DLP based Cryptography

Prof. Ashok K Bhateja

IIT Delhi

Discrete logarithm

- Let G be a finite cyclic group of order n. Let α be a generator of G, and let $\beta \in G$. The discrete logarithm of β to the base α, denoted $\log_\alpha \beta$, is the unique integer x, $0 \le x \le n - 1$, s.t. $\beta = \alpha^x$.
- Ex: $\alpha = 3$ and $\beta = 19683$, since $3^9 = 19683$

  $\therefore \ \log_3 19683 = 9$

Discrete logarithm problem

- Definition DLP: Example: Given a prime p, a generator α of Z_p^*, and an element $\beta \in Z_p^*$, find the integer x, $0 \le x \le p - 2$, such that

$$\alpha^x \equiv \beta \pmod{p}.$$

- Let $p = 97$. Z_{97}^* is a cyclic group of order 96.

  A generator of Z_{97}^* is $\alpha = 5$.

  Since $5^{32} \equiv 35 \pmod{97}$ therefore $\log_5 35 = 32$ in Z_{97}^* .

- Definition GDLP: Given a finite cyclic group G of order n, a generator α of G, and an element $\beta \in G$, find the integer x, $0 \le x \le n - 1$, such that $\alpha^x \equiv \beta$.

ElGamal public-key cryptosystem

- The security of the ElGamal public-key encryption scheme is based on the intractability of the discrete logarithm problem.
- It has the advantage the same plaintext gives a different ciphertext (with near certainty) each time it is encrypted.
- ElGamal has the disadvantage that the ciphertext is twice as long as the plaintext.

Key generation for ElGamal public-key encryption

- Each entity creates a public key and a corresponding private key
- Generate a large random prime p and a generator α of the multiplicative group Z_p^* of the integers modulo p.
- Select a random integer d, $1 \leq d \leq p - 2$, and compute $\beta = \alpha^d \bmod p$
- A's public key is (p, α, β)
- A's private key is d.

ElGamal Encryption & Decryption

- Encryption:
  - To encrypts a message m $(0 \leq m \leq p)$
  - choose a random integer k , $1 \leq k \leq p$ - 2
  - find $r \equiv \alpha^k \bmod p$ & $t \equiv \beta^k \cdot m \bmod p$

  The encrypted message $c = (r, t)$
- Decryption:
  - Compute $r^{p-1-d} \pmod{p}$
  - Compute $m = t \cdot r^{p-1-d} \pmod{p}$

ElGamal Cryptosystem: Justification

$\beta = \alpha^{d} \bmod p$

$r \equiv \alpha^{k} \bmod p \quad \& \quad t \equiv \beta^{k} \cdot m \bmod p$

Claim: $m = t \cdot r^{p-1-d} \pmod{p}$

$$
\begin{aligned}
t \cdot r^{p-1-d} \pmod{p} &= \alpha^{kd}\, m\, \alpha^{k(p-1-d)} \bmod p \\
&= \alpha^{k(p-1)}\, m \bmod p \\
&= (\alpha^{p-1})^{k}\, m \bmod p \\
&\equiv m \bmod p \qquad \text{using Fermat's theorem}
\end{aligned}
$$

Example

Entity A selects prime p = 107, generator α = 2, and private key d = 67

Compute $\beta = \alpha^{d} \bmod p = 2^{67} \pmod{107} \equiv 94$.

A's public key: $(p, \alpha, \beta) = (107, 2, 94)$

A's private key is d = 67.

Encryption: To encrypt a message m = 66

B selects a random integer k = 45

Find $(r, t) = (\alpha^{k} \bmod p\,,\ \beta^{k} m)$

$$\equiv (2^{45} \bmod 107,\ 94^{45} \cdot 66 \bmod 107) \equiv (28, 9)$$

B sends the encrypted message (28, 9) to A.

Example

A receives the message $(r, t) = (28, 9)$

Decryption (by A):

Compute $r^{(p-1-d)} \pmod{p} = 28^{107-1-67} \bmod 107$

$= 43$

Compute $m = t \cdot r^{p-1-d} \pmod{p} = 9 \times 43 \bmod 107$

$= 66$

Security of ElGamal Encryption

- An eavesdropper knows p, α, β, r, t where $\beta \equiv \alpha^{d} \bmod p$ and $r \equiv \alpha^{k} \bmod p$.
- Determining m from (r, t) is equivalent to computing $\alpha^{dk} \bmod p$, since $t \equiv \beta^{k} \cdot m \bmod p$.
- Here, m is masked by the quantity $\alpha^{dk} \bmod p$.
- Both d, k are unknown to the attacker.
- So, the ability to solve the Discrete Logarithm problem lets the eavesdropper break ElGamal encryption.
- Practically, we require p to be of size ≥ 1024 bits for achieving a good level of security.

