暨南大學
JINAN UNIVERSITY

Lecture 7: The ElGamal Cryptosystem and Discrete Logarithms

-Cryptographic Algorithms and Protocols

Huang, Xiujie (黄秀姐)

Office: Nanhai Building, #411

E-mail: t_xiujie@jnu.edu.cn

Dept. Computer Science

Review

- **Public-key Cryptography (PKC)**
  - **1. RSA**

Review

- **Public-key Cryptography (PKC)**
  - **1. RSA ←→ Factoring Integers**

Others ???

  - **2. ElGamal ←→ Discrete Logarithm**

Outline

- **1. The ElGamal Cryptosystem**
  - **Discrete Logarithm Problem (DLP)**
  - **The ElGamal Cryptosystem**
- **2. Algorithms for the DLP**
  - **Shanks' Algorithm**
- **3. Suitable Groups for the DLP**
  - **Finite Fields & Elliptic Curves**
  - **Suitable Groups for the DLP**
- **4. Security of ElGamal Systems**
  - **Bit Security and Semantic Security**
  - **The Diffie-Hellman Problems**

The Finite Group

- **Finite multiplicative group (G, ·)**
  - Cyclic subgroup $\langle \alpha \rangle = \{\alpha^i : 0 \le i \le n-1\}$

✓ **$G = \mathbb{Z}_p^*$, p is prime**

- **$\langle \alpha \rangle = G$ since $G = \mathbb{Z}_p^*$ is a cyclic group.**
- **$\alpha = b^{(p-1)/q}$, where b is the primitive of element in G, i.e.,**

$$\langle \alpha \rangle = \{\alpha^i : 0 \le i \le q-1\}$$

Examples:

① $Z_5^*=\{1,2,3,4\}$. ord(1)=1, ord(2)=4, ord(3)=4, ord(4)=2.

$Z_5^* = \langle 2 \rangle = \langle 3 \rangle$; $\langle 4 \rangle = \{1, 4\}$

② $Z_{13}^*=\{1,2,3,\ldots\ldots, 11, 12\}$. $\langle 2 \rangle = Z_{13}^*$, ($2^5 \equiv 6 \pmod{13}$, $2^{11} \equiv 7 \pmod{13}$);

$\langle 5 \rangle = \{1,5, 8,12\}$, ($5^3 \equiv 8 \pmod{13}$).

The Discrete Logarithm Problem (DLP)

Problem 7.1: Discrete Logarithm

Instance: A multiplicative group $(G, \cdot)$, an element $\alpha \in G$ having order n, and an element $\beta \in \langle \alpha \rangle$.

Question: Find the unique integer a, $0 \leq a \leq n-1$, such that

$$\boxed{\alpha^a = \beta.}$$

We will denote this integer a by $\log_\alpha \beta$; it is called the *discrete logarithm* of β.

$$\boxed{a = \log_\alpha \beta}$$

- **Exponentiation** is a ***one-way*** function in **suitable groups *G***.
  - Operation of exponentiation is computable: Algorithm 6.5
  - Finding *a* is (probably) difficult

The ElGamal Cryptosystem, 1985

- **Set-up of Key Generation:**
  1) Generate a large prime p such that the DLP in Z_p^* is infeasible
  2) Choose a primitive element $\alpha \in Z_p^*$
  3) Choose a random number a and Compute $\beta \equiv \alpha^a \pmod p$
  4) Output: pk = (p, α, β), sk = (a)

- **Encryption:**

Randomized:
one plaintext, p-1 ciphertexts

  1) Choose a secret random number k in Z_{p-1}
  2) Compute $\boxed{e_{pk}(x) = (y_1, y_2)}$ where $y_1 = \alpha^k \bmod p$ and $y_2 = x\beta^k \bmod p$

- **Decryption:**

Correctness: $d_{sk}(y_1, y_2) = x$?

  1) Compute $d_{sk}(y_1, y_2) = y_2(y_1{}^a)^{-1}$ m

$$d_{sk}(y_1, y_2) = y_2(y_1{}^a)^{-1}$$
$$= x\beta^k ((\alpha^k)^a)^{-1} = x\beta^k ((\alpha^a)^k)^{-1}$$
$$= x\beta^k (\beta)^k)^{-1} = x$$

Cryptosystem 7.1, P25

Example 1

- $p = 13$, $\alpha = 2$, $a = 5$, $\beta = 2^5 \bmod 13 = 6$
  - pk = (13, 2, 6), sk = (5)

- Encryption of $x = 10$

1) Choose a secret random number $k = 7$ in $Z_{p\text{-}1}$
2) Compute $e_{\mathrm{pk}}(x) = (y_1, y_2)$ where $y_1 = \alpha^k \bmod p = 2^7 \bmod 13 = 11$, and $y_2 = x\beta^k \bmod p = 10 \cdot 6^7 \bmod 13 = 5$.

