DISCRETE LOGARITHMS

1. DISCRETE LOGARITHMS IN A CYCLIC GROUP

Let G be a finite commutative group. For g an element of G the set $\{g^k | k \in \mathbb{Z}\}$ is the smallest subgroup of G containing g. We call it the subgroup generated by G and denote it $< g >$. If $< g >$ is the whole group G one says that g is a generator of G. If a group G admits a generator we say it is a cyclic group.

For example, let $N \geqslant 2$ be an integer. The group $(\mathbb{Z}/N\mathbb{Z}, +)$ is generated by $1 \bmod N$. A residue class $x \bmod N$ generates $\mathbb{Z}/N\mathbb{Z}$ if and only if x is prime to N. Indeed if $\gcd(x, N) = M$ then the subgroup generated by x is easily seen to be the subroup $M\mathbb{Z}/N\mathbb{Z} \subset \mathbb{Z}/N\mathbb{Z}$.

Let p be a prime integer and set $G = (\mathbb{Z}/p\mathbb{Z})^*$. This is a commutative group and even a cyclic group. Otherwise G would contain a subgroup isomorphic to $(\mathbb{Z}/n\mathbb{Z}) \times (\mathbb{Z}/n\mathbb{Z})$ for some $n \geqslant 2$. This would imply that the polynomial $x^n - 1$ has at least n^2 roots in the field $\mathbb{Z}/p\mathbb{Z}$. A contradiction. Consider the case $p = 11$ for example. The cardinality of $G = (\mathbb{Z}/11\mathbb{Z})^*$ is $\#G = 10$. Set $g = 2 \bmod 11$. We compute all powers of g and find $g^2 = 4 \bmod 11$, $g^3 = 8 \bmod 11$, $g^4 = 5 \bmod 11$, $g^5 = 10 \bmod 11$, $g^6 = 9 \bmod 11$, $g^7 = 7 \bmod 11$, $g^8 = 3 \bmod 11$, $g^9 = 6 \bmod 11$, $g^{10} = 1 \bmod 11$.

We deduce that g is a generator of G. The map $k \mapsto g^k$ defines an isomorphism between $\mathbb{Z}/\#G\mathbb{Z}$ and G. In particular g^x is a generator of G if and only if x is prime to $\#G = 10$. The generators of G are thus $g = 2 \bmod 11$, $g^3 = 8 \bmod 11$, $g^7 = 7 \bmod 11$, $g^9 = 6 \bmod 11$. The number of generators is $\varphi(10) = 4$. The map $k \mapsto g^k$ is denoted

$$\exp_g : \mathbb{Z}/\#G\mathbb{Z} = \mathbb{Z}/10\mathbb{Z} \to G = (\mathbb{Z}/11\mathbb{Z})^*$$

and called the **discrete exponential** with base g. Its table is

$k \in \mathbb{Z}/10\mathbb{Z}$	0	1	2	3	4	5	6	7	8	9
$g^k = \exp_g(k) \in (\mathbb{Z}/11\mathbb{Z})^*$	1	2	4	8	5	10	9	7	3	6

The reciprocal map of $\exp_g$ is called the **discrete logarithm** with base g and denoted

$$\log_g : G = (\mathbb{Z}/11\mathbb{Z})^* \to \mathbb{Z}/e\mathbb{Z} = \mathbb{Z}/10\mathbb{Z}.$$

Its table is

$h \in (\mathbb{Z}/11\mathbb{Z})^*$	1	2	3	4	5	6	7	8	9	10
$k = \log_g(h) \in \mathbb{Z}/10\mathbb{Z}$	0	1	8	2	4	9	7	3	6	5

Let G be a cyclic group and let n be an integer. The map $x \mapsto x^n$ defines a group homomorphism $[n] : G \to G$. This homomorphism is a bijection if and only if n is prime to $\#G$. If n is congruent to 1 modulo $\#G$ then $[n]$ is the identity. So the group $(\mathbb{Z}/\#G\mathbb{Z})^*$ acts on G. Conversely any automorphism of G sends a generator g to another generator g^x for some x that is prime to $\#G$. So the group of automorphisms of G is $\mathrm{Aut}(G) = (\mathbb{Z}/\#G\mathbb{Z})^*$.

