secret keys:
  - A chooses $x_A=97$, B chooses $x_B=233$
- compute public keys:
  - $y_A=\mathbf{3}^{97} \bmod 353 = 40$ (Alice)
  - $y_B=\mathbf{3}^{233} \bmod 353 = 248$ (Bob)
- compute shared session key as:

$K_{AB}= y_B^{x_A} \bmod 353 = \mathbf{248}^{97} = 160$ (Alice)

$K_{AB}= y_A^{x_B} \bmod 353 = \mathbf{40}^{233} = 160$ (Bob)

Man-in-the-Middle Attack

- The Diffie-Helman Key Exchange protocol depicted in previously 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:
  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 $K2 = Y_{D2}^{X_A} \bmod q$.

Man-in-the-Middle Attack

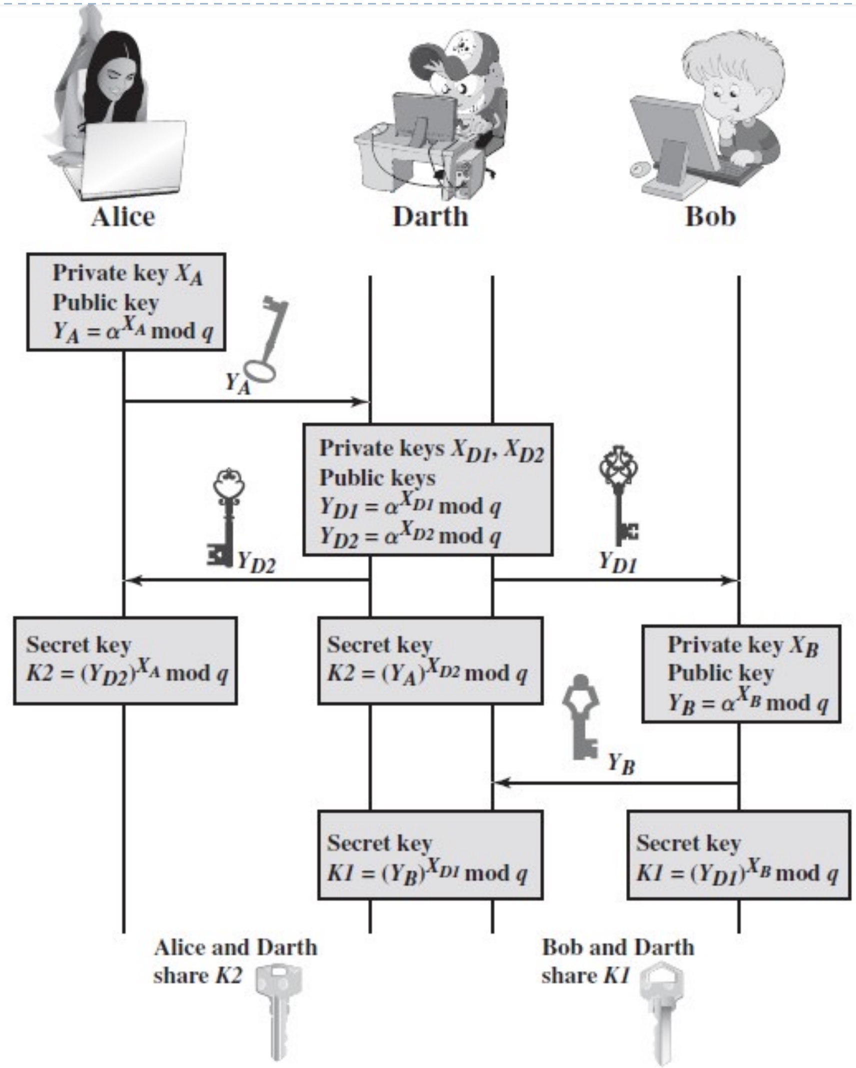
- At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key K1 and Alice and Darth share secret key K2.
- All future communication between Bob and Alice is compromised in the following way:

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.

Man-in-the-Middle Attack



- The key exchange protocol is vulnerable to such an attack because it does not authenticate the participants.
- This vulnerability can be overcome with the use of digital signatures and public-key certificates; these topics are explored in later chapters.

Elgamal Cryptographic System

Elgamal Cryptographic System

- In 1984, T. ElGamal announced a public-key scheme based on discrete logarithms, closely related to the Diffie-Hellman technique
- The ElGamal cryptosystem is used in some form in several standards including the digital signature standard (DSS).
- ElGamal encryption is a public-key cryptosystem.

Elgamal Cryptographic System

- It uses asymmetric key encryption for communicating between two parties and encrypting the message.
- This cryptosystem is based on the difficulty of finding discrete logarithm in a cyclic group.
- As with Diffie-Hellman, the global elements of Elgamal are a prime number q and α, which is a primitive root of q.

Elgamal Cryptographic System

- User A generates a private/public key pair as follows:
  1. Generate a random integer X_A, such that $1 < X_A < q - 1$.
  2. Compute $Y_A = \alpha^{X_A} \, mod \; q$.
  3. A's private key is X_A; A's public key is $\{q, \alpha, Y_A\}$.
- Any user B that has access to A's public key can encrypt a message as follows:
  1. Represent the message as an integer M in the range $1 \le M \le q - 1$. Longer messages are sent as a sequence of blocks, with each block being an integer less than q.
  2. Choose a random integer k such that $1 < k < q - 1$.
  3. Compute a one-time key $K = Y_A^k \, mod \; q$.
  4. Encrypt M as the pair of integers $(C1, C2)$, where

$$C_1 = \alpha^k \, mod \; q; \quad C_2 = KM \, mod \; q$$

Elgamal Cryptographic System

- User A recovers the plaintext as follows:
  1. Recover the key by computing $K = (C_1)^{X_A} \; mod \; q$.
  2. Compute $M = (C_2 K^{-1}) \; mod \; q$.