- Decryption

1) Compute $d_{\mathrm{sk}}(y_1, y_2) = y_2(y_1^{\,a})^{-1} \bmod p = 5 \cdot (11^5)^{-1} \bmod 13$

$$= 5 \cdot (7)^{-1} \bmod 13 = 5 \cdot 2 \bmod 13 = 10.$$

Example 2

Example 7.1: Suppose $p = 2579$ and $\alpha = 2$. α is a primitive element modulo p. Let $a = 765$, so

$$\beta = 2^{765} \bmod 2579 = 949.$$

- **Case 1: x = 1299, k = 853**
  - $e_K(x,k) = (y_1, y_2)$: $y_1 = \alpha^k \bmod p = \underline{2^{853} \bmod 2579}$

    $$= 435,$$

    $$y_2 = x\beta^k \bmod p = 1299 \times \underline{949^{853} \bmod 2579}$$

    $$= 2396.$$
  - $d_K(y_1, y_2) = y_2({y_1}^a)^{-1} \bmod p$

    $$= 2396 \times \underline{(435^{765})^{-1} \bmod 2579}$$

    $$= 1299,$$
- **Case 2: x = 1299, k = 1000?**

The ElGamal Cryptosystem, 1985

Set-up of Key Generation:

1) Generate a large prime p such that the DLP in Z_p^* is infeasible
2) Choose a primitive element $\alpha \in Z_p^*$
3) Choose a random number a and Compute $\beta \equiv \alpha^a \pmod{p}$
4) Output: $\boxed{\text{pk} = (p, \alpha, \beta)}$, $\boxed{\text{sk} = (a)}$

Encryption:

1) Choose a secret random number k in Z_{p-1}
2) Compute $e_{pk}(x) = (y_1, y_2)$ where $y_1 = \alpha^k \bmod p$ and $y_2 = x\beta^k \bmod p$

Decryption:

1) Compute $d_{sk}(y_1, y_2) = y_2(y_1^{a})^{-1} \bmod p$

Cryptosystem 7.1, P257

Choices of Large Prime p

- **A necessary condition to be secure:**
  - **the Discrete Logarithm problem in $\mathbb{Z}_p^*$ is infeasible, i.e., there is no known polynomial-time algorithm to solve DLP**
    - ✓ ***p*** **should have at least 2048 bits**
    - ✓ ***p*****-1 should have at least one "large" prime factor**

Outline

- 1. The ElGamal Cryptosystem
  - Discrete Logarithm Problem (DLP)
  - The ElGamal Cryptosystem
- 2. Algorithms for the DLP
  - Shanks' Algorithm
- 3. Suitable Groups for the DLP
  - Finite Fields & Elliptic Curves
  - Suitable Groups for the DLP
- 4. Security of ElGamal Systems
  - Bit Security and Semantic Security
  - The Diffie-Hellman Problems

DLP

Problem 7.1: Discrete Logarithm

Instance: A multiplicative group $(G, \cdot)$, an element $\alpha \in G$ having order n, and an element $\beta \in \langle \alpha \rangle$.

Question: Find the unique integer a, $0 \leq a \leq n - 1$, such that

$$\boxed{\alpha^a = \beta.}$$

We will denote this integer a by $\log_\alpha \beta$; it is called the *discrete logarithm* of β.

$$\boxed{a = \log_\alpha \beta}$$

Algorithms for DLP

An assumption: computing the product of two elements requires constant (i.e., O(1)) time.

- **Some trivial algorithms**:
  - **1) Exhaustive search:** O(n) time, O(1) memory
  - **2) Precomputing** (i, α^i) **→ sorting → searching:**

    O(n) time; O(n log n) time; O(log n) time; O(n) memory

Algorithms for DLP

- **Some non-trivial algorithms:**
  - **1. Shanks' Alogrithm**
  - **2. The Pollard Rho Discrete Logarithm Algorithm**
  - **3. The Pohig-Hellman Algorithm**
  - **4. The Index Calculus Method (for $\mathbb{Z}_p{}^*$ only)**

Generic algorithm:

- **if the algorithm for the DLP can be applied in any group.**
- **1, 2, and 3 are generic algorithms, while 4 is not.**

Shanks' Algorithm for DLP

Algorithm 7.1: $\text{SHANKS}(G, n, \alpha, \beta)$

1. $m \leftarrow \lceil \sqrt{n} \rceil$
2. **for** $j \leftarrow 0$ **to** $m - 1$
   do compute α^{mj} --O(m) time, O(m) memory
3. Sort the m ordered pairs $\boxed{(j, \alpha^{mj})}$ with respect to their second coordinates, obtaining a list L_1 --O(m log(m)) time, O(m) memory
4. **for** $i \leftarrow 0$ **to** $m - 1$
   do compute $\beta\alpha^{-i}$ --O(m) time, O(m) memory
5. Sort the m ordered pairs $\boxed{(i, \beta\alpha^{-i})}$ with respect to their second coordinates, obtaining a list L_2 --O(m log(m)) time, O (m) memory
6. Find a pair $(j, y) \in L_1$ and a pair $(i, y) \in L_2$ (i.e., find two pairs having identical second coordinates) --O (m) time, O (m) memory
7. $\boxed{\log_\alpha \beta \leftarrow (mj + i)}$ mod n