Discrete exponentials and discrete logarithms have similar properties to their classical counterparts. For k and l in $\mathbb{Z}/e\mathbb{Z}$ one has

$$\exp_g(k+l) = g^{k+l} = g^k g^l = \exp_g(k)\exp_g(l).$$

For h_1 and h_2 in G one has

$$\log_g(h_1 h_2) = \log_g(h_1) + \log_g(h_2) \in \mathbb{Z}/\#G\mathbb{Z}.$$

If both g and h are generators of G then for any a in G one has

$$\log_h(a) = \log_g(a)/\log_g(h).$$

Using fast exponentiation one can compute $\exp_g(k) = g^k$ at the expense of $O(\log k)$ operations in G. It seems that the first occurence of this algorithm is in the Chandah-sûtra, dating back to 200 B.C. and attributed to the poet Piṅgala. See [DatSin, I,13].

On the other hand, there exists no generic algorithm to efficiently compute discrete logarithms. See [Sho]. However, discrete logarithm is easy in some specific groups e.g. the additive groups $\mathbb{Z}/N\mathbb{Z}$. The best known proven algorithms to compute discrete logarithms in $(\mathbb{Z}/N\mathbb{Z})^*$ have complexity $\exp((\log N)^{1/2+o(1)})$. These are probabilistic algorithms.

Discrete exponential is thus a good candidate **asymmetric function**. It is easy to compute and it is expected to be difficult to invert.

If G is cyclic of order e, we denote by H the set of its generators and by A its automorphism group. We have seen that A is isomorphic to $(\mathbb{Z}/e\mathbb{Z})^*$. Every automorphism has the form $[n] : G \to G$ for some invertible exponent n modulo e.

If h is a generator and $a \in (\mathbb{Z}/e\mathbb{Z})^*$ then h^a is also a generator. So the automorphism group A acts on H. If h_1 and h_2 both belong to H there exists un unique $a \in (\mathbb{Z}/e\mathbb{Z})^*$ such that $h_2 = h_1^a = [a](h_1)$. One says that A acts simply transitively on H.

Exercise. For each of the following groups say if it is cyclic and give the set of its generators $(\mathbb{Z}/5\mathbb{Z}, +)$, $((\mathbb{Z}/7\mathbb{Z})^*, \times)$, $((\mathbb{Z}/35\mathbb{Z})^*, \times)$.

□

Exercise. Set $p = 31$. Prove that $G = (\mathbb{Z}/p\mathbb{Z})^*$ has a unique sub-group of order 2. Call it G_1. Prove that G has a unique sub-group of order 3. Call it G_2. Prove that G has a unique sub-group of order 5. Call it G_3. Prove that G is the direct product of G_1, G_2 and G_3.

□

2. Identification

It is possible to build a zero-knowledge identification scheme on the difficulty of computing discrete logarithms. The following scheme is due to Schnorr.

To initialize the scheme,

- Alice chooses a cyclic group G of large enough order e. Let H be the set of generators of G and let $A = (\mathbb{Z}/e\mathbb{Z})^*$.
- Alice picks an element $h_0 \in H$ and a random element $a_{Alice} \in A$.
- Alice computes $h_{Alice} = {h_0}^{a_{Alice}}$.
- She publishes G, h_0, h_{Alice}.

Alice is the only one to know the logarithm of h_{Alice} in base h_0, that is the exponent $a_{Alice} \in A$ such that $h_{Alice} = h_0^{a_{Alice}}$.

Now if Bob wants to contact Alice, he finds G, h_0, h_{Alice} in the phonebook. He contacts Alice and ask her to prove herself.

- Alice picks a random exponent a_r in A with uniform probability.
- Alice computes $h_r = {h_{Alice}}^{a_r} = {h_0}^{a_{Alice}a_r}$ and sends h_r to Bob.
- Bob picks a random $\epsilon \in \{0, 1\}$ with uniform probability
- if $\epsilon = 0$ then Bob asks Alice which is the exponent that sends h_0 onto h_r and Alice is expected to answer $\log_{h_0}(h_r)$. She knows this logarithm because it is the product $a_r a_{Alice} \in (\mathbb{Z}/e\mathbb{Z})^*$.
- if $\epsilon = 1$ then Bob asks Alice which is the exponent that sends h_{Alice} onto h_r and Alice is expected to answer $a_r = \log_{h_{Alice}}(h_r)$.

This protocol is repeated many times. If Eve is trying to impersonate Alice she does not know $a_{Alice} = \log_{h_0}(h_{Alice})$ so she cannot at the same time know $\log_{h_0}(h_r)$ and $\log_{h_{Alice}}(h_r)$. She will fail at each round with probability $\geqslant 1/2$.

3. ElGamal encryption scheme

This is a public key encryption scheme.

Alice generates her secret key and her public key.

