ON THE COMPLEXITY OF THE DISCRETE LOGARITHM AND DIFFIE-HELLMAN PROBLEMS

IAN F. BLAKE AND THEO GAREFALAKIS

ABSTRACT. The discrete logarithm problem plays a central role in cryptographic protocols and computational number theory. To establish the exact complexity, not only of the discrete logarithm problem but also of its relatives, the Diffie-Hellman problem and the decision Diffie-Hellman problem, is of some importance. These problems can be set in a variety of groups, and in some of these they can assume different characteristics. This work considers the bit complexity of the Diffie-Hellman and the decision Diffie-Hellman problems. It was previously shown by Boneh and Venkatesan that it is as hard to compute $O(\sqrt{n})$ of the most significant bits of the Diffie-Hellman function, as it is to compute the whole function, implying that if the Diffie-Hellman function is difficult then so is computing this number of bits of it. The main result of this paper is to show that if the decision Diffie-Hellman problem is hard then computing the two most significant bits of the Diffie-Hellman function is hard. To place the result in perspective a brief overview of relevant recent advances on related problems is given.

1. INTRODUCTION

In their landmark paper [14] Diffie and Hellman introduced the following key exchange protocol, the Diffie-Hellman (DH) protocol, that is in universal use. For two users, Alice and Bob to derive a common key in a finite cyclic group G, $|G| = n$, with generator g, $G = \langle g \rangle$, they respectively choose, at random, integers $a, b \in_R [1, n]$ and exchange g^a, g^b. Each is able to compute g^{ab} from which a common key may be derived. It is assumed here that an adversary, knowing G, g^a and g^b, is unable to compute g^{ab}. The computation of g^{ab} from knowledge of g^a and g^b is referred to as the Diffie-Hellman or DH problem and a function that realizes this the DH function. For security something more is needed than the difficulty of computing the DH function: the amount of information the adversary is able to derive from the available information should be limited. Thus if it is intended to use the 128 most significant bits of the exchanged secret, for a common block cipher key, then these bits should be as difficult to compute as the DH function.

The problem is intimately connected with the discrete logarithm (DL) problem in G: given g, $y = g^x$ find x. Clearly if one is able to find discrete logarithms in G then one can break the DH protocol and the DL problem is at least as hard as the DH problem.

A related problem is the decision DH problem, DDH, which asks whether, given the triple g^a, g^b and g^c, $c \equiv ab \pmod{|G|}$.

These problems, and related ones, are of central importance to cryptography and certain aspects of them are considered in this work. To place the results in perspective a brief overview of the recent advances on these problems is given. The problems are defined more formally in the next section where the relationships between the problems are explored as

Date: November 8, 2002.
Key words and phrases. Cryptography, discrete logarithms.

well as the types of groups that have been considered as the most interesting for these problems.

The complexity of the problems in cyclic groups with no assumed representation, the so-called generic groups, is discussed in Section 3. Here it is of interest to note that with no specific group representation assumed, the complexity of the problems can be shown to be lower bounded by $\Omega(\sqrt{p})$, where p is the largest prime divisor of the group order. However, the introduction of some smoothness criteria for group elements immediately allows the introduction of an index calculus method and a subexponential complexity.

Section 4 considers recent results on the statistical distribution of random versus Diffie-Hellman triples. Clearly any possibility of distinguishing such triples would yield information that might be exploited to break the problem.

The main contribution of the work is considered in Section 5, the bit complexity of the DH problem. Here it is shown that if the DDH problem is hard then computing the two most significant bits of the DH problem is hard. Thus an oracle that is able to return the two most significant bits of the DH function can be used to determine the DDH problem with high probability. The implication of the result is that if the DDH problem is indeed computationally infeasible then so is computing the two most significant bits of the DH function. The final section of the paper considers the possibility of strengthening this result further. It was previously shown by Boneh and Venkatesan [4] that it is as hard to compute $O(\sqrt{n})$ of the most significant bits of the Diffie-Hellman function, as it is to compute the whole function, implying that if the Diffie-Hellman function is difficult then so is computing this number of bits of it. Both results contribute to the understanding of the relative difficulty of the DH and DDH functions.

2. The discrete logarithm and Diffie-Hellman problems

For consistency we define again the *discrete logarithm* (DL) problem as follows:

Definition 2.1. The DL problem for a cyclic group G with respect to a given generator g is, given y and g find an $x \in \mathbb{Z}$ such that $y = g^x$. Such an x is defined modulo $|G|$. We call the smallest such non-negative x the discrete logarithm of y with respect to g.

The instance of most concern in this work will be a subgroup of prime order of the multiplicative group $\mathbb{F}_p^*$. In this case, if α is a primitive element of $\mathbb{F}_p^*$ then $g = \alpha^a$ for some $a|p-1$ and the case of most cryptographic interest is when g has prime order in $\mathbb{F}_p^*$.

A second problem of interest is the *Diffie-Hellman* (DH) problem:

Definition 2.2. The DH problem for a cyclic group G with respect to a given generator g is, given g, g^a and g^b, with $a,\ b \in \mathbb{Z}$, compute g^{ab}.

Certainly if one can compute discrete logarithms one can solve the DH problem. However for the converse it is unknown. For most groups it is believed that the DL and DH problems have similar complexities. Some results to this effect have been obtained in [5, 29, 30, 32].

A related concept that will be of interest in later sections is the notion of a *DH oracle* (e.g [33]):

Definition 2.3. A DH oracle for for a group G with respect to a given g takes as inputs g^a and g^b and returns, without computational cost, g^{ab}.

It is also convenient to speak of a DH function which is an incarnation of a DH oracle which returns g^{ab} given g^a and g^b as input. One might also consider the notion of random DH oracles where the oracle returns the correct answer with a certain probability, but the above will be sufficient for the purposes of interest here. A more formal approach to these problems is taken in [7] where again randomized algorithms are used.

Maurer [33] also introduced the notion of a squaring DH oracle which on input g^a returns g^{a^2}. It is noted there that this is equivalent to a DH oracle in a group whose order is known.

A problem related to the DH problem is the *decision DH problem* (DDH) defined as follows:

Definition 2.4. The DDH problem for a cyclic group G with respect to a given generator g is: for random integers $a,\ b,\ c \in \mathbb{Z}$, given $\{g^a,\ g^b,\ g^c\}$ decide whether $c \equiv ab \pmod{|G|}$ or not.

Clearly, in general, DDH is no harder than DH and DH is no harder than DL. As will be shown later however, something more can be said for certain groups.

The DL problem in cyclic subgroups of $\mathbb{F}_p^*$ of prime order has perhaps received the most attention. The most effective attacks on this case have been a variety of so-called index calculus attacks, the latest being the *number field sieve* version. In general these algorithms proceed in two stages; the first finds the discrete logarithms of elements in a factor base, a set of elements chosen so that it can be conveniently determined if an arbitrary element can be decomposed into elements of it. In the second stage the element whose discrete logarithm is required is operated upon to find a related element that decomposes into elements in the factor base. The notion of decomposability of an element into the factor base often leads to a notion of smoothness which is readily available for integers and polynomials but is noticeably missing from other instances of the problem.

Without elaborating further it is noted that the complexity (generally in both time and storage) is of the form

$$L_p(a, c) = \exp(c(\log p)^a (\log\log p)^{1-a})$$

for some small constant c depending on the algorithm. The current best algorithms for both integer factorization and discrete logarithms have a constant $c = (64/9)^{1/3}$ and $a = 1/3$ (the same complexity as for factoring integers). The L function above is referred to as subexponential: (if $a = 1$ the algorithm is exponential in $\log p$ and if $a = 0$ it is polynomial in $\log p$. Experience seems to indicate that when the group of the DL problem admits a notion of smoothness, and the density of smooth group elements is appropriate, the complexity is invariably subexponential and otherwise $O(\sqrt{p})$. In this last case an algorithm such as Pollard's ρ algorithm or Shanks baby-step-giant-step algorithm achieves this complexity in arbitrary cyclic groups of order p.

Apart from cyclic subgroups of $\mathbb{F}_p^*$ a variety of other groups have been considered as suitable for the DL problem. From the above discussion of course it is of particular interest to find groups that do not satisfy smoothness criteria and hence have the square root complexity, the worst case possible (and best for cryptographic application). A brief overview of the groups considered (elliptic curves, Jacobians of hyperelliptic curves, Abelian varieties and ideal class groups) and recent advances in these groups, is given.

