Noname manuscript No.
(will be inserted by the editor)

Comparison of the complexity of Diffie-Hellman and discrete logarithm problems.

Mikhail Cherepniov

Received: date / Accepted: date

Abstract The article presents an algorithm for solving the discrete logarithm problem with an oracle, solving the Diffie-Hellman problem. Certified the discrete logarithm problem is considered. The Diffie-Hellman oracle works with elements of the original group, but with new group operations that are compositions of the Diffie-Hellman oracle. In particular, a universal (generic) algorithm can be substituted as the Diffie-Hellman oracle. The result is improved since 1996 - the degree of logarithm in the estimation of the complexity of the algorithm presented is reduced to one. Of course, this does not affect the property of polynomial reduction of the considered problems to each other, but excludes from the evaluation in a sense unnecessary terms.

Keywords Diffie-Hellman problem · discrete logarithm problem · Pratt tree lengh

1 Introduction

In 1996 the author published [1] a constructive deterministic algorithm solving the discrete logarithm problem by oracle, solving the Diffie-Hellman problem in some subsets of the original group connected by new group operations. In 1997 V. Shoup [2] showed that the algorithm solving the Diffie-Hellman problem in an arbitrary group (generic algorithm) cannot be simple. However, this does not detract from the previous result, since the representation of the elements is fixed and the group operations have a special appearance. In 2011 N. Koblitz, A. Menezes, I. E. Shparlinski published in Vietnam Journal of

The work is supported by RFBR grant 18-29-03124

M. Cherepniov
RF, Moscow, Leninskiye hills, 1, 52
Tel.: +7-495-9304386
E-mail: cherepniov@gmail.com

Mathematics an article [3], one of the main results of which (Theorem 2.4) and its proof repeats the author's 1996 result and his proof, although there is no reference to the 1996 article. In 2019, the result of 1996 was extended by the author of this paper to the integer factorization problem [4]. In view of such a great interest in this topic, the 1996 theorem here is somewhat strengthened, which, however, is not of fundamental importance for the connection between the complexity of discrete logarithm and Diffie-Hellman problems.

2 Definitions and formulations

Let $G(t, m)$ be arbitrary commutative cyclic group of order m with an operation that we will continue to denote $+$, requiring for its execution t bitwise operations, and let $L(t, m)$ denote the upper estimate of the number of bit operations required to solve the discrete logarithm problem in the group $G(t, m)$, that is, known $a, b \in G(t, m)$ such that

$$a = nb, \tag{1}$$

find $n \in Z_m$. Here nb is $b + \cdots + b$, where the element b repeats n times.

Without limiting the generality of further reasoning, we consider b be a generator of the group $G(t, m)$.

Let $D(t, m)$ be the upper estimate of the number of bit operations required to solve the Diffie-Hellman problem: known a_1, a_2 and b such that $a_1 = n_1 b, a_2 = n_2 b$, find

$$a_3 = (n_1 n_2)b. \tag{2}$$

We will assume that $D(t, m) \geq \max\{t, u(m)\} \log m$ and does not decrease by m, where $u(m)$ is the upper bound on the number of bit operations required to multiply modulo m.

Let $D^*(t, m) \geq \max\{t, u(m)\} \log m$ — also a non-decreasing by m upper estimate of the quantity bitwise operations required to solve the same problem using algorithms, the number of bit operations in which satisfies inequality

$$D^*(t, m) \leq tD^*(C, m),$$

for some absolute constant C (for example, those algorithms that use only operations, which complexity is no more than the complexity of the group operation).

We will prove our results for the case of a certified discrete logarithm problem, that is, when decomposition on primesw $p-1, p_i-1, i = 1, \ldots, r, q_{ij}-1$ for primes $q_{ij} \mid p_i - 1$ and so on as well primitive roots by modules $p, p_i, i = 1, \ldots, r, q_{ij}$ etc. are assumed to be known. As shown in [1], the estimates obtained remain plausible in the general case as well.