(1) Alice chooses a cyclic group G of order e. Let H be the set of generators of G and $A = (\mathbb{Z}/e\mathbb{Z})^*$ the set of invertible elements modulo e.
(2) Alice chooses some $h_0 \in H$ and a random $a_{Alice} \in A$. She computes $h_{Alice} = {h_0}^{a_{Alice}}$ and she publishes (G, h_0, h_{Alice}).

The secret key of Alice is $a_{Alice} \in A$. Her public key is (G, h_0, h_{Alice}).

If Bob wants to cipher a message m for Alice.

(1) Bob finds Alice's public key (G, h_0, h_{Alice}) in the phonebook.
(2) He chooses a random a_{Bob} in A and computes $k = {h_0}^{a_{Bob}}$.
(3) Bob computes $t = {h_{Alice}}^{a_{Bob}}$ and $c = m \oplus t$.
(4) Bob sends (k, c) to Alice.

Alice deciphers Bob's message (k, c).

(1) Alice computes $k^{a_{Alice}} = h_0^{a_{Bob}a_{Alice}} = h_0^{a_{Alice}a_{Bob}} = h_{Alice}^{a_{Bob}} = t$.
(2) Alice computes $m = c \ominus t$.

Note that the security of Schnorr and ElGamal schemes relies on a slightly weaker problem than discrete logarithm. This is called the Diffie and Hellman problem : given G, h_0, h_1 et h_2, find the unique h_3 such that $h_1 = h_0^{a_1}$, $h_2 = h_0^{a_2}$, $h_3 = h_0^{a_3}$ and $a_3 = a_1 a_2 \in A$.

4. Finding a generator

In order to implement the above cryptographic schemes one needs to find a large cyclic group G and a generator of it. Computing discrete logarithms in G should be difficult.

A first possibility is to pick a random large prime p such that $p-1 = 2q$ where q is prime. Pick a random y in $(\mathbb{Z}/p\mathbb{Z})^*$ and set $x = y^2$. If $x \neq 1 \bmod p$ then x generates the unique subgroup G of order q inside $(\mathbb{Z}/p\mathbb{Z})^*$.

A more general method is to pick a random large prime p such that $p - 1 = fq$ where q is prime and $f = \prod_i p_i^{e_i}$ is a product of small primes. To find such a p we pick random primes p and factor all small primes out of $p - 1$. If what remains is a prime we are done.

We now pick a random y in $(\mathbb{Z}/p\mathbb{Z})^*$ and set $x = y^f$. If $x \neq 1 \bmod p$ then x generates the unique subgroup G of order q inside $(\mathbb{Z}/p\mathbb{Z})^*$.

5. First attacks on the discrete logarithm problem

The simplest algorithm to compute discrete logarithm is exhaustive search. To find $\log_g h$ compute all successive powers of g until you find one equal to h. This algorithm works for any group but its running time is proportional to the size of the group.

There are groups where computing discrete logarithms is easy. This is the case for the additive group $G = \mathbb{Z}/N\mathbb{Z}$.

5.1. Groups of smooth order.

Set $p = 211$. This is a prime integer. Set $G = (\mathbb{Z}/p\mathbb{Z})^*$. The order of G is $e = \#G = p - 1 = 2.3.5.7$. We set

$$a_1 = 2, a_2 = 3, a_3 = 5, a_4 = 7.$$

For $1 \leqslant i \leqslant 4$ set $b_i = \prod_{j \neq i} a_i$. So

$$b_1 = 105, b_2 = 70, b_3 = 42, b_4 = 30.$$

These four integers have no common divisor. We find four integers $(c_i)_{1 \leqslant i \leqslant 4}$ such that

$$\sum_{1 \leqslant i \leqslant 4} c_i b_i = 1.$$

For example

$$c_1 = 1, c_2 = 1, c_3 = -2, c_4 = -3.$$

Set $d_i = c_i b_i$ for $1 \leqslant i \leqslant 4$. So

$$d_1 = 105, d_2 = 70, d_3 = -84, d_4 = -90.$$

Let $g = 2 \bmod 211 \in (\mathbb{Z}/p\mathbb{Z})^*$. We compute

$$g_1 = g^{105} = 210 \bmod 211, g_2 = g^{70} = 196 \bmod 211,$$
$$g_3 = g^{42} = 107 \bmod 211, g_4 = g^{30} = 171 \bmod 211.$$

We deduce that g is a generator of $(\mathbb{Z}/p\mathbb{Z})^*$. The group G contains four subgroups G_1, G_2, G_3, G_4 of orders 2, 3, 5, and 7 respectively. And

$$G = G_1 \times G_2 \times G_3 \times G_4.$$

For each $1 \leqslant i \leqslant 4$ the group G_i is generated by g_i.