For simplicity, we consider finite fields $\mathbb{F}_q$ of characteristic $p > 3$. An elliptic curve over the finite field $\mathbb{F}_q$, can be taken, in affine coordinates, without loss of generality, to be of the form:

$$y^2 = x^3 + ax + b, \quad a,\ b \in \mathbb{F}_q.$$

We define

$$E_{a,b}(\mathbb{F}_q) = \{P = (u, v) \in \mathbb{F}_q^2 \mid v^2 = u^3 + au + b\} \cup \{\mathcal{O}\},$$

where $\mathcal{O}$ is a special point (called the point at infinity). The number of isomorphism classes of such curves is known [36] depending on the value of q mod 12. Similar statements can be made about the cases of characteristic 2 or 3. In all cases, there is a natural point addition on the curve that yields an Abelian group and the order of this group, including the group identity, the point at infinity, is denoted $|E_{a,b}(\mathbb{F}_q)|$. In general the number of points on a curve obeys Hasse's theorem:

$$|E_{a,b}(\mathbb{F}_q)| = q + 1 - t, \ |t| \leq 2\sqrt{q}.$$

For the case of $q = p > 3$, an odd prime, it can be shown that (see [36]) that there is at least one curve of order $q + 1 - t$ for each value of $|t| \leq 2\sqrt{p}$. In general a curve is called supersingular if $\mathrm{char}(\mathbb{F}_q) = p|t$ and otherwise nonsupersingular. For the case that $q = p$ a supersingular curve has $p + 1$ points, a case that will be of some cryptographic importance later. As noted, for cryptographic purposes one would like to use a group (in this case, additive) of prime order and the generation of suitable curves for the case of $q = p > 3$ has occupied considerable attention. It is not strictly necessary for the curve order to be prime as a prime order subgroup could be used but for reasons of efficiency there is a natural cryptographic preference for the curve order itself to be prime which, for this case, is always possible (for curves over fields of characteristic 2 one can show that the curve order is always divisible by either 2 or 4). To determine suitable curves requires the number of points on the curve be counted and this has generated a series of papers on the problem of increasing efficiency. The original work of Schoof [45] gave a polynomial time algorithm (in $\log p$) to count points. This was improved considerably by contributions to the problem of Elkies and Atkins to result in the so-called SEA algorithm (for an account of this see [3]). More recently Satoh [46], gave a new algorithm for small characteristic, which has since been improved, (see Fouquet et al [18] and many others) to the point where point counting can be done very rapidly on curves of orders far in excess (at this time) of those contemplated for cryptographic use.

The discrete logarithm problem on an elliptic curve is, given a point P on the curve, along with the curve coefficients and a point $Q = k \cdot P, \ k \in \mathbb{Z}$, find k. A key aspect of this problem is that no concept of smoothness has been formulated in such a group which rules out index calculus techniques discussed above and, by the currently best available algorithms, the complexity of solving this problem is $O(\sqrt{p})$, as might be achieved with Shank's baby-step-giant-step or Pollard's rho algorithm. It should be noted that when the curve order divides $p^k - 1$, the order of the number of nonzero elements in a small extension of $\mathbb{F}_p$, it may be possible to map the elliptic curve problem into an ordinary finite field discrete logarithm problem where the algorithm is again subexponential (see [17, 16, 20, 37]). In practice, this condition (the MOV condition) is always checked and curves that satisfy it are avoided.

A natural generalization of the group of an elliptic curve to consider is a hyperelliptic curve [26]. Such a curve $\mathcal{C}$ of genus g is given by the equation

$$y^2 + h(x)y = f(x)$$

where $h(x) \in \mathbb{F}_q[x]$ is of degree at most g and $f(x) \in \mathbb{F}_q[x]$ is monic of degree $2g + 1$. We assume the curve is smooth at all points, i.e., the partial derivatives are not simultaneously zero at any solution to the equation (and the point at infinity is smooth). The set of such solutions (points) does not form a group and it is necessary to pass to the *Jacobian* of the curve for a suitable group. To define this structure a *divisor* is defined as a formal sum $\sum m_i P_i, \ m_i \in \mathbb{Z}$, where P_i is a point on the curve over the algebraic closure of the field $\bar{\mathbb{F}}_q$ and only a finite number of the terms of the sum are nonzero. The degree of a divisor is

$\sum m_i$ and the order of a divisor at the point P_i is m_i. Let $\mathbb{D}$ denote the set of divisors of the curve and $\mathbb{D}^0$ those of degree zero. For every bivariate rational function g on the curve $\mathcal{C}$ one defines its divisor

$$div(g) = \sum_{P \in \mathcal{C}} ord_P(g) P$$

called a principal divisor, where $ord_P(g)$ is the order of the zero or pole of g at point P on the curve. Since $div(g) \in \mathbb{D}^0$, the principal divisors form a subgroup of $\mathbb{D}^0$, denoted $\mathbb{P}$. The quotient $\mathbb{D}^0/\mathbb{P}$ is called the Jacobian $\mathbb{J}$ of the curve. The Jacobian has been the subject of considerable recent attention for cryptographic use, and the problems associated with the use of elliptic curves for cryptography, especially point counting and efficient arithmetic, are present for this case as well. The question of efficient arithmetic is particularly challenging since computation is in a quotient group and the question of element representation arises. As for the elliptic curve case, one can construct curves over extension fields by using the zeta function approach although for a genus g curve one must first have available the number of points on the first g extensions of the base field. In analogy with the Hasse theorem, one can show [26] that the order of the Jacobian of a curve $\mathcal{C}/\mathbb{F}_q$ of genus g, over $\mathbb{F}_{q^r}$, lies in the range

$$[(q^{r/2} - 1)^{2g}, (q^{r/2} + 1)^{2g}].$$

A recent result of Gaudry [21] shows that it is indeed possible to find a concept of smoothness for this case, and it can be used to reduce the overall complexity of finding discrete logarithms in a Jacobian of a curve of genus g to

$$O(q^2 + g!q)$$

polynomial time operations. For a fixed genus g the complexity of the algorithm takes the form

$$O(q^2).$$

The implication of this important result is that hyperelliptic curves of genus greater than 4 need not be considered since elliptic curves over the degree 4 extension of the base field will offer the same security and it is generally felt that arithmetic over the Jacobian is unlikely to be faster than on the corresponding elliptic curve.

Continuing to look at suitable structures for the DL problem, the natural extension to elliptic curves is *Abelian varieties.* These are simply higher dimensional generalizations of elliptic curves whose points form an Abelian group and, again, the security of the system relies on the difficulty of the DL problem in this group. Again, the attraction is that there is no known subexponential algorithm for this group and the thought is that, for the same level of security as a fixed elliptic curve, the underlying field can be smaller that may allow for faster arithmetic. It is not yet clear if this can in fact be realized. The notion of a supersingular variety will also prove useful for certain cryptographic purposes described below.

The last group to be discussed as a setting for the discrete logarithm problem is the ideal class group of an algebraic number field. and this application is briefly described ([9], [10], [47]), following mainly [10].

