GENERALIZATIONS OF VERHEUL'S THEOREM TO ASYMMETRIC PAIRINGS

KORAY KARABINA, EDWARD KNAPP, AND ALFRED MENEZES

ABSTRACT. For symmetric pairings $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, Verheul proved that the existence of an efficiently-computable isomorphism $\phi : \mathbb{G}_T \to \mathbb{G}$ implies that the Diffie-Hellman problems in $\mathbb{G}$ and $\mathbb{G}_T$ can be efficiently solved. In this paper, we explore the implications of the existence of efficiently-computable isomorphisms $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ and $\phi_2 : \mathbb{G}_T \to \mathbb{G}_2$ for asymmetric pairings $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. We also give a simplified proof of Verheul's theorem.

1. INTRODUCTION

Let r be a prime number, let $\mathbb{G}$ be an additively-written group of order r, and let $\mathbb{G}_T$ be a multiplicatively-written group of order r. A *symmetric pairing* (also called a *Type 1* pairing) on $(\mathbb{G}, \mathbb{G}_T)$ is an efficiently-computable non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$; see [9]. In the cryptographic literature, such pairings are generally constructed using the Weil and Tate pairings where $\mathbb{G}$ is a subgroup of the $\mathbb{F}_q$-rational points on a supersingular elliptic curve defined over a finite field $\mathbb{F}_q$, $\mathbb{G}_T$ is the order-r subgroup of $\mathbb{F}_{q^k}^*$, and where k is the smallest positive integer for which $r \mid (q^k - 1)$ (called the embedding degree of $\mathbb{G}$).

The discrete logarithm problem in $\mathbb{G}$, denoted $\mathrm{DL}_{\mathbb{G}}$, is the following: given a generator $P \in \mathbb{G}$ and a second point $Q \in \mathbb{G}$, find the integer ℓ modulo r such that $Q = \ell P$. Similarly, the discrete logarithm problem in $\mathbb{G}_T$, denoted $\mathrm{DL}_{\mathbb{G}_T}$, is the following: given a generator $g \in \mathbb{G}_T$ and a second element $h \in \mathbb{G}_T$, find the integer ℓ modulo r such that $h = g^\ell$. As observed in [8, 14], for any fixed generator $P \in \mathbb{G}$ the map $\xi : \mathbb{G} \to \mathbb{G}_T$ defined by $\xi : Q \mapsto e(P, Q)$ is an efficiently-computable[1] isomorphism from $\mathbb{G}$ to $\mathbb{G}_T$. Thus, $\mathrm{DL}_{\mathbb{G}}$ can be efficiently reduced to $\mathrm{DL}_{\mathbb{G}_T}$. These so-called Weil and Tate pairing attacks on $\mathrm{DL}_{\mathbb{G}}$ had negative consequences for the security of discrete-log cryptographic schemes in $\mathbb{G}$, because at the time they were discovered subexponential-time algorithms were known for solving $\mathrm{DL}_{\mathbb{G}_T}$ [6, 7] whereas no subexponential-time algorithms for $\mathrm{DL}_{\mathbb{G}}$ were known.

1991 *Mathematics Subject Classification.* 94A60.

Key words and phrases. Verheul's theorem, asymmetric pairings, cryptography, discrete logarithm problem.

[1] In the remainder of the paper, "efficient" is understood to mean "polynomial-time". When we say "algorithm" and "reduction", we mean "efficient algorithm" and "efficient reduction".

