Diffie-Hellman Oracles

Ueli M. Maurer and Stefan Wolf

Institute for Theoretical Computer Science
ETH Zürich
CH-8092 Zürich, Switzerland
E-mail addresses: {maurer,wolf}@inf.ethz.ch

Abstract. This paper consists of three parts. First, various types of Diffie-Hellman oracles for a cyclic group G and subgroups of G are defined and their equivalence is proved. In particular, the security of using a subgroup of G instead of G in the Diffie-Hellman protocol is investigated. Second, we derive several new conditions for the polynomial-time equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms in G which extend former results by den Boer and Maurer. Finally, efficient constructions of Diffie-Hellman groups with provable equivalence are described.

Keywords. Public-key cryptography, Diffie-Hellman protocol, Discrete logarithms, Elliptic curves.

1 Introduction

Let G be a cyclic group with generator g. The Diffie-Hellman (DH) problem [6] is, for given g^u and g^v, to compute g^{uv}. A possible group for the DH protocol [6] is $\mathbf{Z}_p^*$, where p is a prime number, or an elliptic curve over a finite field [17],[9].

The DH problem is at most as difficult as computing discrete logarithms (DL) in G with respect to the base g. By analyzing DH-oracles and their application for computing discrete logarithms we take a number of steps towards proving that the two problems are computationally equivalent. At CRYPTO '94, Maurer showed that a sufficient condition for such an equivalence is that for all large prime factors p of $|G|$, a cyclic elliptic curve over $GF(p)$ with smooth order can be constructed. In this paper the concept of general auxiliary groups is introduced and it is shown that non-cyclic elliptic curves over $GF(p)$ or over an extension field of $GF(p)$, certain subgroups of the multiplicative group of such an extension field, and the Jacobian of a hyperelliptic curve are also suitable auxiliary groups. We give an extended list of expressions in p, including for example all the cyclotomic polynomials of low degree in p (which include the known cases $p-1$ due to den Boer [5] and $p+1$ due to Maurer [13]), such that, if for every large prime factor p of $|G|$ one of the expressions in the list is smooth, then breaking the Diffie-Hellman protocol and computing discrete logarithms are equivalent for G.

2 Various Types of Diffie-Hellman Oracles and Their Equivalence

The natural definition of a DH-oracle is the following.

Definition 1 A *DH-oracle* for a group G with respect to a given generator g takes as inputs two elements g^u and g^v and returns (without computational cost) the element g^{uv}.

In the following we show that certain apparently weaker oracles are almost as strong as a DH-oracle.

2.1 ε-DH-Oracles

Definition 2 For $\varepsilon > 0$, an *ε-DH-oracle* is a probabilistic oracle which returns for an input (g^u, g^v) the correct answer g^{uv} with probability at least ε, provided the input is uniformly distributed over $G \times G$. The *error* of the oracle's answer g^t to the input (g^u, g^v) is defined as $t - uv \pmod{|G|}$. A *translation-invariant* ε-DH-oracle is an ε-DH-oracle whose distribution of the error is the same for every input (g^u, g^v).

We assume that the given ε-DH-oracle is time-invariant, i.e., the error distribution can only depend on the input but remains the same when the oracle is called several times. The following lemma states that any ε-DH-oracle can be made translation-invariant. The proof idea is to randomize the input and was presented in [13].

Lemma 1 *An ε-DH-oracle for a cyclic group G can be transformed into a translation-invariant ε-DH-oracle. One call of the latter requires one call to the former and $O(\log |G|)$ group operations.*

Proof. Given the group elements $a = g^u$ and $b = g^v$ we can randomize the input by choosing r and s at random from $[0, |G|-1]$, providing the oracle with $a' = ag^r$ and $b' = bg^s$ and multiplying the oracle's answer $g^{(u+r)(v+s)+t} = g^{uv+rv+su+rs+t}$ with $(a^{-1})^s \cdot (b^{-1})^r \cdot g^{-rs} = g^{-(rv+su+rs)}$ to obtain g^{uv+t}. Note that a' and b' are random group elements and statistically independent of a and b. The ε-DH-oracle with randomized input is thus a translation-invariant ε-DH-oracle. □

Remark: If $|G|$ is unknown the input can also be randomized, where r and s are chosen at random from a larger interval. The resulting ε-DH-oracle is then "almost translation-invariant" and applicable in the proof of Theorem 1 if the interval is of size at least $2 \cdot |G|/(\varepsilon^2 \cdot \min\{s, 0.1\})$ (this is the reason for the greater number of group operations for this case in Theorem 1).

The straight-forward approach to transforming a translation-invariant ε-DH-oracle into a perfect DH-oracle appears to be to run it $O(1/\varepsilon)$ times until it produces the correct answer. However, because the Diffie-Hellman decision problem (that is, for given g^u, g^v, and g^w, to decide whether $g^{uv} = g^w$) is difficult, a more complicated approach must be used. In a first phase, which is independent of the actual input, the oracle's error distribution is determined. In the second phase, the oracle is used for a given input to compute the correct solution with overwhelming probability. In the case of a symmetric error distribution there can be several candidates, and the correct one can be determined similarly to the detection of the correct root in Lemma 2. A full proof of the following theorem is given in [14].

Theorem 1 *For every cyclic group G with generator g and known order $|G|$ and for every $\beta > 0$ there exists a DH-oracle algorithm which makes calls to an ε-DH-oracle and whose answer is correct with probability at least $1-\beta$. The number of oracle calls is $O(\log(1/\beta\varepsilon)/\varepsilon^4)$. If the order of G is unknown but all the prime factors of $|G|$ are greater than $(1+s)/\varepsilon$ for some $s > 0$, then the number of required calls to the ε-DH-oracle is $O(\log(1/\beta\varepsilon)/(\varepsilon^2 \cdot \min\{s, 0.1\})^2)$. The number of required group operations is $\log|G|$ or $\log(|G|/(\varepsilon^2 \cdot \min\{s, 0.1\}))$ times the number of oracle calls, respectively.*

Note that such an oracle is virtually equivalent to a perfect DH-oracle for our application because the correctness of the output of a probabilistic discrete logarithm algorithm can be tested, and because only a polynomially bounded number of oracle calls is required for the computation of a discrete logarithm.