Let D be a square-free integer and let $\mathcal{K} = \mathbb{Q}(\sqrt{D})$ be the quadratic field obtained by adjoining $\sqrt{D}$ to the rationals. Let

$$\sigma = \begin{cases} 1 & \text{when } D \equiv 2 \text{ or } 3 \pmod 4 \\ 2 & \text{when } D \equiv 1 \pmod 4 \end{cases}$$

The discriminant of $\mathcal{K}$ is given by $\Delta = (2/\sigma)^2 D$. Let $\bar{\alpha}$ denote the conjugate of α in $\mathcal{K}$ and let $N(\cdot)$ and $Tr(\cdot)$ denote the norm and trace respectively. The integers of $\mathcal{K}$, $\mathcal{O}_{\mathcal{K}}$, are the elements for which both the trace and norm are rational integers, $\mathbb{Z}$. Then it is known that $\mathcal{O}_{\mathcal{K}}$ is generated, as a $\mathbb{Z}$-module, by 1 and $(\sigma - 1 + \sqrt{D})/\sigma$, i.e.,

$$\mathcal{O}_{\mathcal{K}} = [1, \omega], \quad \text{where} \quad \omega = (\sigma - 1 + \sqrt{D})/\sigma.$$

Any ideal of I of $\mathcal{O}_{\mathcal{K}}$ can be expressed as $I = [a, b + c\omega]$, the $\mathbb{Z}$-module generated by 1 and $b + c\omega$, and if $c = 1$ the ideal is said to be primitive. The norm of I is ac. While elements of $\mathcal{O}_{\mathcal{K}}$ do not factor uniquely, it is true that any ideal of $\mathcal{O}_{\mathcal{K}}$ can be expressed uniquely, up to order, as a product of distinct prime ideal powers. The notion of a product of ideals is well defined and we say that two ideals $I,\ J$ of $\mathcal{O}_{\mathcal{K}}$ are equivalent ($I \sim J$), if there exist two principal ideals (α) and (β) such that $(\alpha)I = (\beta)J$. This equivalence between ideals of $\mathcal{O}_{\mathcal{K}}$ divides the ideals into equivalence classes, $\mathbf{C}$ and a fundamental result of algebraic number theory is that there are only a finite number h of classes of $\mathcal{O}_{\mathcal{K}}$. There is a natural multiplication on the set of equivalence classes:

$$\mathbf{C}_i \mathbf{C}_j = \{JH | J \in \mathbf{C}_i,\ H \in \mathbf{C}_j\}$$

and under this multiplication, the set of ideal equivalence classes forms a group, the *ideal class group* of $\mathcal{O}_{\mathcal{K}}$, with identity the class containing the principal ideals.

A problem arises in determining representatives for the classes and the notion of reduced ideals is of use. An ideal I is called *reduced* if it is primitive and there is no element α of I for which $|\alpha| < N(I)$ and $|\bar{\alpha}| < N(I)$.

The discrete logarithm in the ideal class group then is: for some $D < 0$ (imaginary case), a given ideal $I \in \mathcal{O}_{\mathcal{K}}$ and $J \sim I^x, x \in \mathbb{Z}$, determine x. The complexity of the DL problem in this setting is examined in a many papers and (see for example [47]) if the parameters are chosen with care, the currently most efficient algorithms available run in exponential time in the order of the class group.

Four candidates for cyclic groups which one might consider to be suitable for the DL, DH and DDH problems as well as for a public key protocol such as key exchange, have been noted. It has been observed, in general, that DDH is no harder than DH and DH is no harder than DL and one purpose of this work is to contribute toward the understanding on the exact nature of these relationships. However for certain groups more can be said. Indeed it has recently been shown that there is a group where DDH is easy but DH is equivalent to DL [24] and this case is briefly noted here.

Consider a pair of n torsion points $P, Q \in E[n]$ on an elliptic curve over $\mathbb{F}_p$, and denote by e_n the Weil pairing, which is bilinear and non-degenerate:

$$\begin{array}{rcl} e_n : E[n] \times E[n] & \longrightarrow & \mu_n \\ (aP, bQ) & \mapsto & \langle aP, bQ \rangle = \langle P, Q \rangle^{ab} \end{array}$$

where μ_n is the set on n-th roots of unity in an extension field, $\mathbb{F}_{p^k}$. For most curves the degree of this extension is too large to make the use of such pairings practical. However when the degree is small then one can map the DL problem on the elliptic curve to one in $\mathbb{F}^*_{p^k}$ where a subexponential algorithm (index calculus) is available. This was the basis of the mapping used in [37] which showed the insecurity of supersingular curves (i.e. where p divides the trace of the curve, t) and, more generally established the so-called MOV condition where the group order on the elliptic curve divides $p^k - 1$ for some relatively small k. For supersingular curves it is possible to find two such linearly independent points, $P,\ Q,\ P \neq cQ$ since the endomorphism ring is nontrivial and in this case an endomorphism which is not a fixed point multiple is available. One can then use the points P and $\phi(P)$ for the two points, where ϕ is the endomorphism. Hence on curves where a non-degenerate pairing exists, the DDH problem becomes easy. In fact Joux and Nguyen [24] use these ideas to construct curves where the DDH problem is easy and the DH and DL problems are provably equivalent. The construction of Menezes et. al. was generalized by Frey and Rück in [16]. Their idea was to use a variant of the Tate pairing, which in addition has the property that $\langle P, P\rangle \neq 1$ for points P of "large" order. Garefalakis subsequently showed [20] that the Tate pairing can be obtained as a special case of a generalized form of the Weil pairing, thus connecting the two previous approaches.

The separation of the DDH problem from the DH and DL problem in these special groups is an interesting aspect of the problem, showing that in certain cases the three problems behave very differently. Rubin and Silverberg [44] extend these notions to the case of supersingular Abelian varieties, showing that all the notions of bilinear forms carry through to this more general setting.

3. Algorithms for generic groups

In this section, we discuss generic algorithms for the DL problem, i.e., algorithms that do make any use of the particular presentation of the group elements, and therefore work for any group. We note that any such algorithm can only make use of the group operation, inversion, and equality test for group elements. Such generic algorithms are important for two reasons: as they work for any group, they give an upper bound for the DL problem regardless of the precise presentation of the group; furthermore, there are groups of cryptographic interest, for which essentially no better method is known (i.e., it is not known how to make essential use of the group presentation). Such groups are for example Jacobian of algebraic curves over finite fields – provided that the MOV and Weil descent attacks do not work.

In what follows, we assume that the group order is a prime p, since we can always reduce the computation of a DL to this case, by first reducing to the prime power case (via the Chinese Remainder Theorem) and then to the prime case (via a p-adic expansion).

The first such method is Shank's baby-step-giant-step (BSGS) algorithm, which for any cyclic group of prime order p has time complexity $O(\sqrt{p})$, and requires $O(\sqrt{p})$ storage. Given $y = g^x$, the basic idea is to write the discrete logarithm x that we want to compute as $x = i\lceil p\rceil + j$ with $0 \leq i, j < \lceil p\rceil$. Then

$$y = g^{i\lceil p\rceil + j} \iff g^{-j}y = g^{i\lceil p\rceil},$$

and we proceed to compute and store the values

$$g^{-j}y, \quad \text{for} \quad j = 0, \dots \lceil p\rceil - 1.$$

Then we compute the values

$$g^{i\lceil p\rceil}, \quad \text{for} \quad i = 0, \dots \lceil p\rceil - 1$$

one by one, and compare with the values stored in the table. Once a collision is found the discrete logarithm can be computed.

A second method is due to Pollard [43] , and achieves the same effect, nondeterministically, without the storage requirement. Here we give the method as described by Teske in [51]. We wish to compute the DL of $y = g^x$, $|\langle g\rangle| = p$, prime. We first divide the elements of $\langle g \rangle$ into three sets T_i, $i = 1, 2, 3$ of roughly equal size, and we define the function

$$f(h) = \begin{cases} gh, & h \in T_1 \\ h^2, & h \in T_2 \\ yh, & h \in T_3 \end{cases} \tag{1}$$

We choose a number $\alpha \in_R \{1, \ldots p\}$ and set

$$h_0 = g^{\alpha}, \quad h_{i+1} = f(h_i), \quad i = 1, 2, 3, \ldots.$$

Observing the definition of f we see that $h_i = g^{\alpha_i} y^{\beta_i}$, and the two sequences $\{\alpha_i\}$ and $\{\beta_i\}$ are given by

$$\alpha_0 = \alpha, \quad \alpha_{i+1} = \begin{cases} \alpha_i + 1 & \text{if } \alpha_i \in T_1 \\ 2\alpha_i & \text{if } \alpha_i \in T_2 \\ \alpha_i & \text{if } \alpha_i \in T_3 \end{cases},$$

and

$$\beta_0 = 0, \quad \beta_{i+1} = \begin{cases} \beta_i & \text{if } \beta_i \in T_1 \\ 2\beta_i & \text{if } \beta_i \in T_2 \\ \beta_i + 1 & \text{if } \beta_i \in T_3 \end{cases}.$$

Since we operate in a finite group, the sequence $\{h_i\}$ is ultimately periodic. We proceed computing (h_i, α_i, β_i), and we store only a finite number of elements. In Teske's version, it suffices to maintain a list of 8 triplets at a time. At every step we check if a newly computed triplet matches one stored triplet. Under the assumption that f behaves as a random map, Teske shows that a match will occur after $O(\sqrt{p})$ steps. In such an event we have, say, $h_i = h_j$, and therefore

$$g^{\alpha_i - \alpha_j} = y^{\beta_j - \beta_i},$$

which implies that

$$\alpha_i - \alpha_j \equiv (\beta_j - \beta_i)x \pmod{p},$$

and the DL is computed. We refer to [51] for an excellent survey of this and other variants of Pollard's rho method.

