Cryptography

Cunsheng DING

HKUST, Hong Kong

COMP3631

Lecture 12: The ElGamal Public-Key Cipher

Main Topics of This Lecture

1. The discrete logarithm problem.
2. The ElGamal public-key cipher.

The Discrete Logarithm Problem

Primitive Roots of a Prime p: Definition

Definition: Let p be a prime. A positive integer α is a **primitive root** of p if each nonzero element $a \in \mathbf{Z}_p$ can be expressed as

$$a = \alpha^i \bmod p$$

for a unique integer i with $0 \leq i \leq p-2$.

Example: 2 is a primitive root of the prime 11.

i	0	1	2	3	4	5	6	7	8	9
$2^i \bmod 11$	1	2	4	8	5	10	9	7	3	6

Primitive Roots of a Prime p: Existence

Theorem: Every prime p has at least one primitive root.

Remark: A proof of this theorem can be found in a book on elementary number theory. You are not required to read a proof of this theorem.

Remark: The exact number of primitive roots of p in the range $\{1, 2, \ldots, p-1\}$ is given by $\phi(p-1)$ where $\phi(n)$ is Euler's totient function (see Lectures 10 and 11)

Primitive Roots of a Prime p: Construction

The rule of thumb: Most primes p have a small primitive root. For example, for the primes less than 100000, approximately 37.5% have 2 as a primitive root, and approximately 87.4% have a primitive root of value 7 or less.

For primes of reasonable size, the Magma software package can be used to find a primitive root. See http://magma.maths.usyd.edu.au/calc/

The Discrete Logarithm

Discrete logarithm: Let p be a prime and α be a **primitive root** of p. By definition, each nonzero element $a \in \mathbf{Z}_p$ can be expressed as

$$a = \alpha^i \bmod p$$

for a unique integer i with $0 \leq i \leq p-2$.

The exponent i is referred to as the **discrete logarithm**, or **index**, of a for the base α, and is denoted $\log_\alpha a$ or $\mathrm{ind}_\alpha(a)$.

Remark: The **discrete logarithm** is very similar to the **logarithm** we learnt in secondary school.

The Discrete Logarithm Problem

The discrete logarithm problem: Let p be a prime, and let α be a primitive root of p. The *discrete logarithm problem* is to find $\log_\alpha a$ for any $1 \leq a \leq p-1$, which is defined to be the unique integer $0 \leq i \leq p-2$ such that

$$a = \alpha^i \bmod p.$$

Comment: No polynomial-time algorithm is known for this problem (except for certain special primes p). See

`https://en.wikipedia.org/wiki/Discrete_logarithm`

Comment: If p has 160 or more digits, the DLP is **believed** to be computationally very hard in general.

An Efficient Algorithm for Computing $\alpha^i \bmod p$

The first step is to compute the binary representation of i (very efficient):

$$i = 2^{i_1} + 2^{i_2} + \cdots + 2^{i_t}$$

for some positive integer t. The second step is to compute

$$\alpha^{2^{i_j}} = (\cdots((\alpha^2))^2 \cdots)^2$$

Computing each $a^{2^{i_j}}$ takes i_j multiplications. The last step is compute their product:

$$a := \alpha^i \bmod p = \alpha^{2^{i_1}} \times \alpha^{2^{i_2}} \times \cdots \times \alpha^{2^{i_t}} \bmod p.$$

Hence, the algorithm takes $i_1 + i_2 + \cdots + i_t + t - 1$ modulo-p multiplications, which is quite efficient.

The ElGamal Public-Key Cipher

Public Parameters for the ElGamal Cipher

Choosing system parameters:

- Choose p to be a large prime, and
- choose α to be a primitive root of p.

Both p and α are in the public domain and are known to everyone.

Key Generation for the ElGamal Cipher

User's key pair:

- Each user chooses a secret number u in $\mathbf{Z}_{p-1}$, as his/her private key $k_d := u$.
- The corresponding public key $k_e = (p, \alpha, \beta)$, where $\beta = \alpha^u \bmod p$.

Attention: The relation between the public key and the private key is clear. They form a related pair.

The public key (p, α, β) contains all information about the private key u, but computing the private key d from the public key (p, α, β) is believed to be hard.

The Building Blocks of the ElGamal Cipher

- $\mathcal{M} := \mathbf{Z}_p^\times := \mathbf{Z}_p \setminus \{0\} = \{1, \ldots, p-1\}$
- $\mathcal{C} = \mathbf{Z}_p^\times \times \mathbf{Z}_p^\times$
- $\mathcal{K}_e = \{p\} \times \{\alpha\} \times \mathbf{Z}_p^\times$. So $|\mathcal{K}_e| = p - 1$.