Remark: Examples of ε-DH-oracles which can *not* be transformed into perfect oracles with our method when $|G|$ is unknown are those which answer the input (g^u, g^v) by one of the values $g^{uv+i|G|/z}$, where $z \leq 1/\varepsilon$ is a factor of $|G|$, and where all the values of i between 0 and $z-1$ are equally likely. If $|G|$ is known, the correct one of the z candidates can be found by $O((\log|G|)^2/\varepsilon + \log|G|/\varepsilon^2)$ group operations.

2.2 The Squaring Oracle

We call an oracle that answers the input g^u by $g^{(u^2)}$ (where u and u^2 are in $\mathbf{Z}_{|G|}$) a *squaring-DH-oracle*. Note that this is not an ε-DH-oracle for any constant $\varepsilon > 0$ because only one out of $|G|$ inputs is answered correctly, and this fraction vanishes with increasing $|G|$.

Let g^u and g^v be given. One can compute $g^{u+v} = g^u \cdot g^v$ and

$$g^{(u+v)^2} \cdot \left(g^{(u^2)}\right)^{-1} \cdot \left(g^{(v^2)}\right)^{-1} = g^{(u+v)^2-u^2-v^2} = g^{2uv} = (g^{uv})^2 \,. \qquad (1)$$

When given $|G|$, square roots in G can efficiently be computed. If $|G|$ is odd, the square root is unique, but if $|G|$ is even, there exist two square roots, g^{uv} and $g^{uv+|G|/2}$, which can be computed by a method of Massey [12] (see also Lemma 2). In this case, the correct square root g^{uv} is determined analogously to the detection of the correct root in the proof of Lemma 2 by computing u and v modulo the maximal power of 2 dividing $|G|$. Hence a squaring-DH-oracle is equally powerful as a perfect DH-oracle in a group G whose order is known.

A probabilistic squaring-DH-oracle for a group with known order that answers correctly only with probability ε (ε-squaring-DH-oracle) can be transformed into a translation-invariant ε^3-DH-oracle by randomizing the inputs in (1). The required complexity is $O((\log|G|)^2)$ group operations per call. This proves the following theorem.

Theorem 2 *For every cyclic group G with generator g and known order $|G|$ and for every $\beta > 0$ there exists a DH-oracle algorithm which makes calls to an ε-squaring-DH-oracle and whose answer is correct with probability at least $1-\beta$. The number of oracle calls is $O(\log(1/\beta\varepsilon^3)/\varepsilon^{12})$. The number of required group operations is $(\log|G|)^2$ times the number of oracle calls.*

2.3 The Security of Subgroups

In this section we assume that the order of G is known. We address the question whether a subgroup is more or less secure than the entire group with respect to the DH protocol. Although the statement of Corollary 5 below is very intuitive (and an analogous result holds trivially for the computation of discrete logarithms), the proofs of Theorems 3 and 4 are not trivial. First we state that a subgroup of G with smooth index is at most as secure as G.

Theorem 3 *Let G be a cyclic group with generator g, and let B be a smoothness bound, polynomial in $\log|G|$. For every B-smooth divisor r of $|G|$ there exists a DH-oracle algorithm for the group $\langle g^r\rangle$ which makes one call to the DH-oracle for $\langle g\rangle$ and uses a polynomial number of group operations per call.*

We first prove the following lemma on the computation of roots in cyclic groups.

Lemma 2 *Let G be a cyclic group with generator g, and let p be a prime divisor of $|G|$. One of the p-th roots of a p-th power in G can be computed in time $O((\log|G|)^2 + p\log|G|)$.*

Proof. The square root algorithm of Massey [12] can be generalized as follows. Let $|G| = p^j s$ (where $j \geq 1$ and $(p,s) = 1$), and let h be a p-th power in G. By the method of Pohlig and Hellman [18] we can compute the remainder k of the discrete logarithm of h to the base g with respect to p^j. Note that k is a multiple of p because h is a p-th power. Let $d \equiv -s^{-1} \pmod{p}$. The element

$$\left(g^{s\cdot\frac{k}{p}\cdot d}\right)^{-1} \cdot h^{\frac{sd+1}{p}}$$

is a p-th root of h. This algorithm requires $O((\log|G|)^2 + p\log|G|)$ operations in G. $\square$

We can now prove the theorem.

Proof of Theorem 3. Let $r = \prod_{i=1}^{s} p_i^{f_i}$, and let $p_i^{e_i}$ be the maximal powers dividing $|G|$ for $i = 1,\ldots,s$. The oracle for G answers the input (g^{ra}, g^{rb}) by

$g^{r^2ab} = g^{p_1^{2f_1} \cdots p_s^{2f_s} ab}$. We obtain $g^{rab} = g^{p_1^{f_1} \cdots p_s^{f_s} ab}$ by computing p_i-th roots and deciding immediately which of the p_i different roots is the correct one. For fixed i and for some $k = 2f_i - 1, 2f_i - 2, \ldots, f_i$, assume that we have already computed

$$g^{p_1^{f_1} \cdots p_{i-1}^{f_{i-1}} \cdot p_i^{k+1} \cdot p_{i+1}^{2f_{i+1}} \cdots p_s^{2f_s} ab} = g^{cp_i^{k+1} ab} \ ,$$

where $c = p_1^{f_1} \cdots p_{i-1}^{f_{i-1}} \cdot p_{i+1}^{2f_{i+1}} \cdots p_s^{2f_s}$ is explicitly known. According to the above lemma we can compute the p_i-th roots

$$g^{cp_i^k ab + j \cdot \frac{|G|}{p_i}}, \ j = 0, \ldots, p_i - 1 \ .$$

Because a and b can be obtained modulo $p_i^{e_i - f_i}$ directly from g^{ra} and g^{rb} by the method of Pohlig and Hellman [18] and c is explicitly known, and because $k \geq f_i$, we can compute $cp_i^k ab$ modulo $p_i^{e_i}$. We have $j \cdot |G|/p_i \equiv 0 \pmod{p_i^{e_i}}$ only for $j = 0$, and the correct root can be determined by computing the discrete logarithms of the candidates modulo $p_i^{e_i}$, using the Pohlig-Hellman method. Finally, we obtain g^{rab}. The running time is polynomial in $\log|G|$ if r is B-smooth. □

Conversely, in many cases a DH-oracle for a subgroup of G or a set of such oracles can be transformed into a DH-oracle for the entire group, and the following theorem gives a criterion for when this is the case. The proof is an application of our concept of computing with implicit representations introduced in Section 3.