It is natural to ask if one can do better in a generic setting like this. This question was considered by Shoup in [48], improving upon previous results of Nechaev [39]. We briefly describe Shoup's lower bound for a cyclic group of order p^r, for a prime p. In Shoup's model, a generic algorithm starts with 1 and g^x and at all times maintains a list of group elements g^{s_i}. New elements are computed and added to the list by applying the two available operations (i.e., group operation and inversion) to the elements already in the list. Shoup sets up the model, so that the algorithm can compute the DL with non-negligible probability only by looking at elements in the list. And since no other test is available (the algorithm does not look at the presentation of the elements), the DL can be computed only by finding a collision. It remains to note that starting with $\{1, g^x\}$ and applying the available operations, we get elements of the form

$$g^{s_i}, \quad s_i = F_i(x)$$

for some *linear* polynomial $F_i(X) \in \mathbb{Z}/p^r[X]$. Then a collision $g^{s_i} = g^{s_j}$ means $F_i(x) = F_j(x)$. The probability that the last equality holds for a value $x \in_R \mathbb{Z}/p^r$ is at most $1/p$. For a set of m polynomials, i.e., after m group operations, the probability for such a collision is $O(m^2/p)$. As the probability is non-negligible only for $m = \Omega(\sqrt{p})$, any successful algorithm for the DL must perform at least that many group operations. Essentially, the model reduces the DL problem to a birthday paradox, where each person corresponds to a computed group element, and a collision happens if two people have the same birthday.

In this generic setting, Shoup has also considered the complexity of the DH and DDH problem, and has shown, using arguments similar to the above, that any successful algorithm for DH has to perform at least $\Omega(\sqrt{p})$ group operations. The situation for DDH is slightly different, as one can always make use of subgroups of small order to distinguish between g^{ab} and g^c. Thus in this case, Shoup showed that a generic algorithm for DDH in G has to perform at least $\Omega(\sqrt{p})$ group operations, where now p is the *smallest* prime factor of $|G|$.

We turn now to generic reductions between the DL and DH problems. The results are due to Maurer [29], and Maurer and Wolf [30]. We are given g^x and an oracle for the DH problem in G. Our task is to compute x. Because of the Chinese Remainder Theorem we may assume that g has prime power order. In fact, we are going to need something more: the order of g needs to be prime. Thus, x is defined modulo p, and can be considered an element of $\mathbb{F}_p$. Suppose now that we can find an elliptic curve $E/\mathbb{F}_p$ such that $E(\mathbb{F}_p)$ is cyclic, and $|E(\mathbb{F}_p)|$ is smooth. Let

$$Y^2 = X^3 + AX + B, \quad A, B \in \mathbb{F}_q$$

be a model for the curve. Then, using group operations and the DH oracle, we can compute g^{x^3+Ax+B}, and then combining a square root algorithm due to Peralta [42] and the DH oracle, we can compute g^y such that $y^2 = x^3 + Ax + B$. If $x^3 + Ax + B$ happens to be a quadratic non-residue mod p, then we repeat with g^x replaced by g^{x+d} for d chosen at random in $\mathbb{F}_p$. Thus we can assume that $Q = (x, y)$ is a point on the curve. We have reduced our problem to computing Q. Fix now a generator P of $E(\mathbb{F}_p)$. One possibility for finding Q would be to compute some k such that $Q = kP$. Going through the list jP for $j = 1, 2, \ldots, |E(\mathbb{F}_p)|$, and for each point (u, v) in the sequence check if $g^u = g^x$ is the straightforward (and exponentially long) method. The idea of Maurer and Wolf is to use the smoothness of $|E(\mathbb{F}_p)|$, and compute k modulo all the prime powers dividing the group order, and finally combine the results using the Chinese Remainder Theorem. Suppose, for simplicity, that q appears to the power 1 in the factorization of $|E(\mathbb{F}_p)|$. Then, from (g^x, g^y) we can compute (g^u, g^v) such that $(u, v) = Q' = (|E(\mathbb{F}_p)|/q) \cdot Q$. Now, going through the sequence $(j|E(\mathbb{F}_p)|/q) \cdot P$ for $j = 1, \ldots, q$ is feasible, since the prime q is assumed to be small. This will give us some k_q such that $k \equiv k_q \pmod{q}$.

On the negative side, Maurer and Wolf have shown in [32] that the condition, that all large prime factors of G appear with multiplicity 1, in the above generic reduction is essential. Using the model defined by Shoup, and similar techniques, they showed that no generic reduction from the DL to the DDH problem can exist if the order of the group is divisible by the square of a large prime. Thus, under the assumption that a suitable elliptic curve can be found, the DL problem and the DH problem for a group G are equivalent if and only if $|G|$ is not divisible by the square of any large prime. Similar results have also been shown by the same authors for the relation between the DH and DDH problems. Namely, the DDH problem and the DH problem for a group G are equivalent (under generic reductions) if and only if all prime factors of $|G|$ are small (in which case both problems are easy).

With respect to generic reductions, the picture is now quite complete. We note, that generic algorithms work regardless of the presentation of the group and therefore the upper bounds are valid for *every* specific group. Negative results, however, do not imply *anything* for any particular group. Thus, studying the relation between the three problems in particular groups is still a very interesting open problem.

The results of Nechaev and Shoup are in sharp contrast with the situation in multiplicative groups of finite fields and in Jacobians of hyperelliptic curves of large genus, where index calculus methods are known, and have subexponential time complexity. The reason is clearly that in those groups one can make essential use of the encoding of group elements: for instance, one is able to define unique decomposition of group elements over a specified set, which is important for index calculus. This "unique decomposition" property is already an assumption about the presentation of the group. In all known cases, where the index calculus beats exponential time complexity, one also has strong results about the probability of a group element decomposing over the specified set – referred to as *smoothness probability*. In a recent work, Enge and Gaudry [15] gave an abstraction of the index calculus method to groups where one has a notion of unique decomposition. They analyzed the algorithm in this general setting, and showed that certain assumptions about the density of smooth elements are in essence all one needs to obtain provably subexponential complexity. As the authors are mainly interested in providing an abstraction of known *provable* methods, their smoothness assumption leads to time complexity of the form $L(1/2, c)$. Stronger assumptions about the smoothness probability would lead to faster algorithms. However, in this case there would be no known example (specific group) which would provably satisfy the assumptions. Of course, unique decomposition and smoothness need to be effective in an algorithmic sense, so efficient smoothness tests and decomposition algorithms have also to be available.

4. Distribution problems for Diffie-Hellman triples

Another way of looking at the DDH problem in $(\mathbb{Z}/p)^*$ is as follows. Since DDH is easily seen to be random self-reducible, deciding the DDH problem in the random case is as hard as in the worst case. What we want then is, given p, g, g^a, g^b and g^c, where $a, b \in_R (\mathbb{Z}/p)^*$, to decide if $c \equiv ab \pmod p$ or c is chosen uniformly at random from $(\mathbb{Z}/p)^*$. Equivalently, we want to distinguish between the distribution (g^a, g^b, g^{ab}) – also called Diffie-Hellman distribution – and the uniform distribution. The assumption that, for suitably chosen g, this is intractable is the so-called Diffie-Hellman indistinguishability assumption (DHI).

Note that to distinguish between the two distributions, an algorithm for DDH suffices. Thus, regardless of the characteristics of the Diffie-Hellman distribution, an algebraic algorithm for deciding DDH would make the DHI assumption false. However, in the absence of any such algorithm, studying the Diffie-Hellman distribution may give us some reassurance about the validity of the DHI assumption.

The Diffie-Hellman distribution has been studied by Canetti et.al. in [11, 12]. The authors formalize the assumption as follows.

Assumption 4.1 ([12]). *Fix $e > 0$ and let p be an n-bit prime. Let $g \in (\mathbb{Z}/p)^*$ have order t such that for every prime divisor ℓ of t, $\ell > p^e$. If a, b, c are chosen uniformly at random from $(\mathbb{Z}/t)^*$, then the distributions (g^a, g^b, g^{ab}) and (g^a, g^b, g^c) are computationally indistinguishable.*

The reason for the assumption that g is a residue for all small prime divisors of $p - 1$ is to avoid trivial cases, where the two distributions can be distinguished using the power residuocity of g.