At CRYPTO 2000, Lenstra and Verheul [13] presented XTR, a discrete-log public-key cryptosystem which operates in an order-r subgroup X of the order-(p^2-p+1) cyclotomic subgroup of $\mathbb{F}_{p^6}^*$; here p is a prime. XTR was claimed to be as efficient as elliptic curve cryptography, but without being affected by the uncertainty that was still marring security of elliptic curve cryptography at the time. At the Rump Session of CRYPTO 2000, Menezes and Vanstone [15] observed that there is a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$ of embedding degree 3 with $\#E(\mathbb{F}_{p^2}) = p^2-p+1$. Thus, the Weil and Tate pairings yield an efficiently-computable isomorphism $\xi : \mathbb{G} \to X$, where $\mathbb{G}$ is the order-r subgroup of $E(\mathbb{F}_{p^2})$. They asked about the existence of an efficiently-computable isomorphism $\phi : X \to \mathbb{G}$, which would establish the equivalence of $\mathrm{DL}_{\mathbb{G}}$ and DL_X. However, Verheul [18] gave some evidence that such an isomorphism ϕ was unlikely to exist. More generally, Verheul proved that if there exists an efficiently-computable isomorphism $\phi : \mathbb{G}_T \to \mathbb{G}$, where $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a symmetric pairing, then $\mathrm{DH}_{\mathbb{G}}$ and $\mathrm{DH}_{\mathbb{G}_T}$ can be efficiently solved. Here, $\mathrm{DH}_{\mathbb{G}}$ is the Diffie-Hellman problem in $\mathbb{G}$: given $P, aP, bP \in \mathbb{G}$, compute abP; $\mathrm{DH}_{\mathbb{G}_T}$ is analogously defined. Verheul's theorem is striking because the only method known for solving the Diffie-Hellman problem in a group is to first solve the discrete logarithm problem in that group.

For order-r groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ with $\mathbb{G}_1 \neq \mathbb{G}_2$, an *asymmetric pairing* on $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ is an efficiently-computable non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$; see [9]. Such pairings can be constructed using the Weil and Tate pairings where $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups of the group of r-torsion points on an ordinary elliptic curve defined over $\mathbb{F}_q$, and where $\mathbb{G}_T$ is the order-r subgroup of $\mathbb{F}_{q^k}^*$. Two kinds of asymmetric pairings have been considered in the cryptographic literature — *Type 2* pairings where an efficiently-computable isomorphism $\psi_1 : \mathbb{G}_1 \to \mathbb{G}_2$ is known but no such isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ is known, and *Type 3* pairings where efficiently-computable isomorphisms from $\mathbb{G}_1$ to $\mathbb{G}_2$ and from $\mathbb{G}_2$ to $\mathbb{G}_1$ are not known [11].

In [10], Galbraith, Hess and Vercauteren studied the implications of the existence of efficient algorithms for certain pairing inversion problems: (i) given $R \in \mathbb{G}_1$ and $z \in \mathbb{G}_T$, find $S \in \mathbb{G}_2$ with $e(R,S) = z$; and (ii) given $S \in \mathbb{G}_2$ and $z \in \mathbb{G}_T$, find $R \in \mathbb{G}_1$ with $e(R,S) = z$. The existence of such algorithms would have devastating consequence for the security of pairing-based cryptosystem. The results in [10] are purported to be generalizations and refinements of Verheul's theorem to asymmetric pairings. However, a strict generalization of Verheul's theorem would be concerned with efficiently-computable isomorphisms from $\mathbb{G}_T$ to $\mathbb{G}_1$ and from $\mathbb{G}_T$ to $\mathbb{G}_2$. In §3, we explore the implications of the existence of such isomorphisms. We begin in §2 with a simplified proof of Verheul's theorem.

2. Symmetric pairings

Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a symmetric pairing. An isomorphism $\phi : \mathbb{G}_T \to \mathbb{G}$ is defined by its action on some generator $g \in \mathbb{G}$; say $\phi(g) = P$. The *Compute-ϕ* problem is the following: given $z \in \mathbb{G}_T$, compute $\phi(z)$. The *Fixed Argument*

Pairing Inversion (FAPI) problem is the following: given $R \in \mathbb{G}$ and $z \in \mathbb{G}_T$, determine $S \in \mathbb{G}$ such that $e(R, S) = z$.