Elgamal Cryptographic System

- Let us demonstrate why the ElGamal scheme works.
- First, we show how K is recovered by the decryption process:

$K = Y_A^k \; mod \; q$ — K is defined during the encryption process

$K = (\alpha^{X_A} \; mod \; q\,)^k \; mod \; q$ — substitute using $Y_A = \alpha^{X_A} \; mod \; q$

$K = \alpha^{kX_A} \; mod \; q$ — by the rules of modular arithmetic

$K = (C_1)^{X_A} \; mod \; q$ — substitute using $C_1 = \alpha^k \; mod \; q$

Next, using K, we recover the plaintext as

$C_2 = KM \; mod \; q$

$(C_2K^{-1}) mod \; q = KMK^{-1} \; mod \; q = M \; mod \; q = M$

Elgamal Cryptographic System

We can restate the ElGamal process as follows:

1. Bob generates a random integer k.
2. Bob generates a one-time key K using Alice's public-key components Y_A, q, and k.
3. Bob encrypts k using the public-key component α, yielding C_1. C_1 provides sufficient information for Alice to recover K.
4. Bob encrypts the plaintext message M using K.
5. Alice recovers K from C_1 using her private key.
6. Alice uses K^{-1} to recover the plaintext message from C_2.

Global Public Elements	
q	prime number
α	$\alpha < q$ and α a primitive root of q

Key Generation by Alice	
Select private X_A	$X_A < q - 1$
Calculate Y_A	$Y_A = \alpha^{XA} \bmod q$
Public key	$PU = \{q, \alpha, Y_A\}$
Private key	X_A

Encryption by Bob with Alice's Public Key	
Plaintext:	$M < q$
Select random integer k	$k < q$
Calculate K	$K = (Y_A)^k \bmod q$
Calculate C_1	$C_1 = \alpha^k \bmod q$
Calculate C_2	$C_2 = KM \bmod q$
Ciphertext:	(C_1, C_2)

Decryption by Alice with Alice's Private Key	
Ciphertext:	(C_1, C_2)
Calculate K	$K = (C_1)^{XA} \bmod q$
Plaintext:	$M = (C_2 K^{-1}) \bmod q$

Elgamal Cryptographic System Example

- Example: Let us start with the prime field GF(19); that is, q = 19. It has primitive roots {2, 3, 10, 13, 14, 15}. We choose α = 10.

Alice generates a key pair as follows:

1. Alice chooses $X_A = 5$.
2. Then $Y_A = \alpha^{X_A} \bmod q = \alpha^5 \bmod 19 = 3$ (see Table 8.3).
3. Alice's private key is 5; Alice's pubic key is $\{q, \alpha, Y_A\} = \{19, 10, 3\}$.

Elgamal Cryptographic System

- Suppose Bob wants to send the message with the value M = 17. Then,

1. Bob chooses $k = 6$.
2. Then $K = (Y_A)^k \bmod q = 3^6 \bmod 19 = 729 \bmod 19 = 7$.
3. So

   $C_1 = \alpha^k \bmod q = \alpha^6 \bmod 19 = 11$

   $C_2 = KM \bmod q = 7 \times 17 \bmod 19 = 119 \bmod 19 = 5$
4. Bob sends the ciphertext $(11, 5)$.

- For decryption:

1. Alice calculates $K = (C_1)^{X_A} \bmod q = 11^5 \bmod 19 = 161051 \bmod 19 = 7$.
2. Then K^{-1} in GF(19) is $7^{-1} \bmod 19 = 11$.
3. Finally, $M = (C_2K^{-1}) \bmod q = 5 \times 11 \bmod 19 = 55 \bmod 19 = 17$.

Elgamal Cryptographic System

- If a message must be broken up into blocks and sent as a sequence of encrypted blocks, a unique value of *k* should be used for each block.
- If *k* is used for more than one block, knowledge of one block of the message enables the user to compute other blocks as follows:

Elgamal Cryptographic System

- Let

$$C_{1,1} = \alpha^k \bmod q;\ C_{2,1} = KM_1 \bmod q$$

$$C_{1,2} = \alpha^k \bmod q;\ C_{2,2} = KM_2 \bmod q$$

Then,

$$\frac{C_{2,1}}{C_{2,2}} = \frac{KM_1 \bmod q}{KM_2 \bmod q} = \frac{M_1 \bmod q}{M_2 \bmod q}$$

If M_1 is known, then M_2 is easily computed as

$$M_2 = (C_{2,1})^{-1}\, C_{2,2}\, M_1 \bmod q$$

Security of Elgamal Cryptographic System

- The security of Elgamal is based on the difficulty of computing discrete logarithms.
- To recover A's private key, an adversary would have to compute
  $X_A = \text{dlog}_{\alpha,q}(Y_A)$.
- Alternatively, to recover the one-time key K, an adversary would have to determine the random number k, and this would require computing the discrete logarithm
  $k = \text{dlog}_{\alpha,q}(C_1)$.
- These calculations are regarded as infeasible if p is at least 300 decimal digits and q - 1 has at least one "large" prime factor.