The main result of [12] is essentially that if one considers a linear (but sufficiently small) proportion of the most significant bits (or the least significant bits) of the elements involved, then the two distributions are statistically very close. More formally, if $\sigma_k(\cdot)$ denotes the k-most (or least) significant bits of a string, then the statistical distance between the uniform distribution on $\{0,1\}^{3k}$ and $(\sigma_k(g^a), \sigma_k(g^b), \sigma_k(g^{ab}))$ is exponentially small. The statistical distance between two distributions α and β over the same domain D is defined to be

$$\mathrm{var}(\alpha, \beta) = \sum_{x \in D} |\alpha(x) - \beta(x)|.$$

Theorem 4.2 ([12]). *Let p be an n-bit prime, and let $g \in (\mathbb{Z}/p)^*$ be of order $t > p^{3/4+\varepsilon}$ for some $\varepsilon > 0$. Then there exists some positive constant γ such that for any $k \le \gamma n$, the statistical distance between $(\sigma_k(g^a), \sigma_k(g^b), \sigma_k(g^{ab}))$ and the uniform distribution on $\{0,1\}^{3k}$ is exponentially small.*

In fact, Theorem 2 of [12] contains also a slightly weaker result for the more general case that one looks at any k bits (not necessarily consecutive).

Theorem 4.2 provides evidence that any algorithm based on statistical data alone, cannot be successful in distinguishing the Diffie-Hellman distribution from the uniform.

5. The bit complexity of the DH problem in $\mathbb{F}_p$

The question of the time complexity of the DH and DDH problems remains unanswered. Assume, however, that the two problems are hard. Is this enough to ensure that the Diffie-Hellman protocol is secure? The answer to that depends on how one uses the established key. For instance, it is conceivable that only some, say 64, of the most significant bits are used as a secret key for a block cipher. In this case, an eavesdropper who cannot compute the Diffie-Hellman function, but can compute the 64 most significant bits can still crack the session. Thus, it becomes important to study the security of small substrings (or individual bits if possible) of the key.

The first breakthrough in this direction came by Boneh and Venkatesan in [4]. Their idea was to show that any algorithm, which can compute about $O(\sqrt{n})$ most significant bits of the DH function can be converted into an algorithm to compute all the bits.

We will need a notion for the m most significant bits of elements of $\mathbb{F}_p^*$. It is common in this context (see [4, 22]) to define the m most significant bits of $t \in \{0, \ldots, p-1\}$ as $f_m(t)$, where

$$f_m(t)\ \frac{p}{2^m} \le t < (f_m(t)+1)\ \frac{p}{2^m}. \tag{2}$$

An explanation here is in order. In general, the m most significant bits of an n-bit string t are defined as the number B such that

$$B\ 2^{n-m} \le t < (B+1)\ 2^{n-m}.$$

This would be in agreement with our definition if p actually were 2^n. The problem arises when p is greater, but very close to a power of 2. As an extreme example consider a prime of the form $p = 2^n + 1$. Then p is of $n+1$ bits, but almost all elements in $\mathbb{F}_p$ are represented by strings whose first bit is 0. Thus, the most significant bit in this case can easily be predicted (and carries almost no information). Note that in the opposite extreme, when say $p = 2^n - 1$, then the two definitions are (for all that matters) equivalent. Finally, we note that the two definitions differ by one bit only. This is the reason, why in [22] choosing between the two definitions is not important: the number of bits used there is $\sim \sqrt{n}$, so the difference becomes insignificant.

Suppose now, that we have an oracle $\mathcal{O}_m$, which on input $g, g^a, g^b \in \mathbb{F}_p$ outputs the m most significant bits of g^{ab}, i.e., $\mathcal{O}_m(g, g^a, g^b) = f_m(g^{ab})$. Is there an efficient algorithm, which makes a polynomial number of queries to the oracle $\mathcal{O}_m$, that can compute the DH function? For $m = n$ the problem is trivial. The goal is to devise an algorithm for m as small as possible. The above problem was first formalized in [4].

Hidden Number Problem (HNP): Fix a prime p and a positive integer m. Fix an element

$\alpha \in \mathbb{F}_p^*$. Compute α in expected polynomial time given access to an oracle $\mathcal{O}_{\alpha,m}$, which on input $t \in \mathbb{F}_p^*$ outputs

$$\mathcal{O}_{\alpha,m}(t) = f_m(\alpha t).$$

The oracle just defined is nothing more than an oracle for the m most significant bits of a DH key. Indeed, consider a Diffie-Hellman triple (g^a, g^b, g^{ab}). The hidden number is $\alpha = g^{ab}$. If we choose $x \in [1, p-1]$, then

$$g^{(a+x)b} = g^{ab+xb} = \alpha g^{bx},$$

therefore

$$\mathcal{O}_m(g, g^{a+x}, g^b) = \mathcal{O}_{\alpha,m}(g^{bx}).$$

Boneh and Venkatesan proposed, in [4], a method for solving the HNP for multipliers t chosen uniformly at random in $\mathbb{F}_p^*$, and for $m = \lceil\sqrt{n}\rceil + \lceil\log n\rceil$ using roughly $2\sqrt{n}$ oracle calls, where n is the number of bits of the prime p. Their method is based on lattice basis reduction techniques. The argument work as follows. After d oracle queries for random inputs $t_1, \dots, t_d \in \mathbb{F}_p^*$, we know d numbers $u_1, \dots, u_d$ such that

$$u_i \frac{p}{2^m} \le \alpha t_i \text{ rem } p < (u_i + 1)\frac{p}{2^m}. \tag{3}$$

We consider now the lattice L generated by the rows of the matrix

$$\begin{pmatrix} p & 0 & \cdots & 0 & 0 \\ 0 & p & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \\ 0 & 0 & \cdots & p & 0 \\ t_1 & t_2 & \cdots & t_d & 1/p \end{pmatrix}.$$

We refer to the first d vectors as p-vectors. Then by multiplying the last vector by α and subtracting appropriate multiples of p-vectors, we see that the vector

$$\mathbf{v}_\alpha = (\alpha t_1 \text{ rem } p, \alpha t_2 \text{ rem } p, \dots, \alpha t_d \text{ rem } p, \frac{\alpha}{p})$$

is a lattice vector. Moreover, this vector is very close to the known vector

$$\mathbf{u} = (u_1 \frac{p}{2^m}, u_2 \frac{p}{2^m}, \dots, u_d \frac{p}{2^m}, 0).$$

Indeed, by Eq.(3) we have

$$0 \le \alpha t_i - u_i \frac{p}{2^m} < \frac{p}{2^m}, \quad i = 1, \dots, d$$

and therefore the Euclidean distance between the two vectors is bounded by

$$\left(d \frac{p^2}{2^{2m}} + \frac{\alpha^2}{p^2}\right)^{1/2} \le \frac{p}{2^m}\sqrt{d} + \frac{\alpha}{p}.$$

For carefully chosen d and m, $\mathbf{v}_\alpha$ is so close to the known vector $\mathbf{u}$ that one does not expect any other lattice vectors to be so close. If this is the case, then we can recover $\mathbf{v}_\alpha$ using an algorithm for the closest lattice vector problem (e.g., [1, 25]).

In choosing d and m we have two goals to meet: the two vectors have to be close enough so that $\mathbf{v}_\alpha$ is unique *and* the closest lattice vector algorithm is guaranteed to find it. Boneh and Venkatesan prove in [4] that for $d, m = O(\sqrt{\log p})$ the vector $\mathbf{v}_\alpha$ is indeed unique provided that the multipliers t_i are chosen uniformly at random in $\mathbb{F}_p^*$.

