Cheon's algorithm, pairing inversion and the discrete logarithm problem

David J. Mireles Morales

Mathematics Department
Royal Holloway, University of London
david.mireles@gmail.com

Abstract. We relate the fixed argument pairing inversion problems (FAPI) and the discrete logarithm problem on an elliptic curve. This is done using the reduction from the DLP to the Diffie-Hellman problem developed by Boneh, Lipton, Maurer and Wolf. This approach fails when only one of the FAPI problems can be solved. In this case we use Cheon's algorithm to get a reduction.

1 Introduction

Pairing-based cryptography has become one of the most active research areas in public key cryptography. The security of a pairing-based cryptosystem depends on the difficulty of several computational problems, some of them exclusive to the area, such as the pairing inversion problem (see Definition 1).

Using results of Verheul [19], later extended by Galbraith, Hess and Vercauteren [8], it is well known that if one can solve certain pairing-inversion problems, then it is also possible to solve the computational Diffie-Hellman (DH) problem in a number of groups, including a class of subgroups of finite fields.

In this article we find results that relate the difficulty of pairing inversion problems and the discrete logarithm problem (DLP). We begin using the techniques of Boheh and Lipton [2], and Maurer [12], to show that if one can solve both the FAPI_1 and FAPI_2 problems (see Definition 1), then there exist a subexponential discrete lograrithm algorithm in the groups involved.

We also explore the implications of being able to solve only one of the FAPI problems. In this case it is not possible to solve the computational Diffie-Hellman problem, so the previous approach does not apply. We prove that it is still possible to solve the *static Diffie-Hellman* problem, this will let use algorithms developed by Brown and Gallant [3], and Cheon [4] that solve the discrete logarithm problem using a static Diffie-Hellman oracle.

Instead of presenting his algorithm in the context of the static Diffie-Hellman problem, Cheon presents his algorithm as a solution to the **l-Strong Diffie-Hellman** problem (l-SDH).

Problem 1. Given P and $\alpha^i P$ for $i = 1 \dots l$, compute $\alpha^{l+1} P$.

This problem was first introduced by Boneh and Boyen in [1] to give a security proof in the standard model for a signature scheme. Cheon's idea consists of exploiting the extra information given by the SDH problem to accelerate the computation of the discrete logarithm α.

This article is organized as follows. In Section 2, we define the pairing inversion problems we are interested in. Section 3 uses the techniques developed to reduce the DLP to the DH problem to show that the existence of pairing inversion algorithms implies the existence of sub-exponential discrete logarithm algorithms. Section 4 presents Cheon's algorithm and explores its implications in the presence of a pairing inversion oracle. We present our conclusions in Section 5

2 Pairings

Throughout this article, we will let $\mathbf{G}_1, \mathbf{G}_2$ and $\mathbf{G}_T$ denote groups of prime order p. We will write the group operation in $\mathbf{G}_1$ and $\mathbf{G}_2$ additively, and we will use multiplicative notation for $\mathbf{G}_T$. We will consider non-degenerate bilinear pairings of the form

$$e : \mathbf{G}_1 \times \mathbf{G}_2 \longrightarrow \mathbf{G}_T.$$

We are interested in the following problems:

Definition 1. *Let e be a non-degenerate bilinear pairing as above.*
The **Fixed Argument Pairing Inversion 1 ($\mathbf{FAPI}_1$)** *problem is: given $P_1 \in \mathbf{G}_1, z \in \mathbf{G}_T$, find $P_2 \in G_2$ such that $e(P_1, P_2) = z$.*
The **Fixed Argument Pairing Inversion 2 ($\mathbf{FAPI}_2$)** *problem is: given $P_2 \in \mathbf{G}_2, z \in \mathbf{G}_T$, find $P_1 \in G_1$ such that $e(P_1, P_2) = z$.*

Given an instance (P_1, z) of the FAPI_1 problem, we will denote its solution as $P_2 = \mathrm{FAPI}_1(P_1, z)$. Analogously, a solution to an instance (P_2, z) of the FAPI_2 problem will be denoted as $P_1 = \mathrm{FAPI}_2(P_2, z)$.