Define the two maps

$$\begin{aligned} \phi : \qquad & G \longrightarrow G_1 \times G_2 \times G_3 \times G_4 \\ & x \longmapsto (x^{b_1}, x^{b_2}, x^{b_3}, x^{b_4}). \end{aligned}$$

$$\begin{aligned} \gamma : \qquad & G_1 \times G_2 \times G_3 \times G_4 \longrightarrow G \\ & (x_1, x_2, x_3, x_4) \longmapsto x_1^{c_1} x_2^{c_2} x_3^{c_3} x_4^{c_4}. \end{aligned}$$

We check that ϕ and γ are bijective and γ is the inverse map of ϕ.

Now let $h = 101 \bmod 211$. We want to compute the logarithm $\ell = \log_g(h) \in \mathbb{Z}/210\mathbb{Z}$. We compute $h^{105} = 1 \bmod 211$, $h^{70} = 196 \bmod 211$, $h^{42} = 1 \bmod 211$, and $h^{30} = 171 \bmod 211$. So

$$\phi(h) = (1 \bmod 211, 196 \bmod 211, 1 \bmod 211, 171 \bmod 211),$$
$$\phi(g) = (210 \bmod 211, 196 \bmod 211, 107 \bmod 211, 171 \bmod 211).$$

We deduce that ℓ is congruent to 0 modulo 2, congruent to 1 modulo 3, to 0 modulo 5 and to 1 modulo 7. So

$$\ell = 0.d_1 + 1.d_2 + 0.d_3 + 1.d_4 = 0.105 + 1.70 - 0.84 - 1.90 = -20 \bmod 210.$$

Indeed $g^{-20} = h$.

This method is due to Pohlig and Hellman. It quickly computes discrete logarithms when the order of the group is **smooth** meaning it only has small prime factors.

Exercise. Let $p = 331$. Show that p is a prime integer.

Factor $p - 1$ as a product of primes.

On checks that $2^{165} = 330 \bmod p$, $2^{110} = 299 \bmod p$, $2^{66} = 64 \bmod p$, $2^{30} = 1 \bmod p$, $5^{165} = 1 \bmod p$, $5^{110} = 31 \bmod p$, $5^{66} = 64 \bmod p$, $5^{30} = 180 \bmod p$.

Give a generator g of $(\mathbb{Z}/p\mathbb{Z})^*$.

Let $h = 329 \bmod p$. One checks that $h^{165} = 1 \bmod p$, $h^{110} = 299 \bmod p$, $h^{66} = 64 \bmod p$, $h^{30} = 1 \bmod p$.

Compute $\log_g(h)$ using the method of Pohlig-Hellman.

□

Exercise. Let p be the number

$$171962010545840643348334056831754301958457563589574256043877110505832165523$$
$$856261308397965147955578800999455782202456522693290629520826275682227566369411 1$$

Check that p is probably a prime using the test of Miller-Rabin.
Check that $p-1$ is a 400-smooth integer.
Write a short code that computes discrete logarithms in $(\mathbb{Z}/p\mathbb{Z})^*$.

□

5.2. **Shanks's method.** The best algorithm to compute discrete logarithms in generic groups is due to Shanks. It is called the *baby steps giant steps* method.

Assume G is the multiplicative group $G = (\mathbb{Z}/101\mathbb{Z})^*$. One checks that $g = 2 \bmod 101$ is a generator. Indeed $\#G = 100$ and the maximum divisors of 100 are $20 = 100/5$ and $50 = 100/2$. Since $2^{50} = -1 \bmod 101$ and $2^{20} = 95 \bmod 101$ we conclude that g generates G. Set $r = \lceil \sqrt{100} \rceil = 10$ and $\gamma = g^r = 14 \bmod 101$. Let $h = 48 \bmod 101$. We want to compute $\log_g h \in \mathbb{Z}/100\mathbb{Z}$. We compute the list of all γ^k for $0 \leqslant k \leqslant r-1$

k	0	1	2	3	4	5	6	7	8	9
γ^k	1	14	95	17	36	100	87	6	84	65

We now compute this list of all hg^{-l} for $0 \leqslant l \leqslant r-1$

l	0	1	2	3	4	5	6	7	8	9
hg^{-l}	48	24	12	6	3	52	26	13	57	79

These two lists have a unique common element which is 6. It corresponds to the values $k = 7$ and $l = 3$. So $\gamma^7 = hg^{-3}$ so $h = g^{73}$.