Going back to the bit security problem of the Diffie-Hellman key, we see that the multipliers are elements of the form g^{bx} for $x \in_R [1, p-1]$, and therefore cannot be uniformly distributed unless $\gcd(b, p-1) = 1$. This problem was realized by González Vasco and Shparlinski in [22]. If g is a generator of a proper subgroup, or b is chosen so that $\gcd(b, p-1) > 1$ then the uniqueness proof (Theorem 5 in [4]) is not general enough to apply. The idea of González Vasco and Shparlinski is to "randomize" the exponent b first, by computing $g_r = g^{b+r}$ for $r \in_R [0, T-1]$ where T is the order of the element g. Then one works as before, computing g^{a+x}, and using an oracle for the most significant bits of $g^{(a+x)(b+r)}$, for randomly chosen values of x. We see that

$$g^{(a+x)(b+r)} = g_r^{a+x} = g_r^a g_r^x.$$

So, again, we have a HNP where now the hidden number, g_r^a, lives in the subgroup $\langle g_r \rangle$ and the multipliers g_r^x are uniformly distributed in the same subgroup. We note that recovering g_r^a is equivalent to recovering g^{ab}.

It remains to show that the lattice based algorithm for the HNP works for multipliers that lie in a proper subgroup. The authors show that for any element $g \in \mathbb{F}_p$ of order $T \geq p^{1/3+\varepsilon}$ for any $\epsilon > 0$ and multipliers $t_i \in_R \langle g \rangle$, any vector sufficiently close to the vector $\mathbf{u}$ is, with high probability, of the form

$$\mathbf{v} = (\beta t_1 \text{ rem } p, \ldots, \beta t_d \text{ rem } p, \frac{\beta}{p}),$$

for some $\beta \equiv \alpha \pmod p$. This, together with the fact that the order of g_r above is greater than $p^{1/3+\varepsilon}$ with very high probability, provide all the necessary components of the proof.

The uniqueness proof is based on a number theoretic result concerning the distribution of g^x modulo p, when g is of order $T \geq p^{1/3+\varepsilon}$. The main tool for such "uniformity" results are exponential sums. We state Lemma 2.1 of [22] here for later reference.

Lemma 5.1 (Lemma 2.1 in [22]). *For any $\varepsilon > 0$ there exists $\delta > 0$ such that for any element $g \in \mathbb{F}_p^*$ of order $T \geq p^{1/3+\varepsilon}$ we have*

$$\max_{r,h} \max_{\gcd(\lambda,p)=1} \left| N_{\lambda,p}(r,h) - \frac{Th}{p} \right| = O(T^{1-\delta}),$$

where $N_{\lambda,p}(r,h)$ is the number of integers $x \in [0, T-1]$ such that λg^x rem $p \in [r+1, r+h]$.

Later, we will require an estimate for the number $N_{\lambda,p}(r_1, r_2, h)$ of $x \in \{0, \ldots, T-1\}$ such that $\lambda g^x \in [r_1, r_1+h) \cup (r_2, r_2+h]$. Such an estimate follows immediately from Lemma 5.1, since

$$N_{\lambda,p}(r_1, r_2, h) = N_{\lambda,p}(r_1, h) + N_{\lambda,p}(r_2, h),$$

and therefore,

$$\begin{aligned}
& \max_{r,h} \max_{\gcd(\lambda,p)=1} \left| N_{\lambda,p}(r_1, r_2, h) - \frac{2Th}{p} \right| \\
= \; & \max_{r,h} \max_{\gcd(\lambda,p)=1} \left| N_{\lambda,p}(r_1, h) + N_{\lambda,p}(r_1, h) - \frac{2Th}{p} \right| \\
\leq \; & \max_{r,h} \max_{\gcd(\lambda,p)=1} \left| N_{\lambda,p}(r_1, h) - \frac{Th}{p} \right| \\
& + \max_{r,h} \max_{\gcd(\lambda,p)=1} \left| N_{\lambda,p}(r_1, h) - \frac{Th}{p} \right| \\
= \; & O(T^{1-\delta}).
\end{aligned}$$

We note that the algorithm of Boneh and Venkatesan in practice preforms better than predicted in theory. Namely, the number of bits needed (for each oracle call) is only a small constant, rather than $\sim \sqrt{\log p}$. The reason for this disagreement is that the LLL algorithm preforms much better in practice than theoretically known. Thus, any advance in lattice basis reduction would have direct implications to the problem of the bit-security of the DH key.

Our goal here is to reduce the number of needed bits, at the expense of an assumption stronger than the Diffie-Hellman assumption. Our basis is the assumption that the decision Diffie-Hellman problem is hard.

Theorem 5.2. *Let p be a prime, and $g \in \mathbb{F}_p$ be an element of multiplicative order $T \geq p^{1/3+\epsilon}$. There exits a probabilistic polynomial time algorithm, which for any triplet $(a, b, c) \in [1, T]^3$, given $\lfloor g^a \rfloor_p, \lfloor g^b \rfloor_p, \lfloor g^c \rfloor_p$, makes k calls to the oracle $\mathcal{O}$, and decides if $\lfloor g^{ab} \rfloor_p = \lfloor g^c \rfloor_p$ with error probability exponentially close to 2^{-k}.*

Proof. The basic idea for the proof is to use a fingerprinting technique to decide the equality $\lfloor g^{ab} \rfloor_p \overset{?}{=} \lfloor g^c \rfloor_p$. Such techniques have been very useful in deciding algebraic equalities such as equalities of matrices very efficiently (see [38]). In our case, one part of the equality, namely $\lfloor g^{ab} \rfloor_p$, is not known. Instead, we will use the oracle $\mathcal{O}$, which gives us some information about $\lfloor g^{ab} \rfloor_p$. To demonstrate the basic idea we first consider the simple situation, where g is a primitive root modulo p, and $\gcd(b, p-1) = 1$. In this case g^b is again a primitive root modulo p.

Denote $\alpha = \lfloor g^{ab} \rfloor_p$, and $\gamma = \lfloor g^c \rfloor_p$. Thus we wish to decide if $\alpha = \gamma$. We repeat the following step k times. Select an integer $x \in \{1, \dots, p-1\}$ uniformly at random. Then

$$\begin{aligned} g^{ab} &\equiv g^c \pmod p \iff \\ g^{ab} g^{bx} &\equiv g^c g^{bx} \pmod p \iff \\ \alpha t &\equiv \gamma t \pmod p, \end{aligned}$$

where $t = \lfloor g^x \rfloor_p$ is uniformly distributed in $\mathbb{F}_p^*$. If $\alpha = \gamma$ then $\lfloor \alpha t \rfloor_p = \lfloor \gamma t \rfloor_p$ for every $t \in \mathbb{F}_p^*$. If, on the other hand, $\alpha \neq \gamma$, then $\lfloor \alpha t \rfloor_p \neq \lfloor \gamma t \rfloor_p$ for every $t \in \mathbb{F}_p^*$. One would expect this inequality to be reflected in the two most significant bits of the two values for *many* t's. Indeed, we show that this is the case, and in fact the two most significant bits are different for at least half of the values of t.

We consider the number of t's such that $f_2(\lfloor \alpha t \rfloor_p) = f_2(\lfloor \gamma t \rfloor_p) = B$, in the case $\alpha \neq \gamma$. Let

$$\alpha t \equiv y_1 \pmod p, \quad \text{and} \quad \gamma t \equiv y_2 \pmod p.$$

Then

$$(\alpha - \gamma) t \equiv y_1 - y_2 \pmod p \tag{4}$$

and

$$\begin{aligned} B\frac{p}{4} &\leq y_1 < (B+1)\,\frac{p}{4} \\ B\frac{p}{4} &\leq y_2 < (B+1)\,\frac{p}{4}. \end{aligned}$$

Therefore

$$-\frac{p}{4} < y_1 - y_2 < \frac{p}{4}.$$

This together with Equation (4), gives

$$(\alpha - \gamma) t \equiv y \pmod p, \quad y \in \left[1, \frac{p-1}{4}\right) \cup \left(p - \frac{p-1}{4}, p-1\right]. \tag{5}$$

Since $\alpha \neq \gamma$, the number of solutions is upper bounded by $(p-1)/2$. Therefore, for at least $(p-1)/2$ values of t, $f_2(\lfloor \alpha t \rfloor_p) \neq f_2(\lfloor \gamma t \rfloor_p)$. It remains to show that one can check the equality

$$f_2(\lfloor \alpha t \rfloor_p) \neq f_2(\lfloor \gamma t \rfloor_p).$$

As both γ and t can be computed from the input, so can $f_2(\lfloor \gamma t \rfloor_p)$. We observe now that $\lfloor \alpha t \rfloor_p$ can be realized as a Diffie-Hellman key

$$\alpha t \equiv g^{ab} g^{bx} \equiv g^{(a+x)b} \pmod p,$$

so the two most significant bits can be given by the oracle:

$$f_2(\lfloor \alpha t \rfloor_p) = \mathcal{O}(g, g^{a+x}, g^b).$$

Repeating the previous experiment, and checking $f_2(\lfloor \alpha t \rfloor_p) \overset{?}{=} f_2(\lfloor \gamma t \rfloor_p)$, k times, we conclude that the error probability is at most 2^{-k}.