An Example of Shanks' Algorithm

Example **7.2:** Suppose we wish to find $\log_3 525$ in $(\mathbb{Z}_{809}{}^*, \cdot)$. Note that 809 is prime and 3 is a primitive element in $\mathbb{Z}_{809}{}^*$, so we have $\alpha = 3$, $n = 808$, $\beta = 525$ and $m = \lceil \sqrt{808} \rceil = 29$. Then

precompute α^m mod p: $\alpha^{29} \bmod 809 = 99.$

First, we compute the ordered pairs $(j, 99^j \bmod 809)$ for $0 \leq j \leq 28$. We obtain the list $(j, \alpha^{mj} \bmod p)$

$\boxed{(0, 1)}$	$(1, 99)$	$(2, 93)$	$(3, 308)$	$(4, 559)$
$(5, 329)$	$(6, 211)$	$(7, 664)$	$(8, 207)$	$(9, 268)$
$(10, 644)$	$(11, 654)$	$\boxed{(12, 26)}$	$(13, 147)$	$(14, 800)$
$(15, 727)$	$(16, 781)$	$(17, 464)$	$(18, 632)$	$(19, 275)$
$(20, 528)$	$(21, 496)$	$(22, 564)$	$\boxed{(23, 15)}$	$(24, 676)$
$(25, 586)$	$(26, 575)$	$(27, 295)$	$(28, 81)$	

which is then sorted to produce L_1.

An Example of Shanks' Algorithm

See Example 7.2

The second list contains the ordered pairs $(i, 525 \times (3^i)^{-1} \bmod 809)$, $0 \le j \le 28$. It is as follows:

$(i, \beta\alpha^{-i} \bmod p)$

$(0, 525)$	$(1, 175)$	$(2, 328)$	$(3, 379)$	$(4, 396)$
$\boxed{(5, 132)}$	$\boxed{(6, 44)}$	$(7, 554)$	$(8, 724)$	$(9, 511)$
$(10, 440)$	$(11, 686)$	$(12, 768)$	$(13, 256)$	$(14, 355)$
$(15, 388)$	$(16, 399)$	$(17, 133)$	$(18, 314)$	$(19, 644)$
$(20, 754)$	$(21, 521)$	$(22, 713)$	$(23, 777)$	$(24, 259)$
$(25, 356)$	$(26, 658)$	$(27, 489)$	$(28, 163)$	

After sorting this list, we get L_2.

Now, if we proceed simultaneously through the two sorted lists, we find that $(10, 644)$ is in L_1 and $(19, 644)$ is in L_2. Hence, we can compute

$$\log_3 525 = (29 \times 10 + 19) \bmod 808$$
$$= 309.$$

$$\boldsymbol{a = \log_\alpha \beta = (mj + i) \bmod n}$$

As a check, it can be verified that $3^{309} \equiv 525 \pmod{809}$.

Outline

- **1. The ElGamal Cryptosystem**
  - **Discrete Logarithm Problem (DLP)**
  - **The ElGamal Cryptosystem**
- **2. Algorithms for the DLP**
  - **Shanks' Algorithm**
- **3. Suitable Groups for the DLP**
  - **Finite Fields & Elliptic Curves**
  - **Suitable Groups for the DLP**
- **4. Security of ElGamal Systems**
  - **Bit Security and Semantic Security**
  - **The Diffie-Hellman Problems**

Suitable Groups for ElGamal Crypt

- **The ElGamal Cryptosystem can be implemented in any group where the DLP is infeasible**
  - $\mathbb{Z}_p{}^*$, ***p*** **is a large prime**
  - **1. the multiplicative group of the $\boxed{\text{Finite Field } \mathbb{F}_{p^n}}$, *p* is prime**
  - **2. the group of an $\boxed{\text{Elliptic Curve}}$ defined over a finite field**

Finite Field

- ($\mathbb{Z}_p{}^*$, +, ·), *p* is prime
- ($\mathbb{F}_{p^n}$, +, ·), *p* is prime

Construction of Finite Field $\mathbb{F}_{p^n}$

❖ Congruence of polynomials

Definition 7.1: Suppose p is prime. Define $\boxed{\mathbb{Z}_p[x]}$ to be the set of all polynomials in the indeterminate x. By defining addition and multiplication of polynomials in the usual way (and reducing coefficients modulo p), we construct a ring.

For $f(x), g(x) \in \mathbb{Z}_p[x]$, we say that $f(x)$ *divides* $g(x)$ (notation: $f(x) \mid g(x)$) if there exists $q(x) \in \mathbb{Z}_p[x]$ such that

$$g(x) = q(x)f(x).$$

For $f(x) \in \mathbb{Z}_p[x]$, define $\deg(f)$, the *degree* of f, to be the highest exponent in a term of f.

Suppose $f(x), g(x), h(x) \in \mathbb{Z}_p[x]$, and $\deg(f) = n \geq 1$. We define

$$\boxed{g(x) \equiv h(x) \pmod{f(x)}}$$

if

$$f(x) \mid (g(x) - h(x)).$$

Construction of Finite Field $\mathbb{F}_{p^n}$

- **Quotient Ring:** $\mathbb{Z}_p[x]/(f(x))$ = {$g(x)$ mod $f(x)$: $g(x)$ is in $\mathbb{Z}_p[x]$ }

Suppose $\deg(f) = n$.