The existence of efficient algorithms to solve the FAPI_1 and FAPI_2 problems would have profound consecuences. Galbraith, Hess and Vercauteren have generalized results of Verheul, and proved in [8] the following:

Theorem 1. *[Theorem 1 in [8]] Let $e : \mathbf{G}_1 \times \mathbf{G}_2 \longrightarrow \mathbf{G}_T$ be a non-degenerate bilinear pairing on cyclic groups of prime order p. Given access to $FAPI_1$ and $FAPI_2$ oracles, it is possible to solve the computational Diffie-Hellman problem in $\mathbf{G}_1, \mathbf{G}_2$ and $\mathbf{G}_T$ in polynomial time.*

In practice, $\mathbf{G}$ will be a subgroup of an elliptic curve $\mathbf{E}$ and e will be the Tate- or Weil-pairing (or a variant thereof). Let the elliptic curve $\mathbf{E}$ be defined over the field $\mathbf{K}$, and suppose that $\mathbf{K}$ contains the group of pth roots of unity μ_p. If $\mathbf{E}[p]$ denotes the p-torsion subgroup of $\mathbf{E}$, and $\mathbf{E}[p] \subset \mathbf{E}(\mathbf{K})$, the Tate-pairing is a non-degenerate bilinear function

$$\langle \cdot, \cdot \rangle : \mathbf{E}[p] \times \mathbf{E}(\mathbf{K})/p\mathbf{E}(\mathbf{K}) \longrightarrow \mathbf{K}^*/(\mathbf{K}^*)^p.$$

If $\mathbf{K}$ is a finite field, it is possible to get a unique element of $\mathbf{K}$ as result of the pairing as

$$e(P,Q) = \langle P,Q\rangle^{(\#\mathbf{K}/p)}.$$

This bilinear function is known as the *reduced Tate-pairing.*

Recent developments in pairing computation techniques, prominently the short Miller loops aforded by the *ate*-pairing [9], have raised questions regarding the possibility of solving one of the FAPI problems for some special curves. In the following sections we will explore some consequences of the existence of pairing inversion algorithms. A detailed description of the subtleties and difficulties regarding efficient pairing inversion can be found in [8, 17].

3 FAPI, the DH and DLP problems

After the publication of Verheul's results [19, 20] and with the results recently obtained by Galbraith, Hess and Vercauteren in [8], it is widely known that the ability to invert pairings in polynomial time implies that the computational Diffie-Hellman problem can also be solved in polynomial time.

Combining the results of den Boer [6], Boneh and Lipton [2], and Maurer and Wolf [12], which relate the Diffie-Hellman problem and the discrete logarithm problem, and the reduction from pairing inversion to the Diffie-Hellman problem proved in [19, 8], we will prove that pairing inversion can be used to solve the discrete logarithm problem in sub-exponential time in the order of the groups. These results, although well-known to experts in the field, have not been published and we include them here to provide a reference.

3.1 Black Box Fields

A black-box field is an abstract construction introduced in [2]. It is analogous to the extensively studied black-box groups construction.

Definition 2. *A black-box field is a* 5*-tuple* (p, S, h, F, G)*, where* p *is a prime and* S *is a set with* p *elements. The functions* h, F, G *are defined as follows:*

- *The function* $h : S \longrightarrow \mathbf{F}_p$ *is a bijection.*
- *The function* $F : S \times S \longrightarrow S$ *corresponds to addition, that is* $h(F(s_1, s_2)) = h(s_1) + h(s_2)$*.*
- *The function* $G : S \times S \longrightarrow S$ *corresponds to multiplication, that is* $h(G(s_1, s_2)) = h(s_1) \cdot h(s_2)$*.*

Following Boneh and Lipton, given x an element of $\mathbf{F}_p$, we will write $[x]$ to denote the element s of S such that $h(s) = x$. Note that the given functions suffice to compute field inversion, since $[x^{-1}] = [x^{p-2}]$. It is interesting to observe that there exist an algorithm by Shanks to extract square roots in $\mathbf{F}_p$ using only operations available in black-box fields [5]. This observation is fundamental in the techniques developed to relate the DH and DL problems.

Definition 3. *Let (p, S, h, F, G) be a black-box field for some prime p. Denote the map sending x to $[x]$ by $[\cdot]$. The black-box field problem is: given oracles for $F, G, [\cdot]$ and an element $[\alpha] \in S$, find α explicitly.*

The concept of a black-box field is important because being able to solve the computational Diffie-Hellman problem in a group $\mathbf{G}$ of order p, gives us a black box representation of $\mathbf{F}_p$ by elements of G.