Theorem 4 *Let G be a cyclic group with generator g and order $|G| = \prod_{i=1}^r p_i^{e_i}$, and let B be a smoothness bound which is polynomial in $\log|G|$. If for certain s_j there exist DH-oracles for the subgroups $G_j := \langle g^{s_j} \rangle$ $(j = 1, \ldots, t)$, and if for all $p_i > B$ there exists j such that p_i does not divide s_j, then there exists a polynomial-time DH-oracle algorithm for G with respect to g which calls each subgroup oracle at most $\log|G| / \log B$ times.*

Proof. Let g^u and g^v be given. We compute g^{uv} by using the available oracles for subgroups. Let $m_i := p_i^{e_i}$, $M_i := |G|/m_i$ and $N_i :\equiv M_i^{-1} \pmod{m_i}$. For prime factors $p_i \leq B$, u and v, and hence also uv, can be computed in polynomial time modulo m_i by the Pohlig-Hellman method [18]. For a prime factor $p_i > B$ let j be such that p_i does not divide s_j. We apply the oracle for G_j to $(g^{s_j})^u = (g^u)^{s_j}$ and $(g^{s_j})^v$ to obtain $(g^{s_j})^{u \cdot v}$, where u, v and $u \cdot v$ are modulo $|G|/s_j$. Because s_j divides M_i, we can compute

$$U_i := g^{M_i \cdot (u \cdot v)} = \left(g^{s_j (u \cdot v)} \right)^{\frac{M_i}{s_j}} \ ,$$

where $u \cdot v$ is modulo m_i. Finally, g^{uv} is computable by Chinese remaindering with implicitly represented arguments by applying only group operations in G:

$$g^{uv} = g^{\sum_i M_i N_i (u \cdot v)} = \prod_i U_i^{N_i} .$$

□

Corollary 5 *Consider a group $G = \langle g \rangle$ and a subgroup $H = \langle g^k \rangle$ of G with smooth index k. The DH problem for H is polynomial-time equivalent to the DH problem for G.*

3 Conditions for Equivalence Between the Diffie-Hellman Problem and Computing Discrete Logarithms

3.1 Computing with Implicit Representations

Let G be a cyclic group generated by g for which the prime factorization of the order $|G|$ is known, and for which a DH-oracle is given. Let p be a prime factor of $|G|$. Every element y of the field $GF(p)$ corresponds to an equivalence class of elements of G (consisting of those whose discrete logarithm is congruent to y modulo p). Any member a of the equivalence class is called an *implicit representation* of y and, conversely, y is called implicitly represented by a. We write $y \leadsto a$. The following operations on elements of $GF(p)$ can be performed on their implicit representations, where the result is also obtained only in an implicit representation. Let y and z be elements of $GF(p)$, with $y \leadsto a$, $z \leadsto b$. Because $y = z$ if and only if $a^{|G|/p} = b^{|G|/p}$, equality of two implicitly represented elements of $GF(p)$ can be tested by $O(\log |G|)$ group operations. Furthermore we have $y + z \leadsto a \cdot b$, $yz \leadsto \mathrm{DH}(a, b)$, and $-y \leadsto a^{-1} = a^{|G|-1}$, and these implicit operations in $GF(p)$ require a single group operation in G, a call to the DH-oracle, and $O(\log |G|)$ group operations, respectively.

In order to simplify the notation, we also introduce the notion of a power-DH-oracle (PDH_e) that computes an implicit representation of the e-th power of an implicitly represented element. A possible implementation of a PDH_e-oracle is to use a (fixed) algorithm for computing powers in a group (e.g. 'square and multiply') for obtaining an implicit representation of y^e, denoted by $\mathrm{PDH}_e(a)$, by $O(\log e)$ calls to a normal DH-oracle (remember that $y \leadsto a$). In particular we can compute inverses of implicitly represented elements because $y^{-1} \leadsto \mathrm{PDH}_{p-2}(a)$. Any computation in $GF(p)$ can be performed on implicit representations whenever it makes use only of addition, subtraction, multiplication, division and equality testing. We call these operations *algebraic*.

3.2 Auxiliary Groups

The next theorem states that for a cyclic group G breaking the DH protocol and computing discrete logarithms are polynomial-time equivalent if an appropriate auxiliary group defined over the field $GF(p)$ is given for each large prime factor p of $|G|$. First we define two properties of such auxiliary groups.

Definition 3 Let P be a fixed expression, polynomial in $\log p$, and let M be a fixed constant. A finite (additively written) group H is said to be *defined algebraically over* $GF(p)$ if, for some $m \leq M$, the elements of H can be represented as m-tuples of elements of $GF(p)$ and if the group operation in this representation can be carried out by at most P algebraic operations in $GF(p)$. We say that H has the *algebraic embedding property* if, when given $x \in GF(p)$, an element $c \in H$ can be constructed by at most P algebraic operations in $GF(p)$ such that x can be computed efficiently when given c. (Typically, x is a coordinate of c.)

Theorem 6 *Let G be a cyclic group with generator g, and let B be a smoothness bound, polynomial in $\log|G|$. Assume that $|G|$ and its factorization $|G| = \prod_{i=1}^{s} p_i^{e_i}$ are known, that every prime factor p of $|G|$ greater than B is single and that for every such p, a finite abelian group H_p with rank $r = O(1)$, algebraically defined over $GF(p)$ and with the algebraic embedding property, is given whose order $|H_p|$ is B-smooth and known or computable in time polynomial in $\log p$. Then breaking the Diffie-Hellman protocol for G with respect to g is polynomial-time equivalent to computing discrete logarithms in G to the base g.*

The complexity of the computation of a discrete logarithm modulo p in G is $O(M^2 B^r \log p \log|G|/\log B)$ group operations in G, $O(M^2(\log p)^2)$ operations in H_p with implicitly represented elements, and $O(M^2(\log p)^2 + M\log p \cdot B^r/\log B)$ explicit operations in H_p.

In case of a multiple prime factor p greater than B, that is if p^e divides $|G|$ for some $e > 1$, the desired equivalence holds with respect to a DH-oracle for one of the subgroups $\langle g^{d\cdot p^{e-1}}\rangle$ (instead of the DH-oracle for G) where $d \cdot p^{e-1}$ divides $|G|/p$, or if a polynomial-time algorithm for computing p-th roots in G is available.