Now we define the elements of $\mathbb{Z}_p[x]/(f(x))$ to be the p^n polynomials in $\mathbb{Z}_p[x]$ of degree at most $n-1$. Addition and multiplication in $\mathbb{Z}_p[x]/(f(x))$ is defined as in $\mathbb{Z}_p[x]$, followed by a reduction modulo $f(x)$. Equipped with these operations, $\mathbb{Z}_p[x]/(f(x))$ is a ring.

$$\mathbb{Z}_p[x]/(f(x)) = \{a_{n-1}x^{n-1} + \ldots + a_2x^2 + a_1x + a_0 : a_i \text{ in } \mathbb{Z}_p\}$$

Construction of Finite Field $\mathbb{F}_{p^n}$

Irreducible polynomial

Definition 7.2: A polynomial $f(x) \in \mathbb{Z}_p[x]$ is said to be *irreducible* if there do not exist polynomials $f_1(x), f_2(x) \in \mathbb{Z}_p[x]$ such that

$$f(x) = f_1(x) f_2(x),$$

where $\deg(f_1) > 0$ and $\deg(f_2) > 0$.

A very important fact is that $\mathbb{Z}_p[x]/(f(x))$ is a field if and only if $f(x)$ is irreducible. Further, multiplicative inverses in $\mathbb{Z}_p[x]/(f(x))$ can be computed using a straightforward modification of the (extended) Euclidean algorithm.

Existence and Uniqueness:

- **Existence for the irreducible poly. of any *n***
- **Isomorphism of any two finite fields with same *p* and *n***

Example of the Finite Field

- **Order: 2^n**
- **$Z_2[x]/(f(x))$: $f(x)$ is irreducible and deg(f) = n**
- $a_{n-1}x^{n-1} + \ldots + a_2x^2 + a_1x + a_0$

  $\leftrightarrow (a_{n-1} \ldots a_2 a_1 a_0)$
- **See Example on Pages 273-274**

For example, to compute $(x^2+1)(x^2+x+1)$ in $\mathbb{Z}_2[x]/(x^3+x+1)$, we first compute the product in $\mathbb{Z}_2[x]$, which is x^4+x^3+x+1. Then we divide by x^3+x+1, obtaining the expression

Extended Euclidean Alg. for Polynomials

$$x^4+x^3+x+1 = (x+1)(x^3+x+1) + x^2 + x.$$

Hence, in the field $\mathbb{Z}_2[x]/(x^3+x+1)$, we have that

$$(x^2+1)(x^2+x+1) = x^2 + x.$$

Elliptic Curves (椭圆曲线)

Elliptic Curves over the Reals

Definition 7.3: Let $a, b \in \mathbb{R}$ be constants such that $4a^3 + 27b^2 \neq 0$. A *non-singular elliptic curve* is the set $\mathcal{E}$ of solutions $(x, y) \in \mathbb{R} \times \mathbb{R}$ to the equation

$$\boxed{y^2 = x^3 + ax + b,}$$

together with a special point $\mathcal{O}$ called the *point at infinity*.

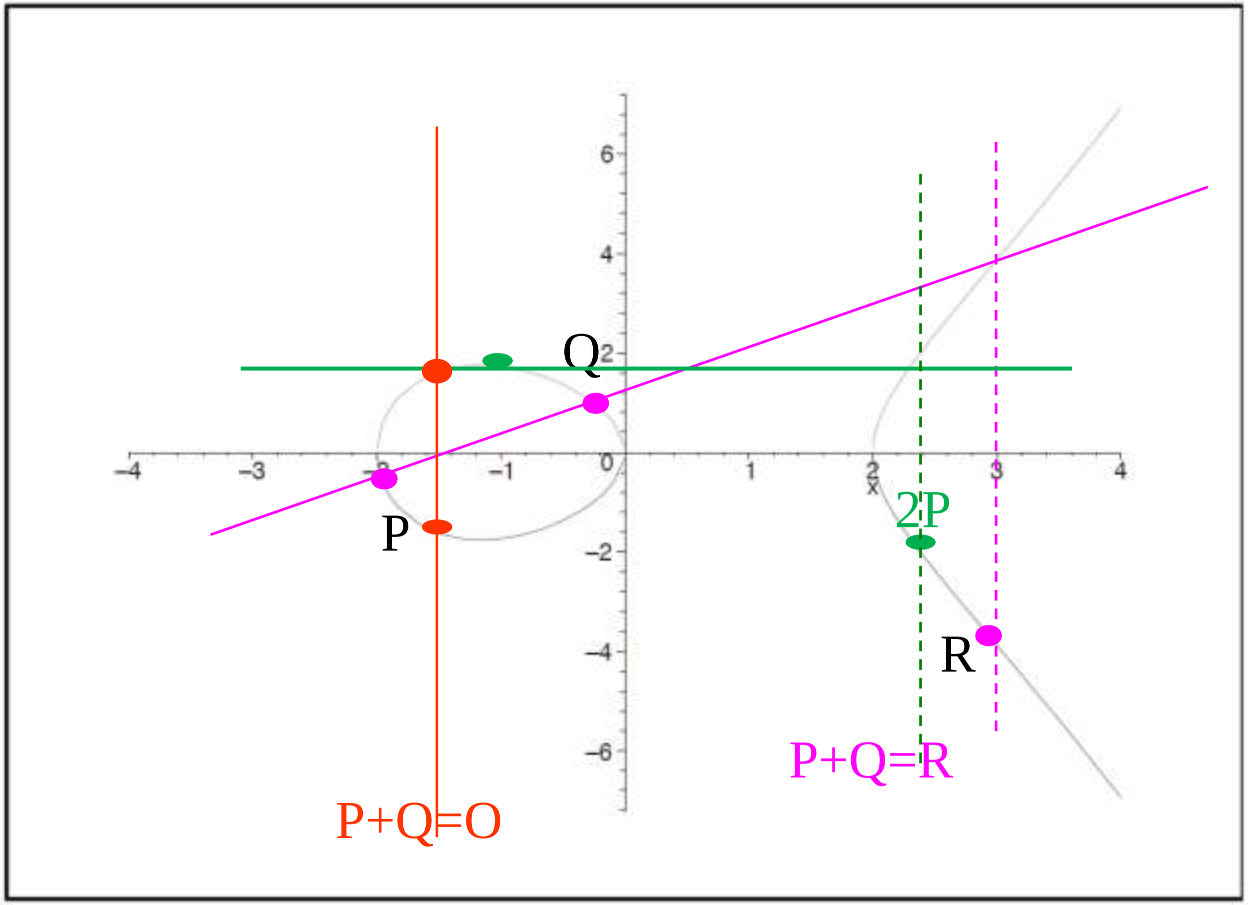
$4a^3 + 27b^2 \neq 0 \Longleftrightarrow x^3 + ax + b = 0$ has three distinct roots