Definition 4. *Given an instance (P, aP, bP) of the computational Diffie-Hellman problem, we denote its solution as $abP = \mathrm{DH}(P, aP, bP)$.*

Lemma 1. *Let $\mathbf{G}$ be group with prime order p generated by P. If we denote the group binary operation as $+$ and let*

$$\begin{aligned} h :&\mathbf{G} \longrightarrow \mathbf{F}_p \\ &aP \mapsto a, \end{aligned}$$

denote a bijection between $\mathbf{G}$ and $\mathbf{F}_p$, the 5-tuple $(p, \mathbf{G}, h, +, \mathrm{DH}(P, \cdot, \cdot))$ forms a black-box field representation of $\mathbf{F}_p$.

Proof. Since $(a + b)P = aP + bP$, it follows that $h(aP) + h(bP) = h((a + b)P)$. Analogously, since $abP = \mathrm{DH}(P, aP, bP)$, we have that $h(\mathrm{DH}(P, aP, bP)) = h(aP) \cdot h(bP)$.

Note that in this construction $[a] = aP$ for $a \in \mathbf{F}_p$. In this context, the DLP for the group $\mathbf{G}$ becomes the black-box field problem for $(p, \mathbf{G}, h, +, \mathrm{DH}(P, \cdot, \cdot))$.

The reduction from the DH problem to the DLP presented in [2, 13] uses the following idea of Maurer [12] to solve the DLP problem in the group $\mathbf{G}$ generated by P:

1. Find an elliptic curve $\mathbf{E}_{A,B}$, defined over $\mathbf{F}_p$ by the equation $y^2 = x^3 + Ax + B$, with N-smooth order for a suitably small N. Assume that $\mathbf{E}_{A,B}(\mathbf{F}_p)$ is generated by Q.
2. Given P and aP in $\mathbf{G}$, use the black-box representation of $\mathbf{F}_p$ on $\mathbf{G}$ afforded by the DH oracle and Lemma 1 to find $[y]$ such that $(a, y) \in \mathbf{E}_{A,B}$.
3. Since the order of $\mathbf{E}_{A,B}$ is N-smooth, use the Pohling-Hellman algorithm to find the discrete logarithm of (a, y) with respect to Q.
4. Recover a using the known coordinates of Q.

The elliptic curve $\mathbf{E}_{A,B}$ is known as an *auxiliary group*, and an approach using more general algebraic groups has been explored in [13].

A run of the algorithm to compute discrete logarithms using a Diffie-Hellman oracle thus consist of two parts: firstly, finding an appropriate curve $\mathbf{E}_{A,B}$, and secondly, computing the discrete logarithm of (a, y) with respect to Q. The best result in this direction was proven by Boneh and Lipton in [2], and is presented here as Theorem 2.

Maurer [12] argues that with high probability there is a number in the interval $[p+1-2\sqrt{p}, p+1+2\sqrt{p}]$ whose largest prime factor is polynomial in $\log p$. Since

for every integer n in this interval there is an elliptic curve over $\mathbf{F}_p$ with n points [7], knowing the equation defining such an elliptic curve would provide a polynomial time algorithm to solve discrete logarithms in groups of order p with access to a DH-oracle. The implications of these result are not clear, since finding n (and hence the elliptic curve) is likely to be exponentially hard.

Incidentally, Muzereau, Smart and Vercauteren have found auxiliary groups with N-smooth order ($2^{20} \leq N \leq 2^{83}$), for most of the NIST elliptic curves [15]. This is, of course, a hardness result for the Diffie-Hellman problem, as there is no reason to expect the DL problem in this curves to be easy.

3.2 Black-Box fields and Pairing Inversion

After the costruction described in the previous subsection, a pairing inversion algorithm could then be used as a DH-oracle in the reduction of Boneh and Lipton to solve discrete logarithms in the p-torsion subgroup of an elliptic curve and the group of p-roots of unity μ_p in $\mathbf{K}$. This proves that (conditional to some conjectures regarding the number of N-smooth numbers in Hasse-Weil intervals) there is a reduction from the discrete logarithm problem to solving both of FAPI_1 and FAPI_2.

Definition 5. *Given a natural number n and a real number α, such that $0 \leq \alpha \leq 1$, denote*

$$L_n(\alpha) = \exp((\log n)^{\alpha}(\log\log n)^{1-\alpha}).$$

Conjecture 1. [Conjecture 2.10 in [11]] A random interger in the interval $(p+1-2\sqrt{p}, p+1+2\sqrt{p})$ is $L_p(\alpha)$ smooth with probability at least $1/L_p(1-\alpha)^{1-\alpha+o(1)}$ for any α.