The complexities stated in the theorem can be reduced by a time-memory trade-off. The use of elliptic curves and subgroups of extension fields as auxiliary groups is discussed in the next sections. In [21] it is shown that Jacobians of hyperelliptic curves are also suitable auxiliary groups.

Proof. Let $a = g^s$ be a given element of G for which the discrete logarithm s should be computed using a DH-oracle for G. We assume first that all the large prime factors of $|G|$ are single. Let p be such a prime factor. We consider the problem of computing the element x of $GF(p)$ such that $s \equiv x \pmod p$ using the auxiliary group $H = H_p$. The basic idea is to embed x into an implicitly represented element of H and to compute its explicit representation.

Using the algebraic embedding property, the implicit representation of a group element c in H can be computed such that x can efficiently be obtained from the *explicit* coordinates of c. We address the problem of finding c explicitly. In the special case where H is cyclic, the following method corresponds to the Pohlig-Hellman algorithm [18] with implicitly represented arguments. Let H be isomorphic to $\mathbf{Z}_{n_1} \times \cdots \times \mathbf{Z}_{n_r}$ such that n_{j+1} divides n_j for $j = 1, \ldots, r-1$, and let $h_1, \ldots, h_r$ be such that H is the internal product of the cyclic subgroups $\langle h_1\rangle, \ldots, \langle h_r\rangle$, i.e., $H = \langle h_1\rangle \times \cdots \times \langle h_r\rangle$. (If no generator set for H is known it can efficiently be computed by a method based on trial and error which is described in [14].) The element $c \in H$ has a unique representation $c = \sum_{j=1}^{r} k_j h_j$ $(0 \le k_j < n_j)$.

We describe the first and second iteration step of an algorithm that computes k_j modulo the highest power of a fixed prime factor q of $|H|$ dividing n_j for all $j = 1, \ldots, r$. The algorithm uses v_j $(j = 1, \ldots, r)$ as local variables (initialized by $v_j \leftarrow 0$).

For the first step, let α_1 be the number of generators h_j whose order contains the same number of factors q as n_1. In other words, $(n_1/q)h_j$ is different from the unity e of H exactly for $j = 1, \dots, \alpha_1$. Because H is algebraically defined over $GF(p)$, an implicit representation of $(n_1/q)c$ can be computed from the implicit representation of c by $O(\log|H|)$ operations in H with implicitly represented elements. For all $(t_1, \dots, t_{\alpha_1}) \in \{0, \dots, q-1\}^{\alpha_1}$, we compute (explicitly) $(n_1/q)t_1h_1 + \cdots + (n_1/q)t_{\alpha_1}h_{\alpha_1}$, transform the coordinates into implicit representations and compare the points with $(n_1/q)c$. Equality indicates that the t_j are congruent to the coefficients k_j modulo q. We set $v_j \longleftarrow t_j$ for $1 \le j \le \alpha_1$.

For the second step, let α_2 be the number of points h_j whose order contains at most one factor q less than n_1, i.e., $(n_1/q^2)h_j \neq e$ for $j = 1, \dots, \alpha_2$. The (implicit representations of the) points

$$\frac{n_1}{q^2}(v_1q + t_1)h_1 + \cdots + \frac{n_1}{q^2}(v_{\alpha_1}q + t_{\alpha_1})h_{\alpha_1} + \frac{n_1}{q^2}t_{\alpha_1+1}h_{\alpha_1+1} + \cdots + \frac{n_1}{q^2}t_{\alpha_2}h_{\alpha_2}$$

are computed for all $(t_1, \dots, t_{\alpha_2}) \in \{0, \dots, q-1\}^{\alpha_2}$ until equality with the implicitly represented point $(n_1/q^2)c$ holds. Then assign $v_j \longleftarrow v_jq + t_j$ for $j = 1, \dots, \alpha_1$ and $v_j \longleftarrow t_j$ for $j = \alpha_1 + 1, \dots, \alpha_2$. When this is done up to the maximal q-power dividing n_1, k_j is congruent to v_j modulo the highest power of q dividing n_j for $j = 1, \dots, r$. After running the algorithm for all primes q dividing $|H|$, one can compute the coefficients k_j modulo n_j by Chinese remaindering, and x can then be obtained by computing c explicitly. The complexity of the computation of x is $O(m^2(\log p)^2)$ operations in H with implicitly represented elements, $O(m^2B^r \log p \log|G|/(r \log B))$ operations in G and $O(m^2r(\log p)^2 + m\log p \cdot B^r/\log B)$ explicit operations in H. (Note that p^m is an upper bound for $|H|$ because H is defined algebraically over $GF(p)$.) Again because H is defined algebraically over $GF(p)$, the running time is polynomial if B is polynomial in $\log|G|$, and if $r = O(1)$. The algorithm can be sped up by a time-memory tradeoff similar to the baby-step giant-step tradeoff for the computation of discrete logarithms.

We finally consider the case of *multiple* large prime factors of $|G|$. If p^e divides $|G|$ (with $e > 1$), the discrete logarithm s must be computed explicitly modulo p^e instead of modulo p. We write $x \equiv \sum_0^{e-1} x_ip^i \pmod{p^e}$ with $x_i \in GF(p)$ for $i = 0, \dots, e-1$. Let $k \le e-1$, assume that $x_0, \dots, x_{k-1}$ are already computed (note that x_0 can be computed as above), and consider the problem of computing x_k. Let $a' := a \cdot g^{-x_0 - \cdots - x_{k-1}p^{k-1}}$. Then $a' = (g^{p^k})^{x_k + p\cdot l}$ for some l. From a', x_k can be obtained in either of two ways: If a DH-oracle for one of the subgroups $\langle g^{d\cdot p^{e-1}} \rangle$, where $d \cdot p^{e-1}$ divides $|G|/p$, is available, then x_k can be computed from $(a')^{d\cdot p^{e-1-k}} = (g^{d\cdot p^{e-1}})^{x_k+p\cdot l}$ by use of this oracle as described. Alternatively, assume that p-th roots can be computed in G. If $a'' := g^{x_k + p\cdot l'}$ (for some l') is computed first, x_k can be obtained as usual. In order to get a'', it suffices to compute any p^k-th root (k times the p-th root) of a' because p divides $|G|/p^k$.