For an arbitrary $m = \prod_{i=1}^{r} {p_i}^{\alpha_i}$ consider prime factorizations of $p_i - 1$, $i = 1, \ldots, r$, prime factorizations of $q_{ij} - 1$ for primes $q_{ij} \mid p_i - 1$, and so on. The resulting branching is called the Pratt tree [5] of the number m,

and the occurring primes are its nodes. Denote $s = s(m)$ length of the largest branch of the considered tree.

Note that the condition $s(p) = 1$ for a prime p means that it is a Fermat prime. To date, only five Fermat primes are known (3, 5, 17, 257, 65537). It has been proved that the remaining Fermat primes, if they exist at all, contain more than a billion bits in their record. The following theorem is proved for natural m, in which the Pratt tree does not contain any Fermat primes except the five indicated ones.

Theorem 1 *For the certified discrete logarithm problem under the condition that $D(t,p) \geq \max\{t, u(p)\} \log p$, and the Pratt tree of the number m does not contain any Fermat primes except the five specified, we have*

$$\underbrace{D(\dots D(D}_{s} L(t,m) \leq s \log m \underbrace{D(\dots D(D}_{s}(m)(t,m),m)\dots,m), \tag{3}$$

where on the right is s - multiple iteration of the function $D(t,m)$,

$$L(t,m) \leq ts \log m (D^*(C,m))^s. \tag{4}$$

The multiplicative constants appearing in our arguments will be considered as taken into account in the basis of logarithms, which we will not specify.

A similar estimate with $s = 1$, but only for prime m, satisfying the smoothness condition was obtained earlier in [7].

3 Proof of the theorem

For simplicity, consider first the case where m is a prime number, $m = p$. Without limiting generality, we can assume that in the equation (1) $n \not\equiv 0 \pmod p$, and b is not an additive unit element of group $G(t,p)$. We introduce on the set of elements

$$\{nb \mid n \not\equiv 0 \pmod p\} \tag{5}$$

a new group operation specified by equality

$$n_1 b \oplus n_2 b = (n_1 n_2) b.$$

Number of bit operations, required to perform this operation is, obviously, equal to $D(t,p)$. Since non-zero residues on a prime module form a cyclic group by multiplication, we get with this operation on the set (5) a cyclic group whose order is $d = \prod_{i=1}^{r} {p_i}^{\alpha_i} \mid \varphi(p) = p-1$, where $p_i, \quad i = 1,\dots,r$, are different primes, and $r = \nu(p-1)$, where $\nu(p-1)$ is the number of different prime divisors of $p-1$. We denote this group $G(D(t,p), p-1)$. It is clear that b is a unit of this group.

Let g be the primitive root modulo p, then gb - is a generator of the group $G(D(t,p), p-1)$. This group is the direct sum of cyclic groups of order ${p_i}^{\alpha_i}, i = 1,\dots,r$, with generators respectively $g_i b$, where

$$g_i \equiv g^{\frac{p-1}{{p_i}^{\alpha_i}}} \pmod{p}, \qquad i = 1, \ldots, r.$$

Let $n = g^v \pmod{p}$. We will look for $v \pmod{d}$, and then by $2u(p)\log p$ bit operations we will receive n. The equation (1) can be rewritten in view

$$a = \bigoplus_i {g_i}^{n_i} b = {g_1}^{n_1} b \oplus \ldots \oplus {g_r}^{n_r} b,$$

where

$$1 \le n_i \le {p_i}^{\alpha_i}, \quad v \equiv \frac{p-1}{{p_i}^{\alpha_i}} n_i \pmod{{p_i}^{\alpha_i}}, \quad i = 1, \ldots, r,$$

or

$$a = n_1 \cdot (g_1 b) \oplus \ldots \oplus n_r \cdot (g_r b),$$

where operation $\cdot$ (operation of multiplication elements of the group $G(D(t,p), p-1)$ by natural numbers) is defined for the operation $\oplus$ just as simple multiplication by natural numbers is defined for operation $+$, namely,

$$l \cdot (kb) = \underbrace{kb \oplus \ldots \oplus kb}_{l} = k^l b,$$

where the $\oplus$ operation binds l elements. Observe, that l, in this case, can be considered as a residue modulo d.

