

Cryptography CS 555

Lecture 16



Discrete Log, Diffie-Hellman, and El Gamal
Encryption

Review

- The SRA Mental Poker Protocol
 - commutative encryption
- RSA is not semantically secure
 - ciphertext leaks the Jacobi symbol of plaintext
- The Goldwasser-Micali encryption
 - probably IND-CPA secure based on the hardness of the QR problem
- Padding schemes for semantically secure public-key encryption schemes
 - A simple scheme and OAEP

A Word on Commutative Symmetric Ciphers

- All existing standard block ciphers are not commutative
- Stream ciphers are commutative
- One can build symmetric commutative ciphers using modular arithmetic
 - e.g., $(M)^e \bmod p$ or $(M)^e \bmod n$, where $n=pq$ is shared

Lecture Outline

- The discrete log problem
- Diffie-Hellman key exchange protocol
- ElGamal Encryption



Discrete Logarithm Problem (DLP)

- Given a multiplicative group $(G, *)$, an element g in G having order n and an element y in the subgroup generated by g , denoted $\langle g \rangle$
- Find the unique integer x such that

$$g^x \bmod n = y$$

- i.e., x is the discrete logarithm $\log_g y$
- For example, given the group Z_p^* , where p is a 1024-bit prime, let g be an element having prime order q , where q is a 160-bit prime
 - $q \mid (p-1)$
 - e.g., $Z_7^* = \{3, 2, 6, 4, 5, 1\}$, we choose the subgroup $\{2, 4, 1\}$

Choices of Parameters

- Why use an element of order q , instead of just using a generator for Z_p^* ?
- Answer:
 - it is often beneficial to have order being a prime
 - e.g., given e , one can find d s.t. $g^{ed}=g$
 - Balance security and size
 - p needs to be large enough for discrete log to be hard, thus 1024 bits
 - we want the group to be relative small, so that an index to an element in the group is short (e.g., 160 bits)
 - it needs to be large enough to prevent exhaustive search

Algorithms for The Discrete Log Problem

- There are generic algorithms that work for every cyclic group
 - Pollard Rho
 - Pohlig-Hellman
- There are algorithms that work just for some groups such as Z_p^*
 - e.g., the index calculus algorithms
 - these algorithms are much more efficient
 - therefore, 1024 bits are needed for adequate level of security

Discrete Log is not always hard

- Consider the group $(\mathbb{Z}_q, +)$, where q is a prime
 - 0 is identity element
 - and 1 is a generator of order q
 - finding out the discrete log $\log_x y$ is easy
- As every order- q cyclic group is homomorphic, why discrete log is hard in some groups?

Bit Security in Discrete Log

- Even though it is difficult to find $\log_g x$, it is possible to determine some bits in $\log_g x$
 - e.g., let g be the generator of Z_p^* , consider the least significant bit (LSB) of $\log_g x$
 - recall that $\log_g x$ is even iff. x is quadratic residue in Z_p^*
- However, finding some bits (aka. hard-core bits) is as hard as computing discrete log
 - in Z_p^* , when $p-1=2^s t$, where t is odd, computing the s least significant bits are easy, computing the $s+1$ LSB is difficult

Parameters Setup

- How to generate primes p and q s.t. $q \mid (p-1)$ and an order q element in Z_p^* ?
- Need to know the factorization of $(p-1)$
 - enables one to determine the order of any element in Z_p^* ,
- Approach 1: generate a random prime p and then factor $(p-1)$
- Approach 2: generate a random q first, then choose r s.t. $p=2rq+1$ is a prime

Diffie-Hellman Key Establishment

- A and B wishes to establish a shared secret key so that no eavesdropper can compute the key:
- A and B shares public parameters a group Z_p and a generator g
 - A randomly chooses x and send $g^x \bmod p$ to B
 - B randomly chooses y and send $g^y \bmod p$ to A
 - Both A and B can compute $g^{xy} \bmod p$
 - It is (believed to be) infeasible for an eavesdropper to compute $g^{xy} \bmod p$
 - A and B can establish a shared secret without sharing any secret to start with

CDH and DDH

- Security of the Diffie-Hellman key establishment protocol based on the CDH problem
- Computational Diffie-Hellman (CDH)
 - Given a multiplicative group $(G, *)$, an element $g \in G$ having order q , given g^x and g^y , find g^{xy}
- Decision Diffie-Hellman (DDH)
 - Given a multiplicative group $(G, *)$, an element $g \in G$ having order q , given g^x , g^y , and g^z , determine if $g^{xy} \equiv g^z \pmod{n}$
- Discrete Log is at least as hard as CDH, which at least as hard as DDH.

