Discrete Logarithm (1994; Shor)

Pranab Sen,
Tata Institute of Fundamental Research, Mumbai, India
www.tcs.tifr.res.in/~pgdsen

Index terms: Discrete logarithm problem, hidden subgroup problem, cryptography, quantum Fourier transform, quantum algorithms.

1 Synonyms

Logarithms in groups.

2 Problem definition

Given positive real numbers $a \neq 1$, b, the logarithm of b to base a is the unique real number s such that $b = a^s$. The notion of *discrete logarithm* is an extension of this concept to general groups.

Problem 1 (Discrete logarithm)
INPUT: *Group G, $a, b \in G$ such that $b = a^s$ for some positive integer s.*
OUTPUT: *The smallest positive integer s satisfying $b = a^s$, also known as the* discrete logarithm *of b to the base a in G.*

The usual logarithm corresponds to the discrete logarithm problem over the group of positive reals under multiplication. The most common case of discrete logarithm is when the group $G = \mathbb{Z}_p^*$, the multiplicative group of integers between 1 and $p - 1$ modulo p, p prime. Another important case is when the group G is the group of points of an elliptic curve over a finite field.

3 Key results

The discrete logarithm problem in $\mathbb{Z}_p^*$, p prime as well as in the group of points of an elliptic curve over a finite field, is believed to be intractable for randomised classical computers. That is any, possibly randomised, algorithm for the problem running on a classical computer will take time that is super-polynomial in the number of bits required to describe an input to the problem. The best classical algorithm for finding discrete logarithms in $\mathbb{Z}_p^*$, p prime is Gordon's [4] adaptation of the number field sieve which runs in time $\exp(O((\log p)^{1/3}(\log \log p)^{2/3}))$.

In a breakthrough result, Shor [9] gave an efficient quantum algorithm for discrete logarithm; his algorithm runs in time polynomial in the bit-size of the input.

Result 1 ([9]) *There is a quantum algorithm solving discrete logarithm on n-bit inputs in time $O(n^3)$ with probability at least $3/4$.*

3.1 Description of the discrete logarithm algorithm

Shor's algorithm for discrete logarithm [9] makes essential use of an efficient quantum procedure for implementing a unitary transformation known as the *quantum Fourier transform*. His original algorithm gave an efficient procedure for performing the quantum Fourier transform only over groups of the form $\mathbb{Z}_r$, r a 'smooth' integer, but nevertheless, he showed that this itself sufficed to solve discrete logarithm in the general case. In this article however, a more modern description of Shor's algorithm is given. In particular, a result by Hales and Hallgren [5] is used which shows that the quantum Fourier transform over any finite cyclic group $\mathbb{Z}_r$ can be efficiently approximated to inverse exponential precision.

A description of the algorithm is given below. A general familiarity with quantum notation on the part of the reader is assumed. A good introduction to quantum computing can be found in the book by Nielsen and Chuang [8]. Let $(G, a, b, \bar{r})$ be an instance of the discrete logarithm problem, where $\bar{r}$ is a supplied upper bound on the order of a in G. That is, there exists a positive integer $r \leq \bar{r}$ such that $a^r = 1$. By using an efficient quantum algorithm for order finding also discovered by Shor [9], it can be assumed that the order of a in G is known, that is, the smallest positive integer r satisfying $a^r = 1$. Shor's order finding algorithm runs in time $O((\log \bar{r})^3)$. Let $\epsilon > 0$. The discrete logarithm algorithm works on three registers, of which the first two are each t-qubits long where $t := O(\log r + \log(1/\epsilon))$, and the third register is big enough to store an element of G. Let U denote the unitary transformation

$$U : |x\rangle|y\rangle|z\rangle \mapsto |x\rangle|y\rangle|z \oplus (b^x a^y)\rangle,$$

where $\oplus$ denotes bitwise XOR. Given access to a reversible oracle for group operations in G, U can be implemented reversibly in time $O(t^3)$ by repeated squaring.