The number of bit operations required for such multiplication with the application of the binary algorithm is obviously not superior than $D(t,p)\log n$. Now to determine the number n in the expression (1) we will look for the numbers n_i and then find n by the formula

$$n \equiv \prod_{i=1}^{r} {g_i}^{n_i} \pmod{p}.$$

For each $i = 1, \ldots, r$ using a binary algorithm we find

$$a_i = \frac{p-1}{{p_i}^{\alpha_i}} \cdot a = \frac{p-1}{{p_i}^{\alpha_i}} n_i \cdot (g_i b).$$

This will require no more than $\nu(p-1)\log(p-1)D(t,p)$ bitwise operations. Still $\nu(p-1)\log(p-1)(u(p)+t)$ bitwise operations are required for computing $g_i, g_i b$ with known g and b for all $i = 1, \ldots, r$. Then, solving the discrete logarithm problem in a cyclic subgroup of order ${p_i}^{\alpha_i}$ of the group $G(D(t,p), p-1)$ generated by the element $g_i b$, find such t_i, what

$$a_i = t_i \cdot (g_i b). \tag{6}$$

This would require $L(D(t,p), {p_i}^{\alpha_i})$ bits operations. Moreover, it is clear that

$$n_i \frac{p-1}{{p_i}^{\alpha_i}} \equiv t_i \pmod{{p_i}^{\alpha_i}}.$$

Using Euclid's algorithm, you can inverse $\frac{p-1}{{p_i}^{\alpha_i}}$ modulo ${p_i}^{\alpha_i}$ for $u(p)\log {p_i}^{\alpha_i}$ bit operations, so the number of bit operations required to find each n_i at known $a_i, g_i b$, does not exceed

$$L(D(t,p),{p_i}^{\alpha_i}) + u(p)\log {p_i}^{\alpha_i}. \tag{7}$$

The equation (6) can be solved in another way, namely by multiplying its on ${p_i}^{\alpha_i - 1}$, then no more than for $L(D(t,p),p_i)$ bitwise operations, we get t_i modulo p_i.

Let $t_i = t_{i0} + p_i t_{i1}, 0 \le t_{i0} < p_i$, then $a_i = (t_{i0} + p_i t_{i1}) \cdot (g_i b)$. Since when multiplied the exponents add up, then

$$a_i \oplus (p_i - t_{i0}) \cdot (g_i b) = (t_{i1} + 1) \cdot ({g_i}^{p_i} b) \quad \text{and so on.}$$

Number t_{i1} modulo ${p_i}^{\alpha_i - 1}$ is defined in no more than $L(D(t,p),{p_i}^{\alpha_i - 1})$ bitwise operations. Required elements ${g_i}^{{p_i}^{\beta_i}} b, {p_i}^{\beta_i} \cdot a_i, \beta_i = 1, \ldots, \alpha_i$ we can obtain consistently by $\alpha_i \log p_i D(t,p)$ bit operations. Thus, in equality (7) we can replace $L(D(t,p),{p_i}^{\alpha_i})$ by

$$\alpha_i(\alpha_i \log p_i D(t,p) + L(D(t,p),p_i)).$$

Here $\alpha_i^2 \log p_i D(t,p)$ - calculation of elements of the form $(p_i - t_{i0}) \cdot (g_i b)$.

Summing up, we obtain the following recurrent formula for estimates of the complexity of the certified discrete logarithm problem:

$$L(t,p) \le D(t,p)\log^2 p + \sum_{{p_i}^{\alpha_i} \| p-1} \alpha_i L(D(t,p),p_i) \tag{8}$$

(we propose that $D(t,p) \ge \max\{t, u(p)\}\log p$)

Repeating the same reasoning with the replacement of p by p_i , etc., in finally we come to the problem of discrete logarithm in the group of order 2, which can be solved by brute force.