We consider now the more general situation, as stated in the theorem. The algorithm has to be modified slightly to accommodate the fact that the multiplicative order T of g is not necessarily $p-1$ any more, and that $\gcd(b,T)$ may be greater than 1. Other than that, the proof is again based on a counting argument.

In this case, we select, once and for all, $r \in [1,T]$ and let $g_r \equiv g^{b+r} \pmod p$. The order of g_r is $T/\gcd(b+r,T)$. Now,

$$\begin{aligned} \frac{T}{\gcd(b+r,T)} &\geq p^{1/3+\epsilon/2} \quad \Longleftrightarrow \\ \gcd(b+r,T) &\leq Tp^{-1/3-\epsilon/2}. \end{aligned} \tag{6}$$

The probability that Equation (6) does not hold is at most $\tau(T)T^{-1}p^{1/3+\epsilon/3}$, where $\tau(T)$ is the number of positive divisors of T. As $\tau(T) = O(T^\delta)$ for any $\delta > 0$, we have that Equation (6) is violated with probability at most

$$O\left(p^{1/3+\epsilon/2}T^{-1+\delta}\right) = O\left(p^{\delta-\epsilon/2}\right).$$

Since this holds for any $\delta > 0$, we see that the probability, over the random choice of r, can be made $O(p^{-\varepsilon})$ for some $\varepsilon > 0$.

Thus, with probability at least $1 - O(p^{-\varepsilon})$, we now have an element g_r of multiplicative order at least $p^{1/3+\epsilon/2}$. We now proceed as in the simple case, and select $x \in \{0, \ldots, T-1\}$ uniformly at random. We have

$$\begin{aligned} g^{ab} &\equiv g^c \pmod p \quad \Longleftrightarrow \\ g^{ab} g^{ar+x(b+r)} &\equiv g^c g^{ar+x(b+r)} \pmod p \quad \Longleftrightarrow \\ g^{(a+x)(b+r)} &\equiv g^{c+ar} g^{x(b+r)} \pmod p \quad \Longleftrightarrow \\ g_r^a g_r^x &\equiv g^{c+ar} g_r^x \pmod p. \end{aligned}$$

As in the simple case, we can compute $\gamma = \lfloor g^{c+ar} \rfloor_p$, and $t = \lfloor g_r^x \rfloor_p$. Of course, the value $\alpha t = g_r^{a+x}$ is unknown, but we observe that it is a Diffie-Hellman key, since $g_r^{a+x} \equiv g^{(a+x)(b+r)} \pmod p$, and therefore the two most significant bits of $\lfloor g_r^a g_r^x \rfloor_p$ are given by a call to the oracle:

$$f_2(\lfloor \alpha t \rfloor_p) = \mathcal{O}(g, g^{a+x}, g^{b+r}).$$

It remains to show that for $\alpha \neq \gamma$ (equivalently, $\lfloor g^{ab} \rfloor_p \neq \lfloor g^c \rfloor_p$), with probability exponentially close to $1/2$, $f_2(\lfloor \alpha t \rfloor_p) \neq f_2(\lfloor \gamma t \rfloor_p)$.

Suppose $f_2(\lfloor \alpha t \rfloor_p) = f_2(\lfloor \gamma t \rfloor_p) = B$. Then with exactly the same reasoning as in the simple case, we have

$$(\alpha - \gamma)t \equiv y \pmod p, \quad y \in \left[1, \frac{p-1}{4}\right) \cup \left(p - \frac{p-1}{4}, p-1\right], \quad t \in \langle g_r \rangle. \tag{7}$$

By Lemma 5.1, and the remark following it, we have that the number of solutions to Equation (7) is

$$\frac{2T(p-1)}{4p} + O(T^{1-\delta}),$$

for some $\delta > 0$. Thus the probability, over t, that Equation (7) is satisfied (i.e., $\alpha \not\equiv \gamma$ but $f_2(\lfloor \alpha t \rfloor_p) = f_2(\lfloor \gamma t \rfloor_p)$) is $1/2 + O(T^{-\delta}) = 1/2 + O(p^{-\delta/3})$.

Repeating the experiment k times for independent choices of x yields the stated result. $\square$

6. Comments

The main result of this work is that if the DDH function is computationally infeasible then the two most significant bits of the DH function are secure. It is interesting to speculate if this result is the best possible i.e. how likely is it that a one-bit result could be obtained? In this regard note the following point [4], that it is easy to compute the Legendre symbol of g^{ab} assuming those of g^a and g^b are known. Thus knowing such information is not entirely unrelated to determining the DH function. While this observation is considerably weaker than the one-bit result sought, it does seem to indicate caution in conjecturing that the most significant bit of g^{ab} is as difficult to determine as the DH function itself.

Another interesting question is whether a deterministic reduction of DDH to the problem of computing the two most significant bits of DH exists. For example, given g^a, g^b, and g^c we can form the sequence

$$y_0 = g^c, \quad y_i \equiv y_{i-1}^2 \pmod p, \quad i \geq 1.$$

A related sequence is

$$x_0 = g^{ab}, \quad x_i \equiv x_{i-1}^2 \pmod p, \quad i \geq 1.$$

Of course, $\{x_i\}$ cannot be computed, but we have access to the 2 most significant bits of each element through the oracle, since

$$x_0 = DH(g^a, g^b), \quad \text{and} \quad x_i = DH(g^{2^i a}, g^b).$$

The two sequences become eventually the same only under very strict conditions on x_0 and y_0. Suppose, for instance, that $p - 1 = 2q$, where q is odd, and let k be the least index such that $x_k \equiv y_k \pmod p$. Then $x_{k-1}^2 \equiv y_{k-1}^2 \pmod p$. By the choice of k we have $x_{k-1} \not\equiv y_{k-1} \pmod p$ Also x_{k-1} is by construction a quadratic residue mod p, while $-y_{k-1}$ is not (note that -1 is not a quadratic residue since there are no elements of order 4 in $\mathbb{F}_p$), so $x_{k-1} \not\equiv -y_{k-1} \pmod p$. We conclude that our assumption was false – unless of course $k = 0$ in which case the two sequences are identical. Using similar arguments one can easily show that if a higher power of 2 divides $p - 1$ then either the two sequences collide very early, or never do. Is comparing the two (or perhaps only one) most significant bits of successive elements in the sequences a good strategy for deciding whether they are identical or not? We note that in the case that a higher power of 2 divides $p - 1$ and g^c is chosen so that $x_k \equiv y_k \pmod p$, for some k (take it to be least), then $x_{k-1} \equiv -y_{k-1} \pmod p$, so the elements x_{k-1} rem p and y_{k-1} rem p differ at the most significant bit. So, if the two sequences collide, they do so early, and the difference can be detected by the most significant bit only.

A resolution of the problem along these lines would be of great interest and there clearly remain interesting questions to pursue.

References

[1] M. Ajtai, S. Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *33rd Annual ACM Symposium on Theory of Computing*, 2001.

[2] Eli Biham, Dan Boneh and Omer Reingold, Breaking generalized Diffie-Hellman modulo a composite is no easier than factoring, Information Processing Letters, vol. 70, pp. 83-87, 1997.

[3] Ian F. Blake, Gadiel Seroussi and Nigel P. Smart, *Elliptic curves in cryptography*, Cambridge University Press, London Math. Soc., Lecture Note Series, vol. 265, 1999).

[4] Dan Boneh and R. Venkatesan, Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes, Crypto '96, Springer-Verlag, Lecture Notes in Computer Science vol. 1109, N. Koblitz ed., pp. 129-142, 1996.

[5] Dan Boneh and Richard J. Lipton, Algorithms for black-box fields and their application to cryptography, Crypto '96, Springer-Verlag, Lecture Notes in Computer Science vol. 1109, N. Koblitz ed., pp. 283-297, 1996.

[6] Dan Boneh and R, Venkatesan, Rounding in lattices and its cryptographic applications, Proc. SODA, pp. 675-681, 1997.