From s modulo p^e for all the maximal powers of the large prime factors of $|G|$, s can be obtained by Chinese remaindering. This concludes the proof. $\square$

3.3 Elliptic Curves as Auxiliary Groups

Elliptic curves over $GF(p)$ or an extension field are suitable auxiliary groups when they have smooth order. (Note that elliptic curves are abelian groups of rank at most 2.) In [13] this was shown for *cyclic* elliptic curves over *prime* fields. It is proved there that, under an unproven number-theoretic conjecture about smooth numbers in small intervals, for every cyclic group G there exists a short side information string S (containing the parameters of a smooth elliptic curve for each large prime factor of $|G|$) such that given S, the DH and DL problems are equivalent for G.

The group order of Jacobians of hyperelliptic curves of genus 2 varies in a larger interval of size $[n - \Theta(n^{3/4}), n + \Theta(n^{3/4})]$, but the more detailed results about the distribution of the orders which are proved in [1] are not sufficient to prove the existence of the side information string without unproven conjecture. The reason is that in [1] the existence of Jacobians with *prime* order is proved, whereas Jacobians with *smooth* order are required here.

For certain expressions $A(p)$, elliptic curves over $\mathbf{F}_p$ with order $A(p)$ can explicitly be constructed. The curve over $\mathbf{F}_p$ defined by the equation $y^2 = x^3 - Dx$ has order $p+1$ if $p \equiv 3 \pmod 4$, and the curve $y^2 = x^3 + D$ has also order $p+1$ if $p \equiv 2 \pmod 3$. Thus if $p \not\equiv 1 \pmod{12}$, elliptic curves of order $p+1$ are explicitly constructable. We will show later that the subgroup of order $p+1$ of $\mathbf{F}^*_{p^2}$ is a useful auxiliary group for all p. The following statements about the orders of curves defined by the equations above in the case they are *not* $p+1$ are proved in [8].

If $p \equiv 1 \pmod 4$, then p can uniquely be represented as a product in the ring $\mathbf{Z}[i]$ of Gaussian integers: $p = \pi\overline{\pi} = (a+bi)(a-bi) = a^2 + b^2$, and $\pi \equiv 1 \pmod{2+2i}$. The curves $y^2 = x^3 - Dx$ have the orders $p+1 \pm 2a$ or $p+1 \pm 2b$, and the four orders occur equally often.

Let $\omega := (-1+\sqrt{-3})/2$. If $p \equiv 1 \pmod 3$, then p can uniquely be represented as a product in the ring $\mathbf{Z}[\omega]$: $p = \pi\overline{\pi} = (a+b\omega)(a-b\omega) = a^2 - ab + b^2$, and $\pi \equiv 2 \pmod 3$. The curves $y^2 = x^3 + D$ have the orders $p+1 \pm 2a$, $p+1 \pm a \mp 2b$, or $p+1 \pm (a+b)$, and the six orders occur equally often.

If $p \equiv 1 \pmod 4$ or $p \equiv 1 \pmod 3$, curves with the above orders are explicitly constructable by varying D. The orders are computable in polynomial time [19].

3.4 Subgroups of Finite Fields as Auxiliary Groups

We refer to [16] for an introduction to finite fields. The group $\mathbf{F}^*_{p^n}$ and hence every subgroup is cyclic. The field $\mathbf{F}_{p^n}$ is an n-dimensional vector space over $\mathbf{F}_p$ and its elements can be represented as n-tuples of $\mathbf{F}_p$-elements with respect to some basis. Let α be an element of $\mathbf{F}_{p^n}$. Let $\alpha_i := \alpha^{p^i}$ for $i = 0, \ldots, n-1$. Then $\{\alpha_0, \ldots, \alpha_{n-1}\}$ is called a *normal basis* if it is linearly independent in which case α is called a *normal element*. Let $\boldsymbol{\alpha} := (\alpha_0, \ldots, \alpha_{n-1})$. The matrix T in $(\mathbf{F}_p)^{n \times n}$ satisfying $\alpha_0 \boldsymbol{\alpha} = T\boldsymbol{\alpha}$ is called the *multiplication table* of the basis.

A normal basis can be found efficiently by trial and error, and its multiplication table can be determined by solving a system of linear equations over $\mathbf{F}_p$.

Let H be a subgroup of $\mathbf{F}_{p^n}^*$. The group operation in H is a multiplication in $\mathbf{F}_{p^n}^*$ and requires $O(n^3)$ multiplications in $\mathbf{F}_p$.

Membership in H can be characterized by an equation over $\mathbf{F}_{p^n}$. Let β be an element of $\mathbf{F}_{p^n}$. Because $\mathbf{F}_{p^n}^*$ is cyclic, β belongs to H if and only if $\beta^{|H|} = 1$. The element β can be represented by its coordinates $(y_0, y_1, \ldots, y_{n-1})$ (with $y_i \in \mathbf{F}_p$) in the normal basis, i.e., $\beta = \sum_{i=0}^{n-1} y_i \alpha_i$. In this representation the characteristic equation of H is equivalent to a system of n polynomial equations in the y_i. The polynomials depend on the multiplication table.

For some orders $|H|$, the polynomials can easily be computed and have small degree, in particular if $|H|$ is a sum of p-powers, multiplied with only small factors. The p^u-th power of the sum $\sum y_i \alpha_i$ is equal to the sum of the p^u-th powers of the summands because $\mathbf{F}_{p^n}$ has characteristic p. In addition we have $y_i^p = y_i$ and $\alpha_i^p = \alpha_{i+1}$ (where the index is reduced modulo n). Hence β^{p^u} is represented by the coordinates $(y_{n-u}, y_{n-u+1}, \ldots, y_n, y_0, \ldots, y_{n-u-1})$.

We prove the algebraic embedding property by showing directly that, given an implicit representation of x, an implicit representation of a point β of H can be computed such that x (or $x + d$ for some d) is one of the coordinates of β. To do this, fix some of the other coordinates (for example by assigning the value 0) and solve the implicitly represented equations to get implicitly represented values for the remaining coordinates such that β belongs to H. The number of unknowns over $\mathbf{F}_p$ in this system depends on the cardinality of H. If we solve for k different $\mathbf{F}_p$-coordinates simultaneously, then the expected number of trials until an element of H is found is $p^{n-k}/|H|$.

It is much easier to solve a univariate polynomial equation than to solve a system of multivariate polynomial equations. We show that it is sufficient to solve one equation for one unknown when the group H has order $|H| = p^{n-k} + p^{n-2k} + \cdots + 1$ for some divisor $k < n$ of n. Let $l := n/k$, and let $\{\alpha'_0, \ldots, \alpha'_{l-1}\}$ be a normal basis of $\mathbf{F}_{p^n}$ over $\mathbf{F}_{p^k}$. An element β of $\mathbf{F}_{p^n}$, represented by $(\beta'_0, \ldots, \beta'_{l-1})$ with $\beta'_i \in \mathbf{F}_{p^k}$, belongs to H if and only if $(\sum_{i=0}^{l-1} \beta'_i \alpha'_i)^{|H|} = 1$, or equivalently

$$\left(\sum_{i=0}^{l-1} \beta'_i \alpha'_{i+l-1}\right) \cdot \left(\sum_{i=0}^{l-1} \beta'_i \alpha'_{i+l-2}\right) \cdots \left(\sum_{i=0}^{l-1} \beta'_i \alpha'_i\right) = 1$$

(where the indices are reduced modulo l). Because $(\beta^{|H|})^{p^k-1} = \beta^{p^n-1} = 1$ for all β, $\beta^{|H|}$ is an element of $\mathbf{F}_{p^k}$, and because $\alpha_0 + \alpha_1 + \cdots + \alpha_{l-1}$ (the trace of α_0, denoted by $\mathrm{Tr}(\alpha_0)$) is an element of $\mathbf{F}_{p^k}$, all the coefficients are automatically equal, and it suffices to solve one instead of l equations. Thus the characteristic equation of the subgroup H with this order leads to an l-degree polynomial in $\beta'_0, \ldots, \beta'_{l-1}$ over $\mathbf{F}_{p^k}$. We assign the (implicitly represented) x to one of the k coordinates of β'_0, and 0 to $\beta'_1, \ldots, \beta'_{l-2}$ (for example) to get an l-degree polynomial for β'_{l-1} with implicitly represented coefficients. The order of H is such that this polynomial has one expected solution. (If no solution is found one can vary the coefficients $\beta'_1, \ldots, \beta'_{l-2}$.)

The roots of a polynomial $f(\gamma)$ over a finite field $\mathbf{F}_{p^k}$ can be computed by the following randomized algorithm due to Berlekamp. The key idea is to factor the polynomial $f(\gamma)$ into

$$\gcd(f(\gamma), (\gamma+\delta)^{\frac{p^k-1}{2}} - 1) \quad \text{and} \quad \gcd(f(\gamma), (\gamma+\delta)^{\frac{p^k-1}{2}} + 1)$$

for some $\delta \in \mathbf{F}_{p^k}$. This is repeated with different δ and leads to the linear factors of $f(\gamma)$.

The computation of polynomial gcd's, and thus the entire root-finding algorithm, require only algebraic operations in $\mathbf{F}_{p^k}$, and the latter can be reduced to algebraic operations (and equality tests) in $\mathbf{F}_p$ (with respect to a normal basis representation). The implicit representations of the roots of an implicitly represented polynomial are thus efficiently computable. The complexity of computing one root is $O(nl \log l \log p \cdot (k^2 + \log|G|))$ group operations and $O(n^2 k \log l)$ calls to the DH-oracle. We conclude that H with order $p^{n-k} + p^{n-2k} + \cdots + 1$ (where n is polynomial in $\log p$) fulfills the requirements of Theorem 6 if its order is smooth.

If $|H|$ is not of this form, but a sum of p-powers (with small coefficients), a system of multivariate polynomial equations with several unknowns must be solved. Let $F(x)$ be a polynomial of positive degree which divides $x^n - 1$. By Φ_n we denote the n-th cyclotomic polynomial. Because cyclotomic polynomials are irreducible and $x^n - 1 = \prod_{d|n} \Phi_d(x)$, at least one cyclotomic polynomial Φ divides F, and smoothness of $F(p)$ implies smoothness of $\Phi(p)$. Therefore, we can assume without loss of generality that F is a cyclotomic polynomial and, again without loss of generality, that $F(x) = \Phi_n(x) = \sum_{j=0}^{\varphi(n)} c_j x^j$. Let H be the (unique) subgroup of $\mathbf{F}_{p^n}^*$ with order $|H| = \Phi_n(p)$. In a normal basis representation any $\beta = \sum_{i=0}^{n-1} y_i \alpha_i$ (with $y_i \in \mathbf{F}_p$) is an element of H if and only if $(\sum_{i=0}^{n-1} y_i \alpha_i)^{\sum c_j p^j} = 1$, which is equivalent to

$$\prod_{c_j \geq 0} \left(\sum_{i=0}^{n-1} y_i \alpha_i \right)^{p^j c_j} - \prod_{c_j < 0} \left(\sum_{i=0}^{n-1} y_i \alpha_i \right)^{p^j (-c_j)} = 0 \ ,$$

and leads to a system of n polynomials in the y_i over $\mathbf{F}_p$ of degree at most $\varphi(n) \cdot \max\{|c_j| : j = 0, \ldots, \varphi(n)\}$. Because $|H| \approx p^{\varphi(n)}$ we have to solve the (implicitly represented) polynomial equations for $n - \varphi(n)$ unknowns. Gröbner bases are a tool for solving systems of polynomial equations. They lead to equivalent systems of equations which have triangular form, such that a method for solving univariate equations (as Berlekamp's algorithm) suffices to solve the whole system. For an introduction to Gröbner bases see [7], and for a detailed description of the computations see [21]. The idea is to compute the polynomials (with implicitly represented coefficients) of a Gröbner basis of the polynomial ideal generated by the polynomials of the equations. The algorithm for the Gröbner basis computation, due to Buchberger, requires only algebraic polynomial arithmetic and can therefore be executed on implicitly represented arguments. The second step is to solve the separated system of implicitly represented polynomials by Berlekamp's method for univariate polynomials. The complexity of the computations is polynomial if $n = O(1)$.