Lemma 1. *The FAPI and Compute-ϕ problems are computationally equivalent.*

Proof. The proof that Compute-ϕ reduces to FAPI is straightforward. Suppose that we are given $z = g^{\ell}$ and a FAPI oracle; we wish to compute $\phi(z) = \ell P$. We first use the FAPI oracle to find $Q \in \mathbb{G}$ such that $e(P, Q) = g$, and then use the FAPI oracle to find $R \in \mathbb{G}$ for which $e(Q, R) = z$; note that $R = \ell P$.

We now prove that FAPI reduces to Compute-ϕ. So, given $R \in \mathbb{G}$, $z \in \mathbb{G}_T$, and an oracle for Compute-ϕ, we wish to compute $S \in \mathbb{G}$ with $e(R, S) = z$. Let $R = aP$, $S = bP$, $e(P, P) = g^c$, and $z = g^t$. Since $e(aP, bP) = g^t$, we have $bP = a^{-1}c^{-1}tP$. Now, define $T(i) = a^i c^{i-1} P$ for $i \geq 1$, and note that $T(1) = R$. Given $T(i)$, one can efficiently compute $T(2i)$ since

$$e(T(i), T(i)) = e(a^i c^{i-1} P, a^i c^{i-1} P) = g^{a^{2i} c^{2i-1}}$$

and $\phi(g^{a^{2i} c^{2i-1}}) = a^{2i} c^{2i-1} P = T(2i)$. Moreover, given $T(2i)$, one can efficiently compute $T(2i + 1)$ since

$$e(T(2i), T(1)) = e(a^{2i} c^{2i-1} P, aP) = g^{a^{2i+1} c^{2i}}$$

and $\phi(g^{a^{2i+1} c^{2i}}) = a^{2i+1} c^{2i} P = T(2i + 1)$. Thus, by processing the bits of the binary representation of $r - 2$ from left to right, one can use a double-and-add strategy to efficiently compute

$$T(r - 2) = a^{r-2} c^{r-3} P = a^{-1} c^{-2} P.$$

Finally, one can efficiently compute $\phi(g^t) = tP$, $e(tP, a^{-1}c^{-2}P) = g^{a^{-1}c^{-1}t}$ and $\phi(g^{a^{-1}c^{-1}t}) = a^{-1}c^{-1}tP = S$. □

Lemma 1 immediately gives a short proof of Verheul's Theorem.

Theorem 1 (Verheul)**.** *Let $\phi : \mathbb{G}_T \to \mathbb{G}$ be an isomorphism defined by $\phi(g) = P$ and suppose that ϕ can be efficiently computed. Then $DH_{\mathbb{G}}$ and $DH_{\mathbb{G}_T}$ can be efficiently solved.*

Proof. By Lemma 1, we can efficiently solve the FAPI problem.

Suppose that we are given a $\mathrm{DH}_{\mathbb{G}}$ instance (Q, aQ, bQ) and wish to compute abQ. We compute $z = e(aQ, bQ)$ and then find $R \in \mathbb{G}$ such that $e(Q, R) = z$; note that $R = abQ$.

Suppose now that we are given a $\mathrm{DH}_{\mathbb{G}_T}$ instance (h, h^x, h^y) and wish to compute h^{xy}. We do the following:

(i) Find $Q \in \mathbb{G}$ with $e(P, Q) = h$.
(ii) Find $R \in \mathbb{G}$ with $e(Q, R) = h^x$; note that $R = xP$.
(iii) Find $S \in \mathbb{G}$ with $e(P, S) = h^y$; note that $S = yQ$.
(iv) Compute $e(R, S) = e(xP, yQ) = h^{xy}$.

□

As mentioned in §1, the point of Verheul's theorem is to argue that $\mathrm{DL}_{\mathbb{G}_T}$ cannot be reduced to $\mathrm{DL}_{\mathbb{G}}$, thus providing evidence that $\mathrm{DL}_{\mathbb{G}_T}$ is harder than $\mathrm{DL}_{\mathbb{G}}$. However, as observed in [12], Verheul's theorem can also be viewed as having negative consequences for pairing-based cryptography wherein someone who mistrusts the security of elliptic curve cryptosystems may have their worries allayed by the assurance that $\mathrm{DL}_{\mathbb{G}}$ is no easier than $\mathrm{DL}_{\mathbb{G}_T}$.