[7] Dan Boneh, The decision Diffie-Hellman problem, ANTS III, Springer-Verlag, Lecture Notes in Computer Science vol. 1423, J.P. Buhler ed., pp. 48-63, 1998.

[8] Dan Boneh and Igor E. Shparlinski, On the unpredictability of elliptic curve Diffie-Hellman scheme, Crypto '01, Springer-Verlag, Lecture Notes in Computer Science vol. 2139, J. Kilian, ed., pp. 201-212, 2001.

[9] J. Buchmann and H.C. Williams, A key-exchange system based on imaginary quadratic fields, *J. Cryptology*, vol. 1, pp. 107-118, 1988.

[10] J. Buchmann and H.C. Williams, Quadratic fields and cryptography, in *Number Theory and Cryptology*, Cambridge University Press, pp. 9-26, 1990.

[11] R. Canetti, J. Friedlander and I. Shparlinski, On certain exponential sums and the distribution of Diffie-Hellman triples, *J. London Math. Soc.*, vol. 59, pp. 799-812 1999.

[12] Ran Canetti, John Friedlander, Sergei Konyagin, Michael Larsen, Daniel Lieman and Igor E. Shparlinski, On the statistical properties of Diffie-Hellman distributions, *Israel J. Math.*, vol. 120, pp. 23-46, 2000.

[13] M.A. Cherepnev, On the connection between the discrete logarithms and the Diffie-Hellman problem, *Discrete Math. Appl.*. vol. 6, pp. 341-349 1996.

[14] W. Diffie and M.E. Hellman, New directions in cryptography, *IEEE Transaction on Information Theory*, vol. 22, pp. 644-654, 1976.

[15] A. Enge and P. Gaudry. A general framework for subexponential discrete logarithm algorithms. *Acta Arithmetica*, 202(1):83–103, 2002.

[16] G. Frey, M. Müller, and H.G. Rück. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Trans. Inform. Theory*, 45(5):1717–1719, 1999.

[17] G. Frey and H.-G. Rück, A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves, *Math. Comp*, vol. 62, pp. 865-874, 1994.

[18] M. Fouquet, P. Gaudry and R. Harley, An extension of Satoh's algorithm and its implementation, *Journal of the Ramanujan Mathematical Society*, vol. 15 pp. 281-318, 2000.

[19] John B. Friedlander and Igor E. Shparlinski, On the distribution of Diffie-Hellman triples with sparse exponents, *SIAM J. Disc. Math.*, vol. 14, pp. 162-169 2001.

[20] T. Garefalakis. The generalized Weil pairing and the discrete logarithm problem on elliptic curves. In S. Rajsbaum, editor, *5th Latin American Symposium on Theoretical Informatics – LATIN 2002*, volume 2286 of *LNCS*, pages 118–130, Cancun, Mexico, 2002.

[21] P. Gaudry, An algorithm for solving the discrete logarithm problem on hyperelliptic curves, Eurocrypto 2000, Springer-Verlag, Lecture Notes in Computer Science vol. 1807, pp. 19-34, 2000.

[22] Maria Isabel González Vasco and Igor E. Shparlinski, On the security of Diffie-Hellman bits, *Proc. Workshop on Crypto. and Comp. Number Theory*, Birkhauser, Singapore, pp. 331-342, 2001.

[23] Helena Handschuh, Yiannis Tsiounis and Moti Young, Decision oraces are equivalent to matching oracles, Tech. Report PK01-1999, Gemplus Corp., 1999.

[24] Antoine Joux and Kim Nguyen, Separating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups, `eprint.iacr.org/2001/003.ps.gz`.

[25] R. Kannan. Algorithmic geometry of numbers. *Annual Reviews in Computer Science*, vol. 2, 231–267, 1987.

[26] N. Koblitz, *Algebraic aspects of cryptography*, Springer-Verlag, Algorithms and Computation in Mathematics vol. 3, 1998.

[27] Sergei Konyagin, Tanja Lange and Igor E. Shparlinski, Linear complexity of the discrete logarithm, *Designs, Codes and Cryptography*, to appear.

[28] Edwin El Mahassni and Igor E. Shparlinski, Polynomial representations of the Diffie-Hellman mapping, *Bull. Austral. Math. Soc.*, vol. 63, pp. 467-473, 2001.

[29] Ueli M. Maurer, Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms, Crypto '94, Springer-Verlag, Lecture Notes in Computer Science vol. 839, pp. 271-281 1994.

[30] Ueli M.Maurer and Stefan Wolf, The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms, *SIAM J. Comput.*, vol. 28, pp. 1689-1721 1999.

[31] Ueli M. Maurer and Stefan Wolf, On the complexity of breaking the Diffie-Hellman protocol, Tech. Report 244, Inst. Theor. Comp. Sci., ETH, Zürich, 1996.

[32] Ueli Maurer and Stefan Wolf, Lower bounds on generic algorithms in groups, Eurocrypt '98, Springer-Verlag, Lecture Notes in Computer Science vol. 1403, K. Nyberg ed., pp. 72-84, 1998.

[33] Ueli M. Maurer and Stefan Wolf, Diffie-Hellman oracles, Crypto '96, Springer-Verlag, Lecture Notes in Computer Science vol. 1109, N. Koblitz ed., pp. 268-282, 1996.

[34] Ueli M. Maurer and Stefan Wolf, The Diffie-Hellman protocol, *Designs, Codes and Cryptography*, vol. 19, pp. 147-171, 2000.

[35] Ueli M. Maurer and Stefan Wolf, Diffie-Hellman, Decision Diffie-Hellman and Discrete logarithms, *International Symposium on Information Theory*, Ulm, June, 1998.

[36] Alfred J. Menezes, *Elliptic curve public key cryptosystems*, Kluwer Academic Publishers, 1993.

[37] Alfred J. Menezes, T. Okamoto and S. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Trans. Inform. Theory*, vol. 39, pp. 1639-1646, 1993.

[38] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.

[39] V.I. Nechaev, Complexity of a deterministic algorithm for the discrete logarithm, *Mathematical Notes*, vol. 55, no. 2, pp. 165-172, 1994.

[40] Tatsuki Okamoto and David Pointcheval, The gap problems: A new class of problems for the security of cryptographic schemes, PKC 2001, Springer-Verlag, Lecture Notes in Computer Science vol. 1992, K. Kim ed., pp. 104-118, 2001.

[41] S. Paulus and H.-G. Rück, Real and imaginary quadratic representations of hyperelliptic function fields, *Math. Comp.*, vol. 68, pp. 1233-1242, 1999.

[42] R. Peralta. A simple and fast probabilistic algorithm for computing square roots modulo a prime number. *IEEE Transactions on Information Theory*, 32, 1986.

[43] J. Pollard. Monte carlo methods for index computations (mod p). *Math. Comp.*, 32:918–924, 1978.

[44] Karl Rubin and Alice Silverberg, Supersingular abelian varieties in cryptology, preprint, 2002.

[45] R. Schoof, Counting points on elliptic curves over finite fields, *Journal Theorie des Nombres de Bordeaux*, vol. 7, pp. 219-254, 1995.

[46] T. Satoh, The canonical lift of an ordinary elliptic curve over a finite field and its point counting, *Journal of the Ramanujan Mathematical Society*, vol. 15, pp.247-270, 2000.

[47] R. Scheidler, Cryptography in quadratic number fields, *Designs, Codes and Cryptography*, vol. 22, pp. 239-264, 2001.

[48] Victor Shoup, Lower bounds for discrete logarithms and related problems, Eurocrypt '97, Springer-Verlag, Lecture Notes in Computer Science vol. 1233, W. Fumy ed., pp. 256-266, 1997.

[49] Igor E. Shparlinski, On the distribution of Diffie-Hellman pairs, *Finite Fields and their Applns.*, vol. 8, pp. 131-141, 2002.

[50] Igor E. Shparlinski, Security of most significant bits of g^{x^2}, *Info. Proc. Letters*, vol. 8, pp. 131-141, 2002.

[51] Edlyn Teske, Square-root algorithms for the discrete logarithm problem (A Survey), In: *Public-Key Cryptography and Computational Number Theory*, Walter de Gruyter, pp. 283-301, 2001.

Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada

E-mail address: `ifblake@comm.toronto.edu`

Department of Mathematics University of Toronto, Toronto, ON M5S 3G4, Canada
E-mail address: `theo@comm.utoronto.ca`