Common System-wide parameters

- All entities may use the same prime p and generator α, in which case p and α need not be published as part of the public key.
- Advantage:
  - Size of public keys will be small
  - Exponentiation can then be expedited via precomputations
- Disadvantage:
  - Precomputation of a database of factor base logarithms
    - requirement of Index Calculus algorithm
  - will compromise the secrecy of all private keys derived using p.

Fixed-base exponentiation algorithms

- To find α^e, write exponent e in a base-b representation, i.e.

  $$e = e_0 b^0 + e_1 b^1 + e_2 b^2 + \ldots + e_t b^t$$

  e is a $(t+1)$ - digit base b integer with $b \geq 2$
- The look-up table of $\alpha_i = \alpha^{b^i}$, $i = 0, \ldots, t$ precomputed
- Example: Compute α^{862}

  Base $b = 4$, $e = (862)_{10} = (31132)_4$

  $$= 2 + 3 \cdot 4^1 + 1 \cdot 4^2 + 1 \cdot 4^3 + 3 \cdot 4^4$$
- The needed precomputations are $\alpha^{4^0}, \alpha^{4^1}, \alpha^{4^2}, \alpha^{4^4}$

Diffie-Hellman Key Exchange

- Discovered by Whitfield Diffie and Martin Hellman in 1976 and published in "New Directions in Cryptography."
- Diffie-Hellman key agreement provided the first practical solution to the key distribution problem.
- The protocol allows two users to exchange a secret key over an insecure medium without any prior secrets.
- Security – Intractability of Discrete Logarithm problem
- This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Introduction: Diffie-Hellman Key Exchange

- Security of transmission is critical for many network and Internet applications
- Requires users to share information in a way that others can't decipher the flow of information

"It is insufficient to protect ourselves with laws; we need to protect ourselves with mathematics."

-Bruce Schneier

Introduction: Diffie-Hellman Key Exchange

- Let $Z_p{}^*$ be a cyclic group, with a generator $\alpha \in Z_p{}^*$
- p and α are both publicly available numbers
  - p is at least 512 bits
- Users pick private values a and b may be randomly.

Diffie-Hellman Key Exchange Protocol

- Alice and Bob agree upon and make public two numbers α and p, where p is a prime and α is a generator of Z_p^*.

Alice		Bob
choose a random number a		
compute $u = \alpha^a \pmod{p}$	$\xrightarrow{\quad u \quad}$	u
		choose a random number b
v	$\xleftarrow{\quad v \quad}$	compute $v = \alpha^b \pmod{p}$
Compute v^a		compute u^b
i.e. $v^a = (\alpha^b)^a \pmod{p}$		$u^b = (\alpha^a)^b \pmod{p}$
The key $k = \alpha^{ab} \pmod{p}$		key $k = \alpha^{ab} \pmod{p}$

Example: Diffie-Hellman Key Exchange Protocol

- Alice and Bob get public numbers
  - $p = 23, \quad \alpha = 9$
  - Alice private number $a = 4$
  - Bob private number $b = 3$
- Alice and Bob compute public values
  - $u \;=\; 9^4 \bmod 23 = \; 6561 \bmod 23 \;=\; 6$
  - $v \;=\; 9^3 \bmod 23 = \; 729 \bmod 23 \;=\; 16$
- Alice and Bob exchange public numbers

Example: Diffie-Hellman Key Exchange Protocol

- Alice and Bob compute symmetric keys
  - $k = v^{a} \bmod p = 16^4 \bmod 23 = 9$
  - $k = u^{b} \bmod p = 6^3 \bmod 23 = 9$
- Alice and Bob now can talk securely!

Difie-Hellman in other groups

- The Diffie-Hellman protocol, and those based on it, can be carried out in any group in which both the discrete logarithm problem is hard, and exponentiation is efficient.
- The most common examples of such groups used in practice are
  - the multiplicative group $Z_p{}^*$
  - the multiplicative group of F_2m
  - the group of points defined by an elliptic curve over a finite field.

Choice of prime p

- Sophie Germain prime: a prime number p is a Sophie Germain prime if $2p + 1$ is also prime. The number $2p + 1$ associated with a Sophie Germain prime is called a safe prime.
- Example: 11 is a Sophie Germain prime and $2 \times 11 + 1 = 23$ is its associated safe prime.

  2, 3, 5, 11, 23, 29, 41, 53, 83, 89, 113 are SG primes
- The order of group should have a large prime factor to prevent use of the Pohlig–Hellman algorithm to obtain discrete log.