The running time of this method is the time to compute and sort the two lists. The size of these lists is $O(\sqrt{\#G})$. So the running time is $(\#G)^{1/2+o(1)}$ using a fast sorting algorithm such as heapsort.

6. SIEVING

Let p be a prime integer. Let $G = (\mathbb{Z}/p\mathbb{Z})^*$. Let g be a generator of G and h any element in G. We want to compute $\log_g h$. We pick a random integer a in $[0, p-2]$ and compute $g^a \times h = r_a \bmod p$ where $1 \leqslant r_a \leqslant p-1$. We hope r_a is a smooth integer. This means that r_a has only small prime divisors. More precisely, we have chosen a positive integer B. We say that an integer is B-smooth if all its prime divisors are $\leqslant B$. If r_a is B-smooth we write

$$g^a h = \prod_{q \leqslant B, \text{ and } q \text{ prime}} q^{e_{a,q}} \bmod p$$

where $e_{a,q}$ is the q-valuation of r_a.

Taking the logarithm in base g on either side we find

$$a + \log_g h = \sum_{q \leqslant B, \text{ and } q \text{ prime}} e_{a,q} \log_g q \bmod p-1.$$

So for each value of a for which r_a is B-smooth we get a linear relation between 1, $\log_g h$, and the $(\log_g q)_{q \leqslant B}$. Once we have collected enough such relations we compute $\log_g h$ and the $(\log_g q)_{q \leqslant B}$ by linear algebra methods in the ring $\mathbb{Z}/(p-1)\mathbb{Z}$.

For example, assume $p = 6761$. We set $g = 765 \bmod p$. We check that $p - 1 = 2^3.5.13^2$ and $g^{(p-1)/2} = -1 \bmod p$ and $g^{(p-1)/5} = 3624 \bmod p$ and $g^{(p-1)/13} = 328 \bmod p$. So g is a generator of $(\mathbb{Z}/p\mathbb{Z})^*$.

Let $h = 456 \bmod p$. We want to compute the discrete logarithm of h in base g. We pick a number of random values for a until we find the following congruences :

$$\begin{aligned}
g^{1783} \times h &= 2^3 \times 3^4 \bmod p, \\
g^{585} \times h &= 2^7 \times 3^2 \bmod p, \\
g^{726} \times h &= 2^2 \times 3^5 \bmod p, \\
g^{1116} \times h &= 2^3 \times 3^3 \bmod p, \\
g^{1393} \times h &= 2^2 \times 3^6 \bmod p.
\end{aligned}$$

We deduce the equations

$$\begin{aligned}
1783 + \log_g h &= 3\log_g 2 + 4\log_g 3 \bmod p-1, \\
585 + \log_g h &= 7\log_g 2 + 2\log_g 3 \bmod p-1, \\
726 + \log_g h &= 2\log_g 2 + 5\log_g 3 \bmod p-1, \\
1116 + \log_g h &= 3\log_g 2 + 3\log_g 3 \bmod p-1, \\
1393 + \log_g h &= 2\log_g 2 + 6\log_g 3 \bmod p-1.
\end{aligned}$$

The lattice of known relations between 1, $\log_g h$, $\log_g 2$, $\log_g 3$ is generared by the columns of the matrix

$$\begin{pmatrix} 1783 & 585 & 726 & 1393 & 6760 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 6760 & 0 & 0 \\ -3 & -7 & -2 & -2 & 0 & 0 & 6760 & 0 \\ -4 & -2 & -5 & -6 & 0 & 0 & 0 & 6760 \end{pmatrix}$$

We compute the Hermite normal form of this matrix. This produces generators for the lattice of relations in echelon form.

$$\begin{pmatrix} 6760 & 703 & 5036 & 6093 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

We deduce that $\log_g h = -703 = 6057 \bmod p-1$. We check that $g^{6057} = h$.

REFERENCES

[DatSin] B. Datta and A.N. Singh. *History of Hindu Mathematics*. Motilal Banarsi Das, Lahore, 1935.

[Gor] D. M. Gordon. *A Survey of Fast Exponentiation Methods*. J. Algorithms 27(1): 129-146 (1998)

[Sho] V. Shoup *Lower bounds for discrete logarithms and related problems*. Lecture Notes in Computer Science. 1233. Advances in Cryptology — Eurocrypt 97. Springer-Verlag. pp. 256–266 (1997).