A more relevant question is whether there exists a *reduction* of $\mathrm{DL}_{\mathbb{G}_T}$ to $\mathrm{DL}_{\mathbb{G}}$. Such a reduction is an algorithm $\mathcal{R}$ which on input $h, h^x \in \mathbb{G}_T$ and an oracle for solving $\mathrm{DL}_{\mathbb{G}}$, computes x. Clearly, an algorithm for computing some isomorphism $\phi : \mathbb{G}_T \to \mathbb{G}$ is also a reduction of $\mathrm{DL}_{\mathbb{G}_T}$ to $\mathrm{DL}_{\mathbb{G}}$. The interesting question is whether *every* reduction of $\mathrm{DL}_{\mathbb{G}_T}$ to $\mathrm{DL}_{\mathbb{G}}$ in fact yields an efficiently-computable isomorphism from $\mathbb{G}_T$ to $\mathbb{G}$. By Lemma 1, an equivalent question is whether every reduction of $\mathrm{DL}_{\mathbb{G}_T}$ to $\mathrm{DL}_{\mathbb{G}}$ yields an efficient FAPI solver:

Question 1. Is there an algorithm $\mathcal{A}$ which, when given oracle access to a reduction algorithm $\mathcal{R}$ and inputs $z \in \mathbb{G}_T$, $R \in \mathbb{G}$, outputs $S \in \mathbb{G}$ such that $e(R, S) = z$?

Suppose there is an efficient algorithm $\mathcal{A}$ which solves the problem posed in Question 1 when given *black-box* access to $\mathcal{R}$[2]. Then $\mathcal{A}$ can be used to efficiently solve the following mixed FAPI-$\mathrm{DL}_{\mathbb{G}}$ problem: given $z \in \mathbb{G}_T$, $R \in \mathbb{G}$, $U \in \mathbb{G}$, $xU \in \mathbb{G}$, compute x or $S \in \mathbb{G}$ with $e(R, S) = z$. Namely, given a FAPI-$\mathrm{DL}_{\mathbb{G}}$ problem instance (z, R, U, xU), we invoke algorithm $\mathcal{A}$ with inputs (z, R). If $\mathcal{A}$ does not make any calls to its oracle $\mathcal{R}$, then $\mathcal{A}$ outputs S which solves the FAPI-$\mathrm{DL}_{\mathbb{G}}$ instance. If $\mathcal{A}$ makes a call to $\mathcal{R}$, then we (in our role as simulator for $\mathcal{R}$), request the solution of the $\mathrm{DL}_{\mathbb{G}}$ instance (U, xU); since $\mathcal{A}$ is responsible for answering $\mathcal{R}$'s oracle queries, $\mathcal{A}$ returns x which again solves the FAPI-$\mathrm{DL}_{\mathbb{G}}$ instance.

Since FAPI-$\mathrm{DL}_{\mathbb{G}}$ is expected to be intractable, the above argument suggests that reductions of $\mathrm{DL}_{\mathbb{G}_T}$ to $\mathrm{DL}_{\mathbb{G}}$ are in fact *more general* than efficiently-computable isomorphisms from $\mathbb{G}_T$ to $\mathbb{G}$. Hence, Verheul's theorem can be viewed as providing somewhat limited evidence that $\mathrm{DL}_{\mathbb{G}_T}$ is harder than $\mathrm{DL}_{\mathbb{G}}$ since it does not fully address the question of whether there is a (general) *reduction* from $\mathrm{DL}_{\mathbb{G}_T}$ to $\mathrm{DL}_{\mathbb{G}}$.