Assuming Conjecture 1, Boneh and Lipton prove the following:

Theorem 2. *Given a group $\mathbf{G}$ of prime order p, and access to a DH oracle for $\mathbf{G}$, it is possible to compute discrete logarithms in $\mathbf{G}$ in time $L_p(1/2)^{2+o(1)}$.*

Using Theorem 2, we are ready to prove the main result of this section.

Theorem 3. *Consider $e : \mathbf{G}_1 \times \mathbf{G}_2 \longrightarrow \mathbf{G}_T$ a non-degenerate bilinear pairing. Given access to $FAPI_1$ and $FAPI_2$ oracles, it is possible to solve the DLP in $\mathbf{G}_1, \mathbf{G}_2$ and $\mathbf{G}_T$ in time $L_p(1/2)^{2+o(1)}$*

Proof. Using Theorem 1, the FAPI oracles can be used to construct a Diffie-Hellman oracle for all the groups involved. The result follows immediately from Theorem 2. □

This theorem proves that the existence of algorithms that efficiently solve the FAPI_1 and FAPI_2 problems implies the existence of sub-exponential DLP algorithms for the groups involved. However, the Quadratic Sieve and the Number Field Sieve already provide sub-exponential DLP algorithms in finite fields, and using the MOV [14] attack, we get a sub-exponential DLP algorithm for elliptic

curves with sufficiently small embedding degree. In this respect, Theorem 3 is hardly a surprising result.

Furthermore, for a fixed embedding degree k, computing discrete logarithms using our reduction would be slower than a direct attack using the Number Field Sieve on the embedding field $\mathbf{F}_{p^k}$, where discrete logarithms can be computed in time $L_{p^k}(1/3)$. It would be very interesting to find algorithms that accelerate the computation of discrete logarithms using a DH oracle in groups that already have a sub-exponential discrete logarithm algorithm, such as the group of invertible elements in a finite field.

4 Cheon's algorithm and the DLP

In the previous section we proved that being able to solve the FAPI problems allows for the computation of discrete logarithms in sub-exponential time. Note that it might be possible to have algorithms that solve only one of the FAPI problems. In that case, the techniques of Boneh, Lipton, Maurer and Wolf can not be used.

We will prove that one might still adapt an approach developed by Brown and Gallant [3], and Cheon [4], to work in this setting. As we mentioned before, Cheon developed an algorithm to solve the DLP in the context of the l-SDH problem. Using oracle calls to just one FAPI problem, we can use Cheon's algorithm to compute discrete logarithms.

4.1 Static Diffie-Hellman Problem

In [3], Brown and Gallant introduce the concept of the *static Diffie-Hellman* (ScDH) problem.

Definition 6. *Given fixed elements P and aP of the group $\mathbf{G}$, and a random element yP, find ayP.*

Given an instance $((P, aP), Q)$ of the ScDH problem, we will denote its solution as $aQ = \mathrm{ScDH}_{(P,aP)}(Q)$.

The interest in this problem comes from the fact that in many protocols, including static Diffie-Hellman key agreement, a user has a fixed public key aP, and attacks to the system would involve solving an instance of the Diffe-Hellman problem with one of the entries equal to aP. The security of the system from the user's perspective thus depends on the difficulty of solving the ScDH problem and not the traditional DH problem.

Brown and Gallant prove that for a group $\mathbf{G}$ with prime order p, if $p-1 = uv$, and one is given access to a ScDH oracle, it is possible to solve the DLP in $\mathbf{G}$ in time $O(\sqrt{u}+\sqrt{v})$. In the next subsection we present an algorithm due to Cheon that is similar to the algorithm presented by Brown and Gallant proving their result.

Our interest in the static Diffie-Hellman problem arises from the fact that having access to an oracle that solves exactly one of the FAPI_1 or FAPI_2 problems provides us with a ScDH oracle when interpreted in the appropriate groups.