The statement of the theorem is proved by induction on s. In view of the assumption made about the Pratt tree under the condition of our theorem, at $s(m) = 1$, the complexity of the discrete logarithm problem does not exceed the constant. Thus the basis of induction is trivial.

When considering the induction step ($s \ge 2$), we take advantage of the inequality (8) and inequality (3) with s replaced by $s-1$. Then under conditions of the theorem $D(t,p) \ge \max\{t, u(p)\}\log p$, and we get that

$$L(t,p) \le D(t,p)\log^2 p + \sum_{{p_i}^{\alpha_i} \| p-1} \alpha_i (s-1) \log p_i \underbrace{D(\ldots D(D}_{s}(t,p),p)\ldots) \le$$

$$\le D(D(t,p),p)\log p + \sum_{{p_i}^{\alpha_i} \| p-1} \alpha_i (s-1) \log p_i \underbrace{D(\ldots D(D}_{s}(t,p),p)\ldots) \le$$

$$\leq \underbrace{D(\ldots D(D}_{s}(t,p),p)\ldots)\left(\log p+(s-1)\sum_{p_i{}^{\alpha_i}\|p-1}\alpha_i\log p_i\right)\leq$$
$$\leq s\log p\,\underbrace{D(\ldots D(D}_{s}(t,p),p)\ldots),$$

where in formulas with dots are s-multiple iterations of the function $D(t,p)$.

The case of an arbitrary,

$$m=\prod_{i=1}^{r}{q_i}^{\alpha_i},$$

reduces to the above by multiplying the equality (1) by $\frac{m}{q_i}$ etc., similar to the above solution of equation (6).

The proof of inequality (4) is obvious. Theorem 1 is proven.

Note that $D(x,y)$ into the inequality (3) (or $D^*(x,y)$ into the inequality (4)) may be replaced by an upper bound (satisfying the condition of the theorem) of the number of bit operations of any mass algorithm (that is, such, which can be applied to any cyclic group of order y with a group operation requiring x bit operations for its implementations.)

Some results related to the $s(m)$ function, introduced in [1], can be found in the works of [8–10]. However, obtaining upper estimates of $s(m)$ that would give a nontrivial relationship between the complexity of discrete logarithm and Diffie-Hellman problems is still an unsolved problem.

References

1. Cherepnev M. A., On the relationship between the complexity of discrete logarithm and Diffie-Hellman problems, Discrete mathematics, vol. 8, n.3, p.22-30 (1996)
2. Shoup V., Lower bounds for discrete logarithms and related problems, Advances in Cryptology - Eurocrypt'97, Lecture Notes in Computer Science, v.1233, pp. 256-266, Springer-Verlag, (1997)
3. Koblitz N., Menezes A., I. E. Shparlinski, Discrete Logarithms, Diffie-Hellman, and Reductions, Vietnam Journal of Math., v.39, n.3, pp.267-285 (2011)
4. Cherepnev M. A., Obtaining an upper bound on the whole factorization problem involving the complexity of the Diffie-Hellman problem, Discrete mathematics, accepted for printing.
5. V. Pratt, Every prime has a succinct certificate, SIAM J. Comput., v.4, n.3, pp. 214-220 (1975)
6. McCurley K. S., A key distribution system equivalent to factoring, J. Cryptology v.1, n.2, pp.95-105 (1988)
7. Maurer U. M., Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms, Crypto'94, p. 271-281 (1994)
8. Cherepniov M.A., Some properties of large prime divisors of the number p-1, Math. notes, v.80, n.6, pp.920-925 (2006)
9. Cherepniov M.A., On the magnitude of prime divisors of numbers of the form p-1, Materials of the international conference Mabit-04, Moscow, pp. 243-246 (2005)
10. Ford K., Konyagin S.V., Luka F., Prime chains and Pratt trees, Geom. Funct. Anal., v.20, pp.1231-1258 (2010)