3. Asymmetric pairings

Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be an asymmetric pairing, where $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are groups of prime order r and $\mathbb{G}_1 \neq \mathbb{G}_2$. Furthermore, let g be a fixed generator of $\mathbb{G}_T$. An isomorphism $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ is defined by its action on g; say $\phi_1(g) = P_1$. Similarly, an isomorphism $\phi_2 : \mathbb{G}_T \to \mathbb{G}_2$ is defined by its action on g; say $\phi_2(g) = P_2$. The *Compute-ϕ_1* problem is the following: given $z \in \mathbb{G}_T$, compute $\phi_1(z)$. The *Compute-ϕ_2* problem is the following: given $z \in \mathbb{G}_T$, compute $\phi_2(z)$. The *FAPI-1* problem is the following: given $R \in \mathbb{G}_1$ and $z \in \mathbb{G}_T$, determine

[2]We have nothing useful to say in the case where $\mathcal{A}$ is given non-black-box access to $\mathcal{R}$.

$S \in \mathbb{G}_2$ with $e(R, S) = z$. Similarly, the *FAPI-2* problem is the following: given $S \in \mathbb{G}_2$ and $z \in \mathbb{G}_T$, determine $R \in \mathbb{G}_1$ with $e(R, S) = z$.

Galbraith, Hess and Vercauteren [10] proved the following:

Theorem 2. (i) *Suppose that FAPI-1 and FAPI-2 can both be efficiently solved. Then $DH_{\mathbb{G}_1}$, $DH_{\mathbb{G}_2}$ and $DH_{\mathbb{G}_T}$ can be efficiently solved.*
(ii) *Suppose that FAPI-1 can be efficiently solved, and suppose that we have an efficiently computable isomorphism $\psi_2 : \mathbb{G}_2 \to \mathbb{G}_1$. Then FAPI-2 can be efficiently solved.*
(iii) *Suppose that FAPI-2 can be efficiently solved, and suppose that we have an efficiently computable isomorphism $\psi_1 : \mathbb{G}_1 \to \mathbb{G}_2$. Then FAPI-1 can be efficiently solved.*

Suppose that FAPI-2 can be efficiently solved. Then one can easily construct an efficiently-computable isomorphism $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ — select arbitrary $S \in \mathbb{G}_2$ and $g \in \mathbb{G}_T$ and define ϕ_1 by $g \mapsto P$ where $e(P, S) = g$; then $\phi_1(z) = R$ where $e(R, S) = z$. However, it is not known whether an efficient FAPI-2 solver can be constructed from an efficiently-computable isomorphism $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$. The next result provides a partial answer to this question.

Theorem 3. (i) *Suppose that efficiently-computable isomorphisms $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ and $\psi_1 : \mathbb{G}_1 \to \mathbb{G}_2$ are known. Then FAPI-2 can be efficiently solved.*
(ii) *Suppose that efficiently-computable isomorphisms $\phi_2 : \mathbb{G}_T \to \mathbb{G}_2$ and $\psi_2 : \mathbb{G}_2 \to \mathbb{G}_1$ are known. Then FAPI-1 can be efficiently solved.*

Proof. We prove (i); the proof of (ii) is analogous.

Given $S \in \mathbb{G}_2$ and $z \in \mathbb{G}_T$, we wish to find $R \in \mathbb{G}_1$ with $e(R, S) = z$. Let $\phi_1(z) = aR$ and $\psi_1(R) = bS$ for some (unknown) integers a and b. Define the maps $\alpha : \mathbb{G}_1 \to \mathbb{G}_1$ and $\beta : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_1$ by $\alpha(U) = \phi_1(e(U, S))$ and $\beta(U, V) = \phi_1(e(U, \psi_1(V)))$. Observe that $\alpha(U) = aU$ and $\beta(U, V) = abcdR$ where $U = cR$ and $V = dR$, and that α and β can be efficiently computed.