Let $\mathbb{C}[\mathbb{Z}_r]$ denote the Hilbert space of functions from $\mathbb{Z}_r$ to complex numbers. The computational basis of $\mathbb{C}[\mathbb{Z}_r]$ consists of the delta functions $\{|l\rangle\}_{0\leq l \leq r-1}$. Let $\mathrm{QFT}_{\mathbb{Z}_r}$ denote the *quantum Fourier transform* over the cyclic group $\mathbb{Z}_r$ defined as the following unitary operator on $\mathbb{C}[\mathbb{Z}_r]$:

$$\mathrm{QFT}_{\mathbb{Z}_r} : |x\rangle \mapsto r^{-1/2} \sum_{y \in \mathbb{Z}_r} e^{-2\pi i xy/r} |y\rangle.$$

It can be implemented in quantum time $O(t \log(t/\epsilon) + \log^2(1/\epsilon))$ up to an error of ϵ using one t-qubit register [5]. Note that for any $k \in \mathbb{Z}_r$, $\mathrm{QFT}_{\mathbb{Z}_r}$ transforms the state $r^{-1/2} \sum_{x \in \mathbb{Z}_r} e^{2\pi i kx/r}|x\rangle$ to the state $|k\rangle$. For any integer l, $0 \leq l \leq r-1$, define

$$|\hat{l}\rangle := r^{-1/2} \sum_{k=0}^{r-1} e^{-2\pi i lk/r} |a^k\rangle. \tag{1}$$

Observe that $\{|\hat{l}\rangle\}_{0\leq l\leq r-1}$ form an orthonormal basis of $\mathbb{C}[\langle a\rangle]$, where $\langle a\rangle$ is the subgroup generated by a in G and is isomorphic to $\mathbb{Z}_r$, and $\mathbb{C}[\langle a\rangle]$ denotes the Hilbert space of functions from $\langle a\rangle$ to complex numbers.

Algorithm 1 (Discrete logarithm)

INPUT: *Elements $a, b \in G$, a quantum circuit for U, the order r of a in G.*

OUTPUT: *The discrete logarithm s of b to the base a in G.*

RUNTIME: *A total of $O(t^3)$ operations including four invocations of $\mathrm{QFT}_{\mathbb{Z}_r}$ and one of U.*

PROCEDURE:

1. *Repeat Steps (a)-(e) twice obtaining $(sl_1 \bmod r, l_1)$ and $(sl_2 \bmod r, l_2)$:*

(a)	$\|0\rangle\|0\rangle\|0\rangle$	*Initialisation;*
(b)	$\mapsto r^{-1}\sum_{x,y\in\mathbb{Z}_r}\|x\rangle\|y\rangle\|0\rangle$	*Apply $\mathrm{QFT}_{\mathbb{Z}_r}$ to the first two registers;*
(c)	$\mapsto r^{-1}\sum_{x,y\in\mathbb{Z}_r}\|x\rangle\|y\rangle\|b^x a^y\rangle$	*Apply U;*
(d)	$\mapsto r^{-1/2}\sum_{l=0}^{r-1}\|sl \bmod r\rangle\|l\rangle\|\hat{l}\rangle$	*Apply $\mathrm{QFT}_{\mathbb{Z}_r}$ to the first two registers;*
(e)	$\mapsto (sl \bmod r, l)$	*Measure the first two registers;*

2. *If l_1 is not co-prime to l_2, abort;*

3. *Let k_1, k_2 be integers such that $k_1l_1 + k_2l_2 = 1$. Then, output $s = k_1(sl_1) + k_2(sl_2) \bmod r$.*

The working of the algorithm is explained below. From equation 1, it is easy to see that

$$|b^x a^y\rangle = r^{-1/2}\sum_{l=0}^{r-1} e^{2\pi i l(sx+y)/r}|\hat{l}\rangle.$$

