Other Public-key Cryptosystems

Erzhena Tcydenova
Department of Computer Science & Engineering
Seoul National University of Science & Technology
CIS (Cryptography and Information Security) Lab

Contents

Diffie-Hellman key exchange

Cryptography Information Security Laboratory

- The first published public-key algorithm by Diffie and Hellman that defined public-key cryptography.
- It is generally referred to as Diffie-Hellman key exchange.
- The purpose of the algorithm is **to enable two users to securely exchange a key** that can then be used for subsequent encryption of messages.
- The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing **discrete logarithms.**

Diffie-Hellman key exchange

Global Public Elements	
q	prime number
α	$\alpha < q$ and α a 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	$Y_B = \alpha^{XB} \bmod q$

Calculation of Secret Key by User A

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

Calculation of Secret Key by User B

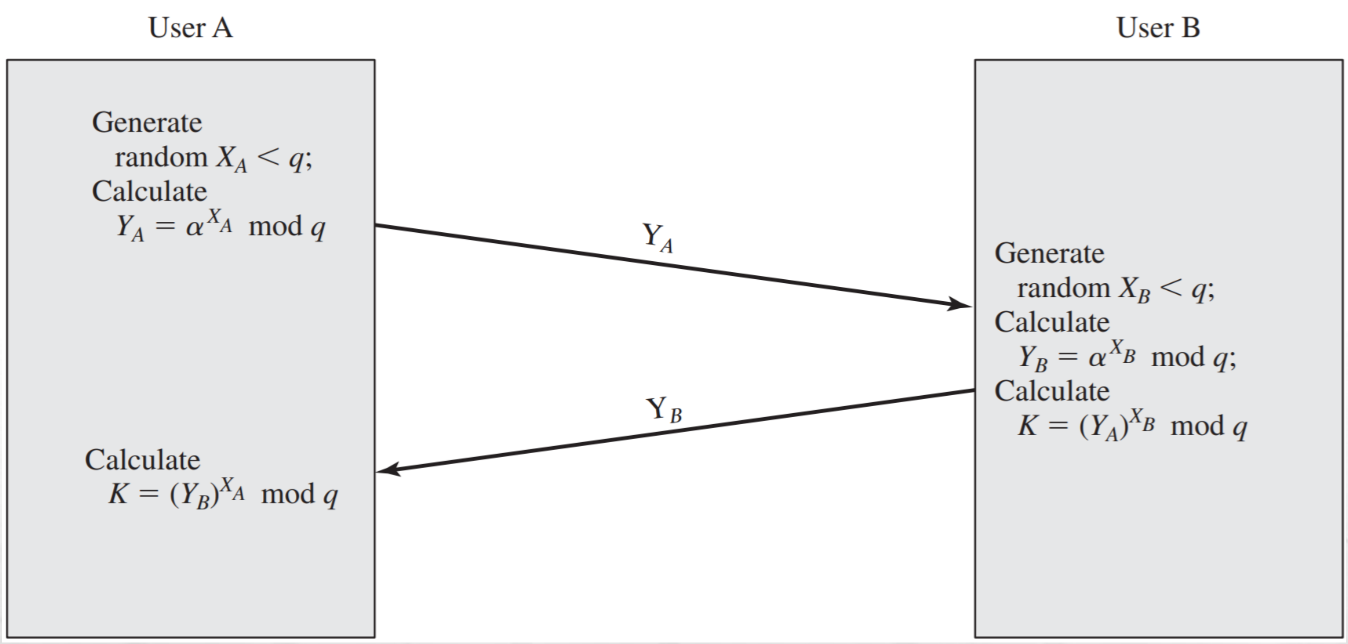
$K = (Y_A)^{XB} \bmod q$

Diffie-Hellman key exchange

- It is relatively **easy to calculate exponentials modulo a prime**, it is very **difficult to calculate discrete logarithms**.
- For large primes, the latter task is considered infeasible.
- **Example:**
  - Key exchange is based on the use of the prime number $\boldsymbol{q = 353}$, primitive root of 353: $\boldsymbol{\alpha = 3}$
  - A and B select secret keys $\boldsymbol{X_A = 97}$ and $\boldsymbol{X_B = 233}$
  - A computes $Y_A = 3^{97} \bmod 353 = 40$
  - B computes $Y_B = 3^{233} \bmod 353 = 248$
  - Then, they exchange public keys
  - A computes $K = (Y_B)^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160$
  - B computes $K = (Y_A)^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160$