  The public key $k_e = (p, \alpha, \beta)$.
- $\mathcal{K}_d = \mathbf{Z}_{p-1}$. Thus $|\mathcal{K}_d| = p - 1$.

  The private key $k_d = u$ such that $\beta = \alpha^u \bmod p$.

The Encryption and Decryption

Encryption: For any public key $k_e = (p, \alpha, \beta)$, and for a (secret) random number $v \in \mathbf{Z}_{p-1}$,

$$E_{k_e}(x, v) = (y_1, y_2),$$

where

$$y_1 = \alpha^v \bmod p, \quad y_2 = x\beta^v \bmod p.$$

Decryption: For any $(y_1, y_2) \in \mathbf{Z}_p^\times \times \mathbf{Z}_p^\times$,

$$D_{k_d}(y_1, y_2) = y_2 \left(y_1^{k_d}\right)^{-1} \bmod p.$$

Remarks: Encryption is nondeterministic, and has data expansion. The receiver can decrypt the ciphertext c without knowing the random number v chosen by the sender.

The Correctness of Decryption

Proof: Note that $\beta = \alpha^u \bmod p$ and

$$y_1 = \alpha^v \bmod p, \quad y_2 = x\beta^v \bmod p.$$

We have

$$\begin{aligned} D_{k_d}(y_1, y_2) &= y_2\,(y_1^u)^{-1} \bmod p \\ &= (x\beta^v \bmod p)\,([\alpha^v \bmod p]^u)^{-1} \bmod p \\ &= x\alpha^{uv}(\alpha^{uv})^{-1} \bmod p \\ &= x(\alpha^{uv}(\alpha^{uv})^{-1} \bmod p) \bmod p \\ &= x \bmod p \\ &= x. \end{aligned}$$

Weak Keys in the ElGamal Cipher

The following two pairs of keys are weak (in fact, cannot be used):

- $k_e = (p, \alpha, \alpha)$, $k_d = u = 1$.

  Once k_e is published, k_d is easily seen to be 1. Hence $x = y_2 y_1^{-1} \bmod p$.

- $k_e = (p, \alpha, 1)$, $k_d = u = 0$.

  Once k_e is published, k_d is easily seen to be 0. Hence $x = y_2$.

Here we have seen specific examples of "weak" keys!

An Example

Choose $p = 2579$, $\alpha = 2$, and $u = 765$. Hence $\beta = 2^{765} \bmod 2579 = 949$.

- Public key $k_e = (p, \alpha, \beta) = (2579, 2, 949)$.
- Private key $k_d = u = 765$.

Encryption: Let $x = 1299$ and choose a random $v = 853$,

$$\begin{aligned} E_{k_e}(x, v) &= (y_1, y_2) = (\alpha^v \bmod p, x\beta^v \bmod p) \\ &= (2^{853} \bmod 2579, 1299 \cdot 949^{853} \bmod 2579) = (435, 2396). \end{aligned}$$

Decryption:

$$D_{k_d}(y_1, y_2) = y_2 \left(y_1^{k_d}\right)^{-1} \bmod p = 2396 \times (435^{765})^{-1} \bmod 2579 = 1299.$$

Security of the ElGamal Cipher

Question 1: Is it computationally feasible to derive the private key k_d from the public key k_e?

Solution: Note that $k_e = (p, \alpha, \beta)$, where

$$\beta = \alpha^u \bmod p = \alpha^{k_d} \bmod p.$$

It depends on whether there is an efficient algorithm for solving the discrete logarithm problem.

It is believed that there is no polynomial-time algorithm for the DLP in general. So if p is large enough, say with 160 digits, and is not in certain special forms, it is computationally infeasible to derive k_d from k_e.

Security of the ElGamal Cipher

Question 2: Given a ciphertext (y_1, y_2), is it computationally feasible to derive its corresponding plaintext x?

Attack 1: One way is to use $x = y_2(\beta^v)^{-1} \bmod p$, where $v \in \mathbf{Z}_{p-1}$ and β is publicly known. Since v is a secret random number, this does not work if p is large enough.

Attack 2: The second way is to use

$$x = D_{k_d}(y_1, y_2) = y_2 \left(y_1^{k_d}\right)^{-1} \bmod p.$$

This does not work either as it is hard to determine k_d.

Answer: It is believed that the answer to Question 2 in general is NO.

An Online Demo of the ElGamal Cipher

`https://8gwifi.org/elgamalfunctions.jsp#google_vignette`

Remark: The ElGamal public key cipher is widely used in real-world security systems.