Thus, the state in Step 1(c) of the above algorithm can be written as

$$\begin{aligned} r^{-1}\sum_{x,y\in\mathbb{Z}_r}|x\rangle|y\rangle|b^x a^y\rangle &= r^{-3/2}\sum_{l=0}^{r-1}\sum_{x,y\in\mathbb{Z}_r} e^{2\pi i l(sx+y)/r}|x\rangle|y\rangle|\hat{l}\rangle \\ &= r^{-3/2}\sum_{l=0}^{r-1}\left[\sum_{x\in\mathbb{Z}_r} e^{2\pi i slx/r}|x\rangle\right]\left[\sum_{y\in\mathbb{Z}_r} e^{2\pi i ly/r}|y\rangle\right]|\hat{l}\rangle. \end{aligned}$$

Now, applying $\mathrm{QFT}_{\mathbb{Z}_r}$ to the first two registers gives the state in Step 1(d) of the above algorithm. Measuring the first two registers gives $(sl \bmod r, l)$ for a uniformly distributed l, $0 \leq l \leq r-1$ in Step 1(e). By elementary number theory, it can be shown that if integers l_1, l_2 are uniformly and independently chosen between 0 and $l-1$, they will be co-prime with constant probability. In that case, there will be integers k_1, k_2 such that $k_1l_1 + k_2l_2 = 1$, leading to the discovery of the discrete logarithm s in Step 3 of the algorithm with constant probability. Since actually speaking only an ϵ-approximate version of $\mathrm{QFT}_{\mathbb{Z}_r}$ can be applied, ϵ can be set to be a sufficiently small constant, and this will still give the correct discrete logarithm s in Step 3 of the algorithm with constant probability. The success probability of Shor's algorithm for discrete logarithm can be boosted to at least $3/4$ by repeating it a constant number of times.

3.2 Generalisations of the discrete logarithm algorithm

The discrete logarithm problem is a special case of a more general problem called the *hidden subgroup problem* [8]. The ideas behind Shor's algorithm for discrete logarithm can be generalised in order to yield an efficient quantum algorithm for hidden subgroups in abelian groups (see e.g. [1] for a brief sketch). It turns out that finding the discrete logarithm of b to the base a in G reduces to the hidden subgroup problem in the group $\mathbb{Z}_r \times \mathbb{Z}_r$ where r is the order of a in G. Besides discrete logarithm, other cryptographically important functions like integer factoring, finding the order of permutations as well as finding self-shift-equivalent polynomials over finite fields can be reduced to instances of hidden subgroup in abelian groups.

4 Applications

The assumed intractability of the discrete logarithm problem lies at the heart of several cryptographic algorithms and protocols. The first example of public-key cryptography, namely the Diffie-Hellman key exchange [2], uses discrete logarithms, usually in the group $\mathbb{Z}_p^*$ for a prime p. The security of the U. S. national standard Digital Signature Algorithm (see e.g. [7] for details and more references) depends on the assumed intractability of discrete logarithms in $\mathbb{Z}_p^*$, p prime. The ElGamal public key cryptosystem [3] and its derivatives use discrete logarithms in appropriately chosen subgroups of $\mathbb{Z}_p^*$, p prime. More recent applications include those in elliptic curve cryptography [6], where the group consists of the group of points of an elliptic curve over a finite field.

5 Cross references

Factoring (00002), Abelian Hidden Subgroup Problem (00004).

6 Recommended reading

[1] G. Brassard and P. Høyer. An exact quantum polynomial-time algorithm for Simon's problem. In *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*, pages 12–23, 1997.

[2] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976.

[3] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[4] D. Gordon. Discrete logarithms in $\mathrm{GF}(p)$ using the number field sieve. *SIAM Journal on Discrete Mathematics*, pages 124–139, 1993.

[5] L. Hales and S. Hallgren. An improved quantum Fourier transform algorithm and applications. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 515–525, 2000.

[6] D. Hankerson, A. Menezes, and S. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag, 2004.

[7] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. C. R. C. Press, 1997.

[8] M. Nielsen and I. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000.

[9] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.