Diffie–Hellman key exchange

- Suppose that user A wishes to set up a connection with user B and use a secret key to encrypt messages on that connection.
- User A can generate private key, calculate public key and send that to user B.
- User B responds by generating private and public keys, and sending to user A.
- The necessary public values q and α would need to be known ahead of time.



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 a number of standards including the digital signature standard (DSS), and the S/MIME e-mail standard.

ElGamal Cryptographic System

Cryptography Information Security Laboratory

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

- As with Diffie-Hellman, the global elements of ElGamal are a prime number $\boldsymbol{q}$ and $\boldsymbol{\alpha}$, which is a primitive root of $\boldsymbol{q}$.
- **Example:**
  - $\boldsymbol{q = 19, \alpha = 10}$
  - Alice chooses $\boldsymbol{X_A = 5}$
  - Then $Y_A = \alpha^{X_A} \bmod q = \alpha^5 \bmod 19 = 3$
  - Alice′s private key is 5, public key – {19, 10, 3}
  - Suppose M = 17
  - Bob chooses $\boldsymbol{k = 6}$
  - Then, $K = (Y_A)^k \bmod q = 3^6 \bmod 19 = 729 \bmod 19 = 7$
  - So, $C_1 = \alpha^k \bmod q = a^6 \bmod 19 = 11$
  - $C_2 = \mathrm{KM} \bmod q = 7 \text{ x } 17 \bmod 19 = 119 \bmod 19 = 5$
  - Bob sends ciphertext (11, 5)
  - Alice calculates $K = (C_1)^{X_A}$ mod $q = 11^5 \bmod 19 = 161051 \bmod 19 = 7$
  - Then K^{-1} is 7^{-1} mod 19 = 11
  - Finally, M = $(C_2 \quad K^{-1})$ mod $q = 5 \text{ x } 11 \bmod 19 = 55 \bmod 19 =$ 17

Elliptic Curve Cryptography

- Most of the products and standards that use public-key cryptography for encryption and digital signatures use RSA.
- The key length for secure RSA use has increased over recent years, and this has put a heavier processing load on applications using RSA.
- A competing system challenges RSA: elliptic curve cryptography (ECC).
- ECC is showing up in standardization efforts, including the IEEE P1363 Standard for Public-Key Cryptography.
- The principal attraction of ECC, compared to RSA, is that it appears to offer equal security for a far smaller key size, thereby reducing processing overhead.

Elliptic Curve Cryptography

- The addition operation in ECC is the counterpart of modular multiplication in RSA, and multiple addition is the counterpart of modular exponentiation.
- To form a cryptographic system using elliptic curves, we need to find a "hard problem" corresponding to factoring the product of two primes or taking the discrete logarithm.
- Consider the equation $Q = kP$ where $Q, P \in E_p(a, b)$ and $\mathrm{k} < \mathrm{p}$.
- It is relatively easy to calculate Q given k and P, but it is relatively hard to determine k given Q and P.
- This is called the discrete logarithm problem for elliptic curves.

Elliptic Curve Cryptography

Cryptography Information Security Laboratory

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

$K = n_B \times P_A$

Elliptic Curve Cryptography

- The security of ECC depends on how difficult it is to determine k given kP and P.
- This is referred to as the elliptic curve logarithm problem.

Symmetric Scheme (key size in bits)	ECC-Based Scheme (size of n in bits)	RSA/DSA (modulus size in bits)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

Thank you