Proposition 1. *Given a bilinear pairing $e : \mathbf{G}_1 \times \mathbf{G}_2 \longrightarrow \mathbf{G}_T$, elements $P_1 \in \mathbf{G}_1$ $P_2, \alpha P_2 \in \mathbf{G}_2$ and access to a $FAPI_2$ oracle, it is possible to solve the ScDH problem in $\mathbf{G}_T$ with static input (z, z^α), where $z = e(P_1, P_2)$, in polynomial time.*

Proof. Let $z = e(P_1, P_2)$. Then $z^\alpha = (P_1, \alpha P_2)$. Given z^y, we can use the FAPI_2 oracle to find $yP_1 = FAPI_2(P_2, z^y)$. We finish simply computing $z^{y\alpha} = e(yP_1, \alpha P_2)$. □

4.2 Cheon's algorithm

We now explore Cheon's algorithm [4] and analyse how can it be combined with a FAPI oracle to solve the DLP. We decided to present Cheon's algorithm instead of the very similar solution presented by Brown and Gallant [3], since Cheon's work allows for the computation of discrete logarithms using the factors of either $p-1$ or $p+1$, and is more general from our perspective.

Theorem 4. *[Theorem 1 in [4]] Let P be an element of prime order p in an abelian group. Suppose that d is a positive divisor of $p-1$. If $P, P_1 = \alpha P$ and $P_d = \alpha^d P$ are given, α can be computed in $O(\log p(\sqrt{(p-1)/d} + \sqrt{d}))$ group operations using $O(\max\{\sqrt{(p-1)/d}, \sqrt{d}\})$ memory.*

To prove Theorem 4 it suffices to show that Algorithm 1 is correct and finishes in the indicated time. The running time of $O(\log p(\sqrt{p/d}+\sqrt{d}))$ for Algorithm 1 was later improved to $O(\sqrt{p/d} + \sqrt{d})$ by Kozaki et al in [10].

Algorithm 1 Cheon's Algorithm

INPUT: A tuple $(P, P_1 = \alpha P, P_d = \alpha^d P)$, where $d|p-1$.
OUTPUT: α

1: Find a generator $\zeta_0 \in \mathbf{F}_p^*$.
2: $\zeta := \zeta_0^d$.
3: $d_1 := \lceil \sqrt{(p-1)/d} \rceil$.
4: Find $0 \leq u_1, v_1 < d_1$ such that $\zeta^{-u_1} P_d = \zeta^{d_1 v_1} P$ by BSGS.
5: $k_0 := d_1 v_1 + u_1$. Note $\alpha^d = \zeta^{k_0}$.
6: $d_2 := \lceil \sqrt{d} \rceil$.
7: Find $0 \leq u_2, v_2 < d_2$ such that $\zeta_0^{-u_2(p-1)/d} P_1 = \zeta_0^{k_0 + d_2 v_2 (p-1)/d} P$ by BSGS.
8: **return** $\zeta_0^{k_0 + (d_2 v_2 + u_2)(p-1)/d}$

Cheon presents several other algorithms to find the discrete logarithm of an element using extra information presented in some cryptographic schemes. Using a special implementation of Algorithm 1, Cheon proves the following:

Corollary 1. *Let P be an element of prime order p in an abelian group. Suppose that $p-1 = \prod_{i=1}^{t} d_i$, for d_i pairwise relatively prime. If P and $P_i = \alpha^{(p-1)/d_i}$ for $1 \leq i \leq t$ are given, then α can be computed in $O\left(\log p\left(\sqrt{\sum_{i=1}^{t} d_i}\right)\right)$ group operations using $\max\{\sqrt{d_i}\}_{1 \leq i \leq t}$ memory.*

Finally, Cheon represents elements of $\mathbb{F}_{p^2}$ as pairs of elements of $\mathbf{F}_p$, and uses a clever representation of elements in the subgroup μ_{p+1} of $\mathbb{F}_{p^2}^*$ to prove:

Theorem 5 (Theorem 2 in [4]). *Let P be an element of prime order p in an abelian group. Suppose that d is a positive divisor of $p+1$ and $P_i = \alpha^i P$ for $i = 1, 2, \ldots 2d$ are given. Then α can be computed in $O(\log p(\sqrt{(p+1)/d} + d))$ group operations using $O(\max\{\sqrt{(p+1)/d}, \sqrt{d}\})$ memory.*

4.3 Cheon's algorithm and FAPI

Algorithm 1 was presented in the context of problems similar to the l-SDH problem as discussed above. However, there is another area where they can potentially be used to attack a cryptosystem.