For notational convenience, we identify a point $a^{i-1}b^{j-1}R$ with the vector $[i, j]$ whose components are integers modulo $r - 1$. Thus, $\alpha([i, j]) = [i + 1, j]$ and

$$\beta([i, j], [k, \ell]) = [i + k, j + \ell]. \tag{1}$$

Our goal is to efficiently compute $[1, 1]$, which corresponds to R. We begin by computing $\phi_1(z) = aR$ which corresponds to $[2, 1]$. Using (1), one can process the bits of the binary representation of $r - 2$ to efficiently compute $(r - 2) \cdot [2, 1] = [-2, -1]$, followed by $\alpha([-2, -1]) = [-1, 1]$. Finally, one uses (1) again to efficiently compute $(r - 2) \cdot [-1, -1] = [1, 1]$. □

The next result, which can be viewed as a refinement of Verheul's theorem for asymmetric pairings, follows immediately from Theorems 2 and 3.

Corollary 1. (i) *Suppose that efficiently-computable isomorphisms $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ and $\psi_1 : \mathbb{G}_1 \to \mathbb{G}_2$ are known. Then $DH_{\mathbb{G}_1}$, $DH_{\mathbb{G}_2}$ and $DH_{\mathbb{G}_T}$ can be efficiently solved.*

(ii) *Suppose that efficiently-computable isomorphisms* $\phi_2 : \mathbb{G}_T \to \mathbb{G}_2$ *and* $\psi_2 : \mathbb{G}_2 \to \mathbb{G}_1$ *are known. Then* $DH_{\mathbb{G}_1}$, $DH_{\mathbb{G}_2}$ *and* $DH_{\mathbb{G}_T}$ *can be efficiently solved.*

Theorem 4. *Suppose that efficiently-computable isomorphisms* $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ *and* $\phi_2 : \mathbb{G}_T \to \mathbb{G}_2$ *are known. Then FAPI-1 and FAPI-2 can be efficiently solved.*

Proof. We show that FAPI-2 can be efficiently solved. Given $S \in \mathbb{G}_2$ and $z \in \mathbb{G}_T$, we wish to find $R \in \mathbb{G}_1$ with $e(R, S) = z$. Let $\phi_1(z) = aR$ and $\phi_2(z) = bS$ for some (unknown) integers a and b. Define the maps $\alpha : \mathbb{G}_T \to \mathbb{G}_T$ and $\beta : \mathbb{G}_T \times \mathbb{G}_T \to \mathbb{G}_T$ by $\alpha(u) = e(\phi_1(u), S))$ and $\beta(u, v) = e(\phi_1(u), \phi_2(v))$. Observe that $\alpha(u) = u^a$ and $\beta(u, v) = z^{abcd}$ where $u = z^c$ and $v = z^d$, and that α and β can be efficiently computed.

For notational convenience, we identify an element $z^{a^{i-1}b^{j-1}}$ with the vector $[i, j]$ whose components are integers modulo $r - 1$. Thus, $\alpha([i, j]) = [i + 1, j]$ and $\beta([i, j], [k, \ell]) = [i + k, j + \ell]$. We are given the element z, which corresponds to $[1, 1]$, and our goal is to efficiently compute $R = \phi_1(z^{a^{-1}})$. We compute $(r - 2) \cdot [1, 1] = [-1, -1]$, followed by $\alpha([-1, -1]) = [0, -1]$. Finally, we compute $(r - 2) \cdot [0, -1] = [0, 1]$ which corresponds to $z^{a^{-1}}$ and $\phi_1(z^{a^{-1}})$. □

Theorems 2(i) and 4 immediately give the following generalization of Verheul's theorem for asymmetric pairings.

Corollary 2. *Suppose that efficiently-computable isomorphisms* $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ *and* $\phi_2 : \mathbb{G}_T \to \mathbb{G}_2$ *are known. Then* $DH_{\mathbb{G}_1}$, $DH_{\mathbb{G}_2}$ *and* $DH_{\mathbb{G}_T}$ *can be efficiently solved.*