We conclude that the subgroup H of $\mathbf{F}_{p^n}^*$ of order $\Phi_n(p)$, $n = O(1)$, is applicable in Theorem 6 if it has smooth order. For example, smoothness of $\Phi_6(p) = p^2 - p + 1$, $\Phi_8(p) = p^4 + 1$, or $\Phi_9(p) = p^6 + p^3 + 1$ implies that an appropriate group H_p over $GF(p)$ is constructable. As mentioned, this is now proved for $F(p)$ for any non-trivial polynomial $F(x)$ dividing $x^n - 1$ if $n = O(1)$. Other examples are the alternating sums $p^{2l} - p^{2l-1} + - \cdots - p + 1$ when $l = O(1)$.

3.5 The Main Equivalence Result

Corollary 7 *Let G be a cyclic group with generator g, and let B be a smoothness bound, polynomial in $\log|G|$. Then there exists a list of expressions $A(p)$ in p with the following property: if for every prime factor p of $|G|$ greater than B, at least one of the expressions $A(p)$ is B-smooth, then breaking the Diffie-Hellman protocol in G with respect to g is polynomial-time equivalent to computing discrete logarithms in G to the base g. (In the case of a multiple large prime factor p of $|G|$, the equivalence holds with respect to breaking the DH protocol in one of a certain subset of subgroups of G, or if an algorithm for computing p-th roots in G is given.) The list contains the following expressions:*

$$p - 1,\ p + 1,$$

$$p + 1 \pm 2a,\ p + 1 \pm 2b,$$

if $p \equiv 1 \pmod 4$, where $p = a^2 + b^2$ and $a + bi \equiv 1 \pmod{2 + 2i}$,

$$p + 1 \pm 2a,\ p + 1 \mp a \pm 2b,\ p + 1 \pm (a + b),$$

if $p \equiv 1 \pmod 3$, where $p = a^2 - ab + b^2$ and $a + b\omega \equiv 2 \pmod 3$,

$$\frac{(p^k)^l - 1}{p^k - 1} = (p^k)^{l-1} + \cdots + p^k + 1,$$

where $k, l = O((\log p)^c)$ and $c = O(1)$, and $\Phi_n(p)$, where $n = O(1)$ and Φ_n is the n-th cyclotomic polynomial. □

4 Construction of Secure Diffie-Hellman Groups

It appears desirable to use a group G in the DH protocol for which the equivalence to computing discrete logarithms can be proved. However, such reasoning should be used with care because it is conceivable that knowledge of the auxiliary groups makes computing discrete logarithms easier. There are three possible scenarios for such an equivalence:

1. When given G it is easy (also for an opponent) to find suitable auxiliary groups.
2. The designer of the group G knows suitable auxiliary groups but they are difficult to find for an opponent.

3. The designer of the group G knows that suitable auxiliary groups exist, without knowing them.

In the first case the equivalence holds, whereas in the other two cases breaking the DH protocol is at least as difficult as computing discrete logarithms when the auxiliary groups are known. Note that the second case can always be transformed into the first by publishing the suitable auxiliary groups. Of course, because this information can only help an opponent in breaking the Diffie-Hellman protocol, there is no reason for the designer of the group to make it public.

Constructing a group G of the third type is trivial: choose a (secret) arbitrary large smooth number m and search for a prime p in the interval $[m - 2\sqrt{m} + 1, m + 2\sqrt{m} + 1]$. A group G whose order contains only such large prime factors satisfies the third property. Note that it is easy to construct, for a given n, a DH-group G whose order is a multiple of n. One possibility is to find a multiple l of n (where l/n is small) such that $l + 1$ is prime and to use $G = GF(l + 1)^*$. An alternative, which may be more secure, is to use the construction of Lay and Zimmer [10] for finding an elliptic curve of order n.

The second case is somewhat more involved. Such a group G can be obtained by choosing a large smooth number m and using the method of Lay and Zimmer [10] for constructing a prime p together with an elliptic curve of order m.

We now consider efficient constructions for the first case. We generalize a method, presented in [20] by Vanstone and Zuccherato, for constructing a large prime p such that either a quarter of the curves $y^2 = x^3 - Dx$ or every sixth curve of the form $y^2 = x^3 + D$ have smooth order. We show how to construct primes $p = a^2 + (k \pm 1)^2$ (for a fixed k with l digits) such that $a^2 + k^2$, which is then one of the possible orders of the curves $y^2 = x^3 - Dx$ over $\mathbf{F}_p$ (see Section 3.3), is smooth. First, l'-digit numbers x_1, x_2, y_1, and y_2 are chosen at random. Define $u + vi := (x_1 + y_1 i)(x_2 + y_2 i)$, that is $u = x_1 x_2 - y_1 y_2$, $v = x_1 y_2 + x_2 y_1$. u and v have approximately $2l'$ digits. If $\gcd(u, v)$ divides k (otherwise choose again), one can compute numbers c and d (of at most $2l' + l$ digits) such that $cv + du = k$. Define $a := cu - dv$, and restart the process if a is even. Then $a + ki = (c + di)(u + vi) = (c + di)(x_1 + y_1 i)(x_2 + y_2 i)$. The process is repeated until $a^2 + k^2 = (c^2 + d^2)(x_1^2 + y_1^2)(x_2^2 + y_2^2)$ is s-digit-smooth, which happens with probability approximately $((4l' + 2l)/s)^{-(4l'+2l)/s} \cdot (2l'/s)^{-2l'/s} \cdot (2l'/s)^{-2l'/s}$. This follows from the fact that for every fixed u, $\psi(n, n^{1/u})/n = u^{-(1+o(u))u}$, where $\psi(n, y)$ denotes the number of integers $\leq n$ with no prime divisor $\geq y$ (see [4]). Smoothness can be tested with the elliptic curve factoring algorithm [11]. Because a and k are odd, exactly one of the expressions $a + (k \pm 1)i$ is congruent to 1 modulo $2 + 2i$. Let $\alpha := a + (k \pm 1)i$, respectively. Repeat the computations until $p := \alpha\overline{\alpha} = a^2 + (k \pm 1)^2$ is prime. According to Section 3.3, a quarter of the curves $y^2 = x^3 - Dx$ over $\mathbf{F}_p$ have smooth order $a^2 + k^2$. Hence p is an $(8l' + 2l)$-digit prime such that an elliptic curve with s-digit-smooth order is constructable over $\mathbf{F}_p$. The expected number of trials is

$$O\left(\left(\frac{4l' + 2l}{s}\right)^{\frac{4l'+2l}{s}} \cdot \left(\frac{2l'}{s}\right)^{\frac{4l'}{s}} \cdot (8l' + 2l)\right) . \tag{2}$$

In a similar way, primes can be constructed such that curves of type $y^2 = x^3 + D$ have smooth order (see [21] for a detailed description). More precisely, we generate primes $p = a^2 - a(k \pm 1) + (k \pm 1)^2$ (where $a + (k \pm 1)\omega \equiv 2 \pmod 3$) such that $a^2 - ak + k^2$, which is one of the orders of the curves $y^2 = x^3 + D$ over $\mathbf{F}_p$, is s-digit-smooth. The expected number of repetitions is again given by (2).

In case of a small k, an L-digit prime p such that an s-digit-smooth curve is constructable over $\mathbf{F}_p$ can be found by $O((L/(\sqrt{8} \cdot s))^{L/s} \cdot L)$ trials instead of $O((L/s)^{L/s} \cdot L)$ trials when varying p among L-digit numbers until p is prime and one of the considered curves is s-digit-smooth. For example, a 100-digit prime p such that a 10-digit-smooth curve over $\mathbf{F}_p$ is efficiently constructable can be found by approximately $3 \cdot 10^6$ trials (instead of about 10^{11} trials when using the straightforward strategy).

5 Concluding Remarks

Our results imply that the DH problem is at least as difficult as the DL problem *with knowledge* of suitable auxiliary groups. Although it appears unlikely, it is possible that this knowledge helps computing discrete logarithms.

Throughout this paper, we have assumed that the group order and its factorization are known. This is the case in most known applications. It is conceivable that knowledge of $|G|$ could be of some help in computing discrete logarithms. For example, the algorithm of Pollard (see [15]) requires knowledge of the group order. For the case of unknown factorization of the group order, note that in some cases the parameters of a smooth auxiliary group H_p allow to compute p. If an appropriate multiplicative subgroup of an extension field of $\mathbf{F}_p$ has smooth order, then p can be found efficiently as a factor of $|G|$ (see [2]). The parameters A and B of a smooth elliptic curve over $\mathbf{F}_p$ defined by $y^2 = x^3 + Ax + B$ do generally not allow to find p efficiently by the method of [11], because no point can be generated on the curve modulo $|G|$.

In [14] a method is described, presented initially in [21] and independently considered in [3], for obtaining stronger results under the assumption of efficient DH-oracle *algorithms* using algebraic operations for certain groups. For example, a cyclic auxiliary group H_p whose order contains a large prime factor q and a smooth auxiliary group H_q over $\mathbf{F}_q$ are sufficient under the assumption of a polynomial-time DH-oracle algorithm for H_p, using algebraic operations in $\mathbf{F}_p$. The idea is to execute the oracle algorithm on implicitly represented arguments.

Acknowledgments

We would like to thank Dan Boneh for interesting discussions related to the subject of this paper, and an anonymous program committee member for helpful comments.

References

1. L.M. Adleman and M.A. Huang, Primality testing and abelian varieties over finite fields, *Lecture Notes in Mathematics*, vol. 1512, Springer-Verlag, 1992.
2. E. Bach and J. Shallit, Factoring with cyclotomic polynomials, *Math. Comp.*, vol. 52, pp. 201-219, 1989.
3. D. Boneh and R.J. Lipton, Algorithms for black-box fields and their application to cryptography, preprint, 1995.
4. E.R. Canfield, P. Erdös and C. Pomerance, On a problem of Oppenheim concerning "Factorisatio Numerorum", *J. Number Theory*, vol. 17, pp. 1-28, 1983.
5. B. den Boer, Diffie-Hellman is as strong as discrete log for certain primes, *Advances in Cryptology – CRYPTO '88*, Lecture Notes in Computer Science, vol. 403, pp. 530-539, Berlin: Springer-Verlag, 1989.
6. W. Diffie and M.E. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.
7. K.O. Geddes, S.R. Czapor and G. Labhan, Algorithms for computer algebra, Kluwer Academic Publisher, 1992.
8. K. Ireland and M. Rosen, A classical introduction to modern number theory, Springer-Verlag, 1982.
9. N. Koblitz, Elliptic curve cryptosystems, *Math. Comp.*, vol. 48, pp. 203-209, 1987.
10. G.-J. Lay and H.G. Zimmer, Constructing elliptic curves with given group order over large finite fields, preprint, 1994.
11. H.W. Lenstra, Jr., Factoring integers with elliptic curves, *Annals of Mathematics*, vol. 126, pp. 649-673, 1987.
12. J.L. Massey, Advanced Technology Seminars Short Course Notes, pp. 6.66-6.68, Zürich, 1993.
13. U.M. Maurer, Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms, *Advances in Cryptology - CRYPTO '94*, Y. Desmedt (ed.), Lecture Notes in Computer Science, Berlin: Springer-Verlag, vol. 839, pp. 271-281, 1994.
14. U.M. Maurer and S. Wolf, On the complexity of breaking the Diffie-Hellman protocol, Tech. Rep. 244, Computer Science Department, ETH Zürich, April 1996. (Accessible at http://www.inf.ethz.ch/publications/isc.html)
15. K.S. McCurley, The discrete logarithm problem, in *Cryptology and computational number theory*, C. Pomerance (ed.), Proc. of Symp. in Applied Math., vol. 42, pp. 49-74, American Mathematical Society, 1990.
16. A.J. Menezes (ed.), Applications of finite fields, Kluwer Academic Publishers, 1992.
17. V. Miller, Uses of elliptic curves in cryptography, *Advances in Cryptology - CRYPTO '85*, Lecture Notes in Computer Science, Springer-Verlag, vol. 218, pp. 417-426, 1986.
18. S.C. Pohlig and M.E. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance, *IEEE Transactions on Information Theory*, vol. 24, no. 1, pp. 106-110, 1978.
19. R. Schoof, Elliptic curves over finite fields and the computation of square roots mod p, *Math. Comp.*, vol. 44, No. 170, pp. 483-494, 1985.
20. S.A. Vanstone and R.J. Zuccherato, Elliptic curve cryptosystems using curves of smooth order over the ring $\mathbf{Z}_n$, Preliminary version, 1994.
21. S. Wolf, Diffie-Hellman and discrete logarithms, Thesis, March 1995.