Proposition 2. *Given $P_1 \in \mathbf{G}_1$, $P_2, \alpha P_2 \in \mathbf{G}_2$, and access to a FAPI$_2$ oracle, then it is possible to compute $\alpha^i P_1$ for every i using $O(i)$ calls to the FAPI$_2$ oracle.*

Proof. The proof follows from a simple induction argument. Having found $\alpha^n P_1$, we can compute

$$\alpha^{n+1} P_1 = FAPI_2(P_2, e(\alpha^n P_1, P_2).$$

An analogous computation recovers $\alpha^{n-1} P_1$. □

Using Cheon's algorithm and the previous Proposition, we get the following:

Theorem 6. *Consider $e : \mathbf{G}_1 \times \mathbf{G}_2 \longrightarrow \mathbf{G}_T$ a non-degenerate bilinear pairing between groups of prime order p. Given $P_2, \alpha P_2 \in \mathbf{G}_2$, access to a FAPI$_2$ oracle and a positive integer d dividing $p-1$ or $p+1$, there exists an algorithm that computes α in time $O(\sqrt{p/d} + d)$.*

Proof. Use d calls to the FAPI$_2$ oracle to compute $\alpha^d g$ as described in Proposition 2. Given $P, \alpha P$ and $\alpha^d P$ we can use the algorithm in [10] to recover α running in time $O(\sqrt{p/d} + \sqrt{d})$. The result follows. □

Using heuristic results describing the divisors of $p+1$ and $p-1$, we can give an effective version of Theorem 6. If we assume that for a prime number p, the prime decomposition of $p+1$ and $p-1$ is the same as that of a random integer, we get the following.

Conjecture 2 (Section 3 of [3]). The largest prime factor of $p-1$ and $p+1$ is typically of size $O(p^{2/3})$.

Combining Theorem 6 with Conjecture 2, we can prove:

Corollary 2. *Under the hypotheses of Theorem 6, if p is a random prime, with high probability there exists an algorithm to find α in time $O(p^{1/3})$.*

Proof. If either of $p+1$ or $p-1$ has a prime factor of size $O(p^{2/3})$, then it also has a divisor of size $O(p^{1/3})$. Using this divisor as d in Theorem 6 gives a running time of $O(p^{1/3})$. □

Note that Pollard's rho method [16] has a running time of $O(n^{1/2})$ to compute discrete logarithms. If we are in a situation where Pollard's rho in $\mathbf{G}_1$ is balanced with the cost of the Number Field Sieve in $\mathbf{G}_T$, Theorem 6 and Corollary 2 provide an actual speed-up in the computation of discrete logarithms.

5 Conclusions

The relation between pairing inversion algorithms and other well-studied computational problems has only recently received widespread attention [8, 17]. In many pairing-based cryptosystems, the groups $\mathbf{G}_1$ and $\mathbf{G}_2$ are the same, or there is an efficiently computable morphism between them. In this cases, Theorem 3 proves that if the DLP is hard, no efficient pairing inversion algorithm exists. The same can be argued from Theorem 6, although in this case the reduction is much looser. As mentioned, if the embedding degree is fixed, the MOV attack [14] already provides a faster sub-exponential reduction. For some values of k, the MOV attack becomes exponentially slow while pairings can still be computed in polynomial time.

The families of pairing friendly elliptic curves for which the authors of [8] prove that the Miller inversion problem can be solved in polynomial time have embedding degree

$$k \approx \alpha \left(\frac{\log r}{\log \log r} \right),$$

so *if* one could invert pairings for this families, the reduction given by Theorem 2 would be asymptotically faster than that provided by the MOV attack. Note that the curves in this family are ordinary elliptic curves, so there is no obvious non-trivial morphism between $\mathbf{G}_1$ and $\mathbf{G}_2$.

From a practical point of view, if a sub-exponential but expensive pairing inversion algorithm existed, Theorem 6 might provide a faster tool to compute discrete logarithms even in cases where an efficiently computable map $\Psi : \mathbf{G}_1 \longrightarrow \mathbf{G}_2$ exists. This is because the algorithms described in Section 3 use significantly more calls to a DH oracle in order to compute discrete logarithms than those based in Cheon's algorithm.

If an efficiently computable isomorphim between $\mathbf{G}_1$ and $\mathbf{G}_2$ is known, it is possible to find $\alpha^d P$ using only $O(\log d)$ applications of the FAPI_2 algorithm combining Proposition 2 and a square-and-multiply algorithm; this would allow us to compute discrete logarithms in $O(\sqrt{p/d}+\sqrt{d}+C \log p)$ operations, where C is the cost of a run of the FAPI_2 algorithm. For example, if either of $p+1$ or $p-1$ has a divisor of size $O(p^{1/2})$, discrete logarithms can be found in $O(p^{1/4}+C \log p)$ operations. Depending on the value of C, this could provide a great speed-up in the computation of discrete logarithms.