$\mathcal{E}$ **is an abelian addition group** P+Q=R

1. addition is closed on the set $\mathcal{E}$,
2. addition is commutative,
3. $\mathcal{O}$ is an identity with respect to addition, and
4. every point on $\mathcal{E}$ has an inverse with respect to addition.
5. addition is associative

Elliptic Curves

- **Elliptic Curves over the Reals**
  - **Example:** the elliptic curve $y^2 = x^3 - 4x$.



Elliptic Curves

Elliptic Curves over the Reals

Abelian group

Suppose E is a non-singular elliptic curve. We will define a binary operation over E which makes E into an abelian group. This operation is usually denoted by addition. The point at infinity, $\mathcal{O}$, will be the identity element, so $P + \mathcal{O} = \mathcal{O} + P = P$ for all $P \in E$.

Suppose $P, Q \in E$, where $P = (x_1, y_1)$ and $Q = (x_2, y_2)$. $P + Q = (x_3, y_3)$

1. $x_1 \neq x_2$

$$x_3 = \lambda^2 - x_1 - x_2, \qquad y_3 = \lambda(x_1 - x_3) - y_1, \qquad \lambda = \frac{y_2 - y_1}{x_2 - x_1}.$$

2. $x_1 = x_2$ and $y_1 = -y_2$

$$(x, y) + (x, -y) = \mathcal{O}$$

3. $x_1 = x_2$ and $y_1 = y_2$

$$x_3 = \lambda^2 - x_1 - x_2, \qquad y_3 = \lambda(x_1 - x_3) - y_1, \qquad \lambda = \frac{3{x_1}^2 + a}{2y_1}.$$

Elliptic Curves Modulo a Prime

❖ Elliptic Curves over $\mathbb{Z}_p$, where $p > 3$ is prime

Definition 7.4: Let $p > 3$ be prime. The *elliptic curve* $y^2 = x^3 + ax + b$ over $\mathbb{Z}_p$ is the set of solutions $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ to the congruence

$$\boxed{y^2 \equiv x^3 + ax + b \pmod{p},} \tag{7.10}$$

where $a, b \in \mathbb{Z}_p$ are constants such that $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$, together with a special point $\mathcal{O}$ called the *point at infinity*.

Suppose $P, Q \in E$, where $P = (x_1, y_1)$ and $Q = (x_2, y_2)$.

If $x_2 = x_1$ and $y_2 = -y_1$, then $P + Q = \mathcal{O}$;

else: $P + Q = (x_3, y_3)$

$$x_3 = \lambda^2 - x_1 - x_2,$$

$$y_3 = \lambda(x_1 - x_3) - y_1,$$

$$\lambda = \begin{cases} (y_2 - y_1)(x_2 - x_1)^{-1}, & \text{if } P \neq Q \\ (3{x_1}^2 + a)(2y_1)^{-1}, & \text{if } P = Q. \end{cases}$$

$$P + \mathcal{O} = \mathcal{O} + P = P$$

Elliptic Curves (椭圆曲线)

- 1. Elliptic Curves over the Reals
- 2. Elliptic Curves over $\mathbb{Z}_p$, where p > 3 is prime
  - Example7.9
- 3. Elliptic Curves over Finite Fields $\mathbb{F}_{p^n}$
  - Example 7.10

Elliptic Curves

4. Properties of Elliptic Curves

An elliptic curve $\mathcal{E}$ defined over $\mathbb{F}_q$ (where $q = p^n$ for p prime,)

Hasse asserts: $q+1-2\sqrt{q} \leq \#\mathcal{E} \leq q+1+2\sqrt{q}$

Schoof's algorithm: to compute $\#\mathcal{E}$ efficiently

THEOREM 7.1 *Let $\mathcal{E}$ be an elliptic curve defined over $\mathbb{F}_q$, where $q = p^n$ for some prime p. Then there exist positive integers n_1 and n_2 such that $(\mathcal{E}, +)$ is isomorphic to $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$. Further, $n_2 \mid n_1$.*

- **An elliptic curve having a cyclic subgroup *G* of size about 2^{224} will provide a secure setting for a cryptosystem, provided that #*G* is divisible by at least one large prime factor.**
  - **The ECDLP is hard to be solved**

ECDLP:

Given P, Q ∈ G (where Q = *m*P). Find *m*.

Elliptic Curves

5.ElGamal Cryptosystems on Elliptic Curves

Cryptosystem 7.2: *Elliptic Curve ElGamal*

Let $\mathcal{E}$ be an elliptic curve defined over $\mathbb{Z}_p$ (where $p > 3$ is prime) such that $\mathcal{E}$ contains a cyclic subgroup $H = \langle P \rangle$ of prime order n in which the **Discrete Logarithm** problem is infeasible. Let $h : \mathcal{E} \to \mathbb{Z}_p$ be a secure hash function.
Let $\mathcal{P} = \mathbb{Z}_p$ and $\mathcal{C} = (\mathbb{Z}_p \times \mathbb{Z}_2) \times \mathbb{Z}_p$. Define

$$\mathcal{K} = \{(\mathcal{E}, P, m, Q, n, h) : Q = mP\},$$

where P and Q are points on $\mathcal{E}$ and $m \in \mathbb{Z}_n^*$. The values $\mathcal{E}$, P, Q, n, and h are the pub

For K ... or a plainte

For a ci

EC-ElGamal Crypt:

1. **KeyGen(θ): pk = ($\mathcal{E}$, P, Q, n, h), sk = (m);**
2. **Enc$_{pk}$(x)=(y_1, y_2): first choose k randomly,**
   then compute $y_1 = kP$, $y_2 = x+h(kQ) \bmod p$;
3. **Dec$_{sk}$(y_1, y_2) = $y_2 - h(my_1) \bmod p$.**

Correctness: $R = my_1 = mkP = kQ$.

where

$$R = m\,\text{Point-Decompress}(y_1).$$

Elliptic Curves

6. Pairings on Elliptic Curves

- **First used in Cryptography by Menezes, Okamoto and Vanstone for solving DLP**
- **widely used in identity-based cryptosystems**

Definition 7.5: A *pairing* is a function e that takes elements P_1 from an abelian group G_1 and P_2 from an abelian group G_2 and returns an element $e(P_1, P_2) = g$ belonging to a group G_3:

$$\begin{aligned} e : G_1 \times G_2 &\rightarrow G_3, \\ (P_1, P_2) &\mapsto g. \end{aligned}$$

We follow the convention of using additive notation for the group operations in G_1 and G_2, but multiplicative notation for G_3.

A pairing e should also satisfy the *bilinear* property: for all $P_1, Q_1 \in G_1$ and $P_2, Q_2 \in G_2$, we have

$$e(P_1 + Q_1, P_2) = e(P_1, P_2)e(Q_1, P_2),$$

and

$$e(P_1, P_2 + Q_2) = e(P_1, P_2)e(P_1, Q_2).$$

$$e(aP, bQ) = e(P, Q)^{ab}$$

for positive integers a and b.

6. Pairings on Elliptic Curves

Paring-based DLP (Skipped)

Algorithm 7.4: PAIRING-BASED-DL($\mathcal{E}, m, P, R$)

1. Find the smallest integer k for which the points of $\mathcal{E}[m]$ all have coordinates from $\mathbb{F}_{q^k}$.
2. Find $Q \in \mathcal{E}[m]$ for which $\alpha = e_m(P, Q)$ has order m.
3. Compute $\beta = e_m(R, Q)$.
4. Determine the discrete logarithm r of β with respect to the base α.

$\mathcal{E}[m]$: *m-torsion subgroup* of $\mathcal{E}$

A point P on $\mathcal{E}$ is an *m-torsion point* if $mP = \mathcal{O}$

is isomorphic to $\mathbb{Z}_m \times \mathbb{Z}_m$ for proper choices of m and q

Suitable Groups of the "Difficult" DLP

The most important settings (G, α) for the **Discrete Logarithm** problem in cryptographic applications are the following:

1. $G = (\mathbb{Z}_p{}^*, \cdot)$, p prime, α a primitive el $\boxed{p > 2^{2048}}$ ulo p;
2. $G = (\mathbb{Z}_p{}^*, \cdot)$, p, q prime, $p \equiv 1 \bmod q$ $\boxed{p > 2^{2048}}$ ent $\boxed{q > 2^{224}}$ ng order q;
3. $G = (\mathbb{F}_{2^n}{}^*, \cdot)$, α a primitive element in $\mathbb{F}_{2^n}{}^*$;
4. $G = (E, +)$, where E is an ellip $\boxed{p > 2^{224}}$ modulo a prime p, $\alpha \in E$ is a point having prime order $q = \#E/h$, where (typically) $h = 1, 2$ or 4; and
5. $G = (E, +)$, where E is an ellip $\boxed{n \approx 224}$ over a finite field $\mathbb{F}_{2^n}$, $\alpha \in E$ is a point having prime order $q = \#E/h$, where (typically) $h = 2$ or 4. (Note that we have defined ellipic curve over finite fields $\mathbb{F}_p$ only when p is a prime exceeding 3. Elliptic curves can be defined over any finite field, though a different equation is required if the field has characteristic 2 or 3.)

Outline

- **1. The ElGamal Cryptosystem**
  - **Discrete Logarithm Problem (DLP)**
  - **The ElGamal Cryptosystem**
- **2. Algorithms for the DLP**
  - **Shanks' Algorithm**
- **3. Suitable Groups for the DLP**
  - **Finite Fields & Elliptic Curves**
  - **Suitable Groups for the DLP**
- **4. Security of ElGamal Systems**
  - **Bit Security and Semantic Security**
  - **The Diffie-Hellman Problems**

Different Attack Goals

- **Total Break:** to know the private key or the secret key
- **Partial Break:** be able to decrypt a previously **unseen ciphertext** without the key, or to determine **some specific information about the plaintext** given the ciphertext, **with non-negligible probability**
- **Distinguishability of Ciphertexts:** be able to **distinguish between encryptions of two given plaintexts**, or between an encryption of a given plaintext and a random string, **with probability exceeding 1/2**

Security (against Total Break)

- **Security (Choice of proper *p*)**
  - **based on that the Discrete Logarithm problem in $\mathbb{Z}_p^*$ is infeasible, i.e., there is no polynomial-time algorithm to solve DLP**
    - ✓ ***p* should have at least 2048 bits**
    - ✓ ***p*-1 should have at least one large prime factor**
  - **The secret key *a* and the random number *k* used in encryption can not be small**
  - **The random number *k* is used once and changed for a new encryption**

Misuse of the secret k

- **If the same k is used for two encryptions:**
  1) **Choose a secret random number k in Z_{p-1}**
  2) **Encrypt 1: $e_{pk}(x_1) = (y_1, y_2)$ where $y_1 = \alpha^k \bmod p$ and $y_2 = x_1\beta^k \bmod p$**
  3) **Encrypt 2: $e_{pk}(x_2) = (z_1, z_2)$ where $z_1 = \alpha^k \bmod p$ and $z_2 = x_2\beta^k \bmod p$**

$$z_2/y_2 = x_2/x_1$$

If the plaintext x_1 is known (for example, under the known plaintext attack), then it is easy to obtain x_2.

Bit Security (against Partial Break)

Problem 7.2: **Discrete Logarithm ith Bit**

Instance: $I = (p, \alpha, \beta, i)$, where p is prime, $\alpha \in \mathbb{Z}_p{}^*$ is a primitive element, $\beta \in \mathbb{Z}_p{}^*$, and i is an integer such that $1 \leq i \leq \lceil \log_2(p-1) \rceil$.

Question: Compute $L_i(\beta)$, which (for the specified α and p) denotes the ith least significant bit in the binary representation of $\log_\alpha \beta$.

$$L_1(\beta) = \begin{cases} 0 & \text{if } \beta^{(p-1)/2} \equiv 1 \pmod{p} \\ 1 & \text{otherwise.} \end{cases}$$

- **Computing the least significant bit $L_1(\beta)$ of a DL is easy.**
- **Computing $L_i(\beta)$ (where $i \leq s$, $p-1 = 2^s t$, and t is odd) is easy.**
- **Computing $L_{s+1}(\beta)$ is (probably) difficult.**

Suppose s=1 since p-1 should have at least one large prime factor.

So $L_2(\beta)$ is difficult to obtain, while $L_1(\beta)$ is easy to compute.

Semantic Security

- <u>A cryptosystem is said to achieve *semantic security*</u> if the cryptosystem satisfies that the adversary cannot (in polynomial time) distinguish ciphertexts.

Problem 6.3:	**Ciphertext Distinguishability**
Instance:	An encryption function $f : X \rightarrow X$; two plaintexts $x_1, x_2 \in X$; and a ciphertext $y = f(x_i)$, where $i \in \{1, 2\}$.
Question:	Is $i = 1$?

Semantic Security of ElGamals

- **The basic ElGamal Cryptosystem, as described in Cryptosystem 7.1, is not semantically secure.**
  - **By the properties of the quadratic residuosity and Euler's criterion**
    - x_1 is a quadratic residue modulo p, i.e., x_1 is in $\mathsf{QR}(p) = \{x^2 \bmod p : x \in \mathbb{Z}_p{}^*\}$
    - x_2 is a quadratic non-residue modulo p
    - (y_1, y_2) is an encryption of x_1 iff β^k and y_2 are both quadratic residues or both quadratic non-residues (i.e., **$L_1(\beta^k) = 0$ and $L_1(y_2) = 0$, or $L_1(\beta^k) = 1$ and $L_1(y_2) = 1$, which are easy to compute).**
- **A variant of the ElGamal Cryptosystem is conjectured to be semantically secure if the DLP in QR(p) $\subset Z_p^*$ is infeasible**
  - **Over the subgroup (cyclic, of order q) of quadratic residues modulo p and $p=2q+1$ where p and q are prime**

The Diffie–Hellman Problems

◆ **Connection with Diffie-Hellman key agreement protocols in Section 12.2**

Problem 7.3: **Computational Diffie-Hellman** CDH

Instance: A multiplicative group $(G, \cdot)$, an element $\alpha \in G$ having order n, and two elements $\beta, \gamma \in \langle \alpha \rangle$.

Question: Find $\delta \in \langle \alpha \rangle$ such that $\log_\alpha \delta \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod{n}$. (Equivalently, given α^b and α^c, find α^{bc}.)

Problem 7.4: **Decision Diffie-Hellman** DDH

Instance: A multiplicative group $(G, \cdot)$, an element $\alpha \in G$ having order n, and three elements $\beta, \gamma, \delta \in \langle \alpha \rangle$.

Question: Is it the case that $\log_\alpha \delta \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod{n}$? (Equivalently, given α^b, α^c and α^d, determine if $d \equiv bc \pmod{n}$.)

Reductions of DHPs

DDH $\propto_T$ CDH

The first reduction is proven as follows: Let $\alpha, \beta, \gamma, \delta$ be given. Use an algorithm that solves **CDH** to find the value δ' such that

orcale

$$\log_\alpha \delta' \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod{n}.$$

Then check to see if $\delta' = \delta$.

CDH $\propto_T$ Discrete Logarithm

The second reduction is also very simple. Let α, β, γ be given. Use an algorithm that solves **Discrete Logarithm** to find $b = \log_\alpha \beta$ and $c = \log_\alpha \gamma$. Then compute $d = bc \bmod n$ and $\delta = \alpha^d$.

orcale

These reductions show that the assumption that **DDH** is infeasible is at least as strong as the assumption that **CDH** is infeasible, which in turn is at least as strong as the assumption that **Discrete Logarithm** is infeasible.

difficulty of solving DDH ≤ difficulty of solving CDH ≤ difficulty of solving DL

Security of DHPs

- **The security of DDH, CDH, DL may not be equivalent.**
  - **semantic security of the ElGamal Crypt $\leftarrow\rightarrow$ infeasibility of DDH**

    See Ex7.23
  - **ElGamal decryption $\leftarrow\rightarrow$ solving CDH**

    See next slide
  - **necessary assumption to prove the security of the ElGamal Crypt is stronger than the infeasibility of DL**

the *ElGamal Cryptosystem* in $\mathbb{Z}_p{}^*$ is not semantically secure, whereas the **Discrete Logarithm** problem is conjectured to be infeasible in $\mathbb{Z}_p{}^*$ for appropriately chosen primes p.

ElGamal decryption ⟷ CDH

ElG Dec $\propto_T$ CDH we give a proof that any algorithm that solves **CDH** can be used to decrypt ElGamal ciphertexts, and vice versa. Suppose first that ORACLECDH is an algorithm for **CDH**, and let (y_1, y_2) be a ciphertext for the *ElGamal Cryptosystem* with public key α and β. Compute

$$\delta = \text{ORACLECDH}(\alpha, \beta, y_1),$$

and then define

$$x = y_2\delta^{-1}.$$

It is easy to see that x is the decryption of the ciphertext (y_1, y_2).

CDH $\propto_T$ ElG Dec suppose that ORACLE-ELGAMAL-DECRYPT is an algorithm that decrypts ElGamal ciphertexts. Let α, β, γ be given as in **CDH**. Define α and β to be the public key for the *ElGamal Cryptosystem*. Then define $y_1 = \gamma$ and let $y_2 \in \langle\alpha\rangle$ be chosen randomly. Compute

$$x = \text{ORACLE-ELGAMAL-DECRYPT}(\alpha, \beta, (y_1, y_2)),$$

which is the decryption of the ciphertext (y_1, y_2). Finally, compute

$$\delta = y_2 x^{-1}.$$

δ is the solution to the given instance of **CDH**.

Summary

- **1. The ElGamal Cryptosystem**
  - **Discrete Logarithm Problem (DLP)**
  - **The ElGamal Cryptosystem**
- **2. Algorithms for the DLP**
  - **Shanks' Algorithm**
- **3. Suitable Groups for the DLP**
  - **Finite Fields & Elliptic Curves**
  - **Suitable Groups for the DLP**
- **4. Security of ElGamal Systems**
  - **Bit Security and Semantic Security**
  - **The Diffie-Hellman Problems**

Homework 6:

Exercises: 7.1, 7.9(show the basic idea), 7.17, 7.23.

7.9 Decrypt the ElGamal ciphertext presented in Table 7.4. The parameters of the system are $p = 31847$, $\alpha = 5$, $a = 7899$ and $\beta = 18074$. Each element of $\mathbb{Z}_n$ represents three alphabetic characters as in Exercise ~~6.12.~~

6.13

Thank you!

Questions?