ElGamal

- Published in 1985 by ElGamal
- Its security based on the intractability of the discrete logarithm problem and the CDH and DDH problem
- Message expansion: the ciphertext is twice as big as the original message
- Uses randomization, each message has $p-1$ possible different encryptions

El Gamal

Key Generation

- Generate a large random prime p such that DLP is infeasible in Z_p and a generator g of the multiplicative group Z_p of the integers modulo p
- Select a random integer a , $1 \leq a \leq p-2$, and compute
$$g^a \bmod p$$
- Public key is $(p; g; \beta=g^a)$
- Private key is a .

ElGamal (cont.)

Encryption:

Message M into ciphertext C

Select a random integer k , $0 \leq k \leq p-2$.

Compute $\gamma = g^k \bmod p$ and $\delta = m \beta^k \bmod p$.

Ciphertext $C = (\gamma, \delta)$

Decryption:

Compute γ^{-a} as follows: $\gamma^{p-1-a} \bmod p = \gamma^{-a} \bmod p$

$m = \gamma^{-a} \delta \bmod p$

WHY DECRYPTION WORKS?

$$\gamma^{-a} \delta \bmod p \equiv g^{-ka} m \cdot (g^a)^k \bmod p \equiv m \bmod p$$

Parameters Size

- All parties could use the same modulus p and generator g
 - they choose different
- Different encryptions should use different k
- Prime p should be chosen as 1024 bits to ensure that DLP is infeasible, while k should be 160 bits
- Algorithm can also be defined in groups other than Z_p^*
 - e.g., in elliptic curves

Security of ElGamal

- ElGamal is not semantically secure.
- WHY? An attacker can learn information about the plaintext without decrypting: given two encryptions, can say which plaintext was a quadratic residue and which one was not.

Semantically Secure ElGamal

- Choose p such that $p = 2q + 1$, where q is also prime
- Then define ElGamal in Q_p , the subgroup of quadratic residues modulo p , this subgroup is a cyclic subgroup of Z_p having order q
- Equivalent with restricting the message m , α^a and $y_1 = \alpha^k \pmod p$ to be quadratic residues

ElGamal and DH Problems

- Semantic security of ElGamal is equivalent to the infeasibility of Decision Diffie-Hellman
- ElGamal decryption (without knowing the public key) is equivalent to solving Computational Diffie-Hellman

CDH and ElGamal

Prove that any algorithm that solves CDH can be used to decrypt ElGamal ciphertexts

Intuition: Compute m from $(\gamma = g^k, \delta = m \beta^k)$ is equivalent to compute β^k , one knows $\gamma = g^k$, $\beta = g^a$, and needs to compute g^{ka} .

Formal Proof: “ \Rightarrow ” Assume that algorithm OracleCDH solves CDH

and let (γ, δ) be an ElGamal encryption and

let public key $(g, \beta = g^a)$ $\gamma = g^k \pmod p$, $\delta = m (g^a)^k \pmod p$

$y = \text{OracleCDH}(g, \beta, \gamma)$ and

$x = \delta y^{-1}$ then x is the decryption of (γ, δ)

Decision D-H \Rightarrow ElGamal

- Given decision D-H oracle, find two messages whose ElGamal encryptions can be distinguished
- For any two M_0, M_1 : ($\beta = g^a$)
 - $E(M_0) = g^x, M_0 \beta^x, \quad E(m_1) = g^y, M_1 \beta^y$
 - Suppose receive ciphertext (γ, δ)
 - Feed $\langle \gamma, \beta g^r, \delta \gamma^r / m_0 \rangle$
 - when (γ, δ) is $E(M_0)$, this is $\langle g^x, g^{a+r}, M_0 g^{ax} g^{xr} / M_0 \rangle = \langle g^x, g^{a+r}, g^{x(a+r)} \rangle$
 - when (γ, δ) is $E(M_1)$, this is $\langle g^x, g^{a+r}, g^{x(a+r)} M_1 / M_0 \rangle$
 - if the DDH oracle say yes, we say 0, otherwise we say 1

Summary

- Discrete Log
- Diffie-Hellman
- El Gamal



Next ...

- Digital Signatures
- Reading: 7.1, 7.2, 7.3