Our results show that the existence of efficient algorithms to solve the FAPI problems would accelerate the computation of discrete logarithms on some elliptic curves. Depending on the parameters being used, our reduction from the FAPI problems to the DLP might be faster than the reduction given by the MOV attack on pairing frieldy elliptic curves.

References

1. Boneh, D., and Boyen, X. Short signatures without random oracles. In *EUROCRYPT* (2004), C. Cachin and J. Camenisch, Eds., vol. 3027 of *Lecture Notes in Computer Science*, Springer, pp. 56–73.
2. Boneh, D., and Lipton, R. J. Algorithms for black-box fields and their application to cryptography (extended abstract). In *CRYPTO* (1996), N. Koblitz, Ed., vol. 1109 of *Lecture Notes in Computer Science*, Springer, pp. 283–297.
3. Brown, D. R. L., and Gallant, R. P. The static diffie-hellman problem. Cryptology ePrint Archive, Report 2004/306, 2004. http://eprint.iacr.org/.
4. Cheon, J. H. Security analysis of the strong diffie-hellman problem. In *EUROCRYPT* (2006), S. Vaudenay, Ed., vol. 4004 of *Lecture Notes in Computer Science*, Springer, pp. 1–11.
5. Cohen, H. *A course in computational algebraic number theory*, vol. 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993.
6. den Boer, B. Diffie-hillman is as strong as discrete log for certain primes. In *CRYPTO* (1988), S. Goldwasser, Ed., vol. 403 of *Lecture Notes in Computer Science*, Springer, pp. 530–539.
7. Deuring, M. Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. *Abh. Math. Sem. Hansischen Univ. 14* (1941), 197–272.
8. Galbraith, S. D., Hess, F., and Vercauteren, F. Aspects of pairing inversion. Cryptology ePrint Archive, Report 2007/256, 2007. http://eprint.iacr.org/.
9. Hess, F., Smart, N. P., and Vercauteren, F. The eta pairing revisited. *IEEE Transactions on Information Theory 52*, 10 (2006), 4595–4602.
10. Kozaki, S., Kutsuma, T., and Matsuo, K. Remarks on cheon's algorithms for pairing-related problems. In Takagi et al. [18], pp. 302–316.
11. Lenstra, Jr., H. W. Factoring integers with elliptic curves. *Ann. of Math. (2) 126*, 3 (1987), 649–673.
12. Maurer, U. M. Towards the equivalence of breaking the diffie-hellman protocol and computing discrete algorithms. In *CRYPTO* (1994), Y. Desmedt, Ed., vol. 839 of *Lecture Notes in Computer Science*, Springer, pp. 271–281.
13. Maurer, U. M., and Wolf, S. The diffie-hellman protocol. *Des. Codes Cryptography 19*, 2/3 (2000), 147–171.
14. Menezes, A., Vanstone, S. A., and Okamoto, T. Reducing elliptic curve logarithms to logarithms in a finite field. In *STOC* (1991), ACM, pp. 80–89.
15. Muzereau, A., Smart, N. P., and Vercauteren, F. The equivalence between the DHP and DLP for elliptic curves used in practical applications. *LMS J. Comput. Math. 7* (2004), 50–72 (electronic).
16. Pollard, J. M. Monte Carlo methods for index computation (mod p). *Math. Comp. 32*, 143 (1978), 918–924.
17. Satoh, T. On pairing inversion problems. In Takagi et al. [18], pp. 317–328.

18. Takagi, T., Okamoto, T., Okamoto, E., and Okamoto, T., Eds. *Pairing-Based Cryptography - Pairing 2007, First International Conference, Tokyo, Japan, July 2-4, 2007, Proceedings* (2007), vol. 4575 of *Lecture Notes in Computer Science*, Springer.
19. Verheul, E. R. Evidence that xtr is more secure than supersingular elliptic curve cryptosystems. In *EUROCRYPT* (2001), B. Pfitzmann, Ed., vol. 2045 of *Lecture Notes in Computer Science*, Springer, pp. 195–210.
20. Verheul, E. R. Evidence that xtr is more secure than supersingular elliptic curve cryptosystems. *J. Cryptology 17*, 4 (2004), 277–296.