Let $\mathbb{G}$ be an additively-written group of prime order r, and suppose that $d \mid r - 1$. Cheon [5] (see also [4]) showed how, given $P, xP, x^dP \in \mathbb{G}$, one can compute x in time

$$O\left(\log r \left(\sqrt{\frac{r}{d}} + d\right)\right) \tag{2}$$

and memory $O(\max\{\sqrt{r/d}, \sqrt{d}\})$. Note that if $d \approx r^{1/3}$, the running time and memory of Cheon's algorithm is $\tilde{O}(r^{1/3})$, which is faster than the running time $\tilde{O}(\sqrt{r})$ of Pollard's rho algorithm [17] for computing logarithms in $\mathbb{G}$. For convenience, we will refer to (2) as 'Cheon time'. Morales [16] showed that if FAPI-2 can be efficiently solved for an asymmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, then $\mathrm{DL}_{\mathbb{G}_2}$ can be solved in Cheon time with d calls to the FAPI-2 oracle.

We present extensions of Morales's result to the situation where an efficiently-computable isomorphism $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ is known. We let P, Q, g denoted fixed generators of $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ with $e(P, Q) = g$.

Theorem 5. *Suppose that an efficiently-computable isomorphism* $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ *is known. Then FAPI-2 can be solved in Cheon time.*

Proof. Let $S \in \mathbb{G}_2$ and $z \in \mathbb{G}_T$ be an instance of the FAPI-2 problem. Let $S = yQ$ and $z = g^{xy}$ for some $x, y \in [1, r - 1]$. Our goal is to compute $R = xP$. Let

$\phi(g) = aP$. Define $z_i = z^{(ay)^i}$ and $P_i = (ay)^i xP$ for $i \geq 0$. Since $z_0 = z$, $P_{i+1} = \phi_1(z_i)$, and $e(P_i, S) = z_i$ for $i \geq 0$, we can iteratively compute $z_1, z_2, \ldots, z_d$ in time $\tilde{O}(d)$. Now, using Cheon's algorithm with input z_i for $i = 0, 1, \ldots, d$, we can compute ay in Cheon time. Finally, we compute $R = (ay)^{-1}P_1 = xP$. □

Theorem 5 and Morales's result immediately show, given an efficiently-computable isomorphism $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$, that $\mathrm{DL}_{\mathbb{G}_2}$ can be solved in time $\tilde{O}(\sqrt{rd} + d^2)$. However, this result is not interesting since Pollard's rho algorithm already solves $\mathrm{DL}_{\mathbb{G}_2}$ in $\tilde{O}(\sqrt{r})$ time. Theorem 6 is a useful variant of this result.

Lemma 2. *Suppose that an efficiently-computable isomorphism $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ is known, and suppose that $\phi_1(g) = aP$. Then the integer a can be computed in Cheon time.*

Proof. Let $P_i = a^i P$ and $g_i = g^{a^i}$ for $i \geq 0$. Since $P_1 = aP$, $e(P_i, Q) = g_i$, and $\phi_1(g_i) = P_{i+1}$, we can iteratively compute $g_1, g_2, \ldots, g_d$ in Cheon time. Finally, Cheon's algorithm can be used to compute a. □

Theorem 6. (i) *Suppose that an efficiently-computable isomorphism $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ is known. Then $DL_{\mathbb{G}_2}$ can be solved in Cheon time.*

(ii) *Suppose that an efficiently-computable isomorphism $\phi_2 : \mathbb{G}_T \to \mathbb{G}_2$ is known. Then $DL_{\mathbb{G}_1}$ can be solved in Cheon time.*

Proof. We prove (i); the proof of (ii) is analogous.

Let $\phi_1(g) = aP$. Suppose that we are given a $\mathrm{DL}_{\mathbb{G}_2}$ instance (Q, S); we need to find the modulo-r integer y such that $S = yQ$. Let $P_i = (ay)^i aP$ and $g_i = g^{(ay)^i}$ for $i \geq 0$. Since $P_0 = aP$, $e(P_i, S) = g_{i+1}$, and $\phi_1(g_i) = P_i$, we can iteratively compute $g_1, g_2, \ldots, g_d$ in Cheon time. Next, Cheon's algorithm can be used to compute $ay \bmod r$. Finally, