Discrete logarithm and related cryptosystems - I

Diego F. Aranha

Institute of Computing
UNICAMP

More on groups

Let us remember a few facts and notation:

Theorem

Let an integer $N > 1$ with factorization $N = \prod_i p_i^{e_i}$ e $\mathbb{Z}_N^* = \{a \in \{1, \ldots, N-1\} \mid gcd(a, N) = 1\}$. Then $\mathbb{Z}_N^*$ is an abelian group under multiplication modulo N. The order of the group is given by Euler's *totient function* $\phi(N) = N \prod_i (1 - \frac{1}{p_i})$.

Corollary

Let integer $N > 1$ and $a \in \mathbb{Z}_N^*$. Then $a^{\phi(N)} = 1 \bmod N$. If N is prime and $a \in \mathbb{Z}_p$, we have that $a^{p-1} = 1 \bmod p$.

Cyclic groups

Definitions

Let $\mathbb{G}$ be a finite group (in multiplicative notation) and $g \in \mathbb{G}$. The *order* of g is the smallest positive integer i such that $g^i = 1$.

The set of elements with generator g is given by $\langle g \rangle = \{g^0, \ldots, g^{i-1}\}$ and forms a *cyclic subgroup* of $\mathbb{G}$.

Theorem

Let $\mathbb{G}$ a finite group and $g \in \mathbb{G}$ an element of order i. For any $x \in \mathbb{Z}$, we have $g^x = g^{x \bmod i}$.

Corollary

Let $\mathbb{G}$ a finite group and $g \in \mathbb{G}$ an element of order i. Then $g^x = g^y$ iff $x = y \bmod i$.

Example: $\mathbb{G}_1 = (Z_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}, \times \bmod 15)$

	$g = 1$	$g = 2$	$g = 4$	$g = 7$	$g = 8$
i	1^i	$2^i \bmod 15$	4^i	7^i	8^i
0	1	1	1	1	1
1	1	2	4	7	8
2	1	4	1	4	4
3	1	8	4	13	2
4	1	1	1	1	1
5	1	2	4	7	8
6	1	4	1	4	4
7	1	8	4	13	2
8	1	1	1	1	1

$\mathbb{G}_1$ has order 8. Note that $8^i \equiv 7^i \pmod{15}$ when i is even, and $8^i \equiv -(7^i) \pmod{15}$ when i is odd.
Exercise: Generalize (and prove) this simple fact for the remaining columns of the table.

Example: $\mathbb{G}_2 = (\mathbb{Z}_{11}^* = \{1, 2, 3, \ldots, 10\}, \times \bmod 11)$

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

$\mathbb{G}_2$ has order 10. Note that the powers of 2 are all the group elements, i.e., 2 is a generator of $\mathbb{Z}_{11}^*$. Question: Are there other generators in the hidden columns?

Cyclic groups

Lagrange Theorem

Let $\mathbb{G}$ a finite group of order N and $g \in \mathbb{G}$ an element of order i. Then $i \mid N$.

Corollary

If $\mathbb{G}$ is a group or order N, with N prime, then $\mathbb{G}$ is cyclic and all elements in $\mathbb{G}$, except the identity, are generators of $\mathbb{G}$.

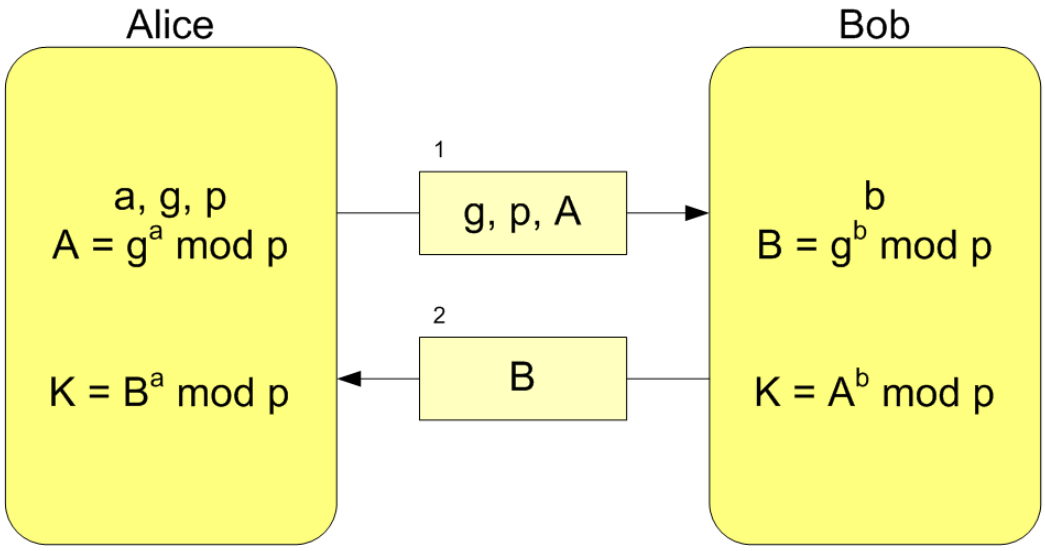
Theorem

$\mathbb{Z}_N^*$ is cyclic if and only if $N \in \{2, 4, p^k, 2p^k\}$, for p an odd prime; moreover, the number of generators of $\mathbb{Z}_N^*$ is $\phi(\phi(N))$.

Exercise: Verify these results for $\mathbb{G}_1, \mathbb{G}_2$ and prove the Theorem above.

Key agreement based on discrete logarithms

Diffie-Hellman Protocol (Diffie&Hellman, 1976)



$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$

Encryption based on discrete logarithms

Lemma

Let $\mathbb{G}$ a group and $m \in \mathbb{G}$ an arbitrarily chosen element. Choosing $g \leftarrow \mathbb{G}$ randomly and computing $g' = m \cdot g$ has the same distribution for g' as choosing $g' \leftarrow \mathbb{G}$ randomly. Thus, $\Pr[m \cdot g = g'] = 1/|\mathbb{G}|$.

Encryption based on discrete logarithms

Lemma

Let $\mathbb{G}$ a group and $m \in \mathbb{G}$ an arbitrarily chosen element. Choosing $g \leftarrow \mathbb{G}$ randomly and computing $g' = m \cdot g$ has the same distribution for g' as choosing $g' \leftarrow \mathbb{G}$ randomly. Thus, $\Pr[m \cdot g = g'] = 1/|\mathbb{G}|$.

Idea: To have something akin to symmetric construction achieving **perfect secrecy** with uniformly random key $g \in \mathbb{G}$:

$$c = Enc_g(m) = m \cdot g$$

and

$$Dec_g(c) = c \cdot g^{-1} = m.$$

Encryption based on discrete logarithms

Lemma

Let $\mathbb{G}$ a group and $m \in \mathbb{G}$ an arbitrarily chosen element. Choosing $g \leftarrow \mathbb{G}$ randomly and computing $g' = m \cdot g$ has the same distribution for g' as choosing $g' \leftarrow \mathbb{G}$ randomly. Thus, $\Pr[m \cdot g = g'] = 1/|\mathbb{G}|$.

Idea: To have something akin to symmetric construction achieving **perfect secrecy** with uniformly random key $g \in \mathbb{G}$:

$$c = Enc_g(m) = m \cdot g$$

and

$$Dec_g(c) = c \cdot g^{-1} = m.$$

Problem: To make construction asymmetric, g must be pseudo-random and recoverable by the private key holder.

Encryption based on discrete logarithms

General ElGamal encryption (ElGamal, 1984)

Key generation:

1. Choose finite group $\mathbb{G}$ with order q and a generator g of G.
2. Message space is $\mathcal{M} = \mathbb{G}$.
3. Ciphertext space is $\mathcal{C} = \mathbb{G} \times \mathbb{G}$.
4. Key space is $\mathcal{K} = \{(g, a, h) : h = g^a\}$.
5. Public key is $(\mathbb{G}, q, g, h)$. Private key is a.

Encryption of m:

1. Choose integer k uniformly random in $\mathbb{Z}_q$.
2. Compute $Enc_K(m, k) = (c_1, c_2)$, where $c_1 = g^k$, $c_2 = m \cdot h^k$.

Encryption based on discrete logarithms

General ElGamal encryption (ElGamal, 1984)

Key generation:

1. Choose finite group $\mathbb{G}$ with order q and a generator g of G.
2. Message space is $\mathcal{M} = \mathbb{G}$.
3. Ciphertext space is $\mathcal{C} = \mathbb{G} \times \mathbb{G}$.
4. Key space is $\mathcal{K} = \{(g, a, h) : h = g^a\}$.
5. Public key is $(\mathbb{G}, q, g, h)$. Private key is a.

Encryption of m:

1. Choose integer k uniformly random in $\mathbb{Z}_q$.
2. Compute $Enc_K(m, k) = (c_1, c_2)$, where $c_1 = g^k$, $c_2 = m \cdot h^k$.

Decryption: Compute $Dec_K(c_1, c_2) = c_2 \cdot (c_1^a)^{-1}$.

Important: Verify consistency!

ElGamal encryption ($\mathbb{Z}_p^*$ version)

Key generation:

1. Choose large prime p such that $p-1$ has a big factor.
2. Choose generator (primitive element) α for $\mathbb{Z}_p^*$.
3. Message space is $\mathcal{M} = \mathbb{Z}_p^*$.
4. Ciphertext space is $\mathcal{C} = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$.
5. Key space is $\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}$.
6. Public key is (p, α, β). Private key is a.

Encryption of m:

1. Choose integer k uniformly random in $\mathbb{Z}_{p-1}$.
2. Compute $Enc_K(m, k) = (c_1, c_2)$, where $c_1 = \alpha^k \bmod p$, $c_2 = m\beta^k \bmod p$.

ElGamal encryption ($\mathbb{Z}_p^*$ version)

Key generation:

1. Choose large prime p such that $p-1$ has a big factor.
2. Choose generator (primitive element) α for $\mathbb{Z}_p^*$.
3. Message space is $\mathcal{M} = \mathbb{Z}_p^*$.
4. Ciphertext space is $\mathcal{C} = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$.
5. Key space is $\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}$.
6. Public key is (p, α, β). Private key is a.

Encryption of m:

1. Choose integer k uniformly random in $\mathbb{Z}_{p-1}$.
2. Compute $Enc_K(m, k) = (c_1, c_2)$, where $c_1 = \alpha^k \bmod p$, $c_2 = m\beta^k \bmod p$.

Decryption: Compute $Dec_K(c_1, c_2) = c_2(c_1^a)^{-1} \bmod p$.

Important: Verify consistency!

Hardness assumptions

The Discrete Logarithm Problem

Let $\mathcal{A}$ be a PPT (probabilistic polynomial time) algorithm and $DLog_{\mathcal{A},GenGroup}(n)$ the execution of an experiment with $\mathcal{A}$ at security level n:

1. Group $(\mathbb{G}, q, g) \leftarrow GenGroup(1^n)$ is generated with order q (where $|q| = n$) and generator g.
2. Choose $h \in \mathbb{G}$.
3. $\mathcal{A}$ receives $\mathbb{G}, q, g, h$ and produces $x \in \mathbb{Z}_q$.
4. The experiment's output is 1 if $g^x = h$, and 0 otherwise.

Hardness condition

The *discrete logarithm problem* is hard if for all $\mathcal{A}$, there is a *negligible* function δ, such that

$$\Pr[DLog_{\mathcal{A},GenGroup}(n) = 1] \leq \delta(n).$$

Hardness assumptions

The Computational Diffie-Hellman Problem

Let $\mathcal{A}$ be a PPT algorithm and $CDH_{\mathcal{A},GenGroup}(n)$ the execution of an experiment with $\mathcal{A}$ at security level n:

1. Group $(\mathbb{G}, q, g) \leftarrow GenGroup(1^n)$ is generated with order q (where $|q| = n$) and generator g.
2. Find $h_1 = g^x \in \mathbb{G}$ and $h_2 = g^y \in \mathbb{G}$.
3. $\mathcal{A}$ receives $\mathbb{G}, q, g, h_1, h_2$ and outputs $g^{xy} \in \mathbb{Z}_q$.
4. The experiment output is 1 if $g^{xy} = h_1^y = h_2^x$, and 0 otherwise.

Hardness of the Computational Diffie-Hellman Problem

The *Computational Diffie-Hellman Problem* (CDH) is hard if for all $\mathcal{A}$, there is a negligible function δ:

$$\Pr[CDH_{\mathcal{A},GenGroup}(n) = 1] \leq \delta(n).$$

Hardness assumptions

The Decisional Diffie-Hellman Problem

Let $\mathcal{A}$ be a PPT algorithm and $CDH_{\mathcal{A},GenGroup}(n)$ the execution of an experiment with $\mathcal{A}$ at security level n:

1. Group $(\mathbb{G}, q, g) \leftarrow GenGroup(1^n)$ is generated with order q (where $|q| = n$) and generator g.
2. Find $h_1 = g^x \in \mathbb{G}$ and $h_2 = g^y \in \mathbb{G}$.
3. $\mathcal{A}$ receives $\mathbb{G}, q, g, h_1, h_2$ and outputs $g^{xy} \in \mathbb{Z}_q$.
4. The experiment output is 1 if $g^{xy} = h_1^y = h_2^x$, and 0 otherwise.

Hardness of the Decisional Diffie-Hellman Problem

The *Decisional Diffie-Hellman Problem* (DDH) is hard if for all $\mathcal{A}$, there is a negligible function δ:

$$|\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1]| \leq \delta(n).$$

Hardness assumptions

Let $\mathbb{G}$ be a group with order q and generator g.

Discrete Logarithm Problem (*DLog*)

Given $r \in \mathbb{G}$, it is hard to find an integer $0 \leq k \leq n-1$ such that $r = g^k$.

Computational Diffie-Hellman Problem (*CDH*)

Given g^x, g^y for integers x, y, it is hard to compute g^{xy}.

Decisional Diffie-Hellman Problem (*DDH*)

Given g^x, g^y, g^z for integers x, y, z, it is hard to decide if $g^z = g^{xy}$.

Important: Reductions from *CDH* to *DLog* and *DDH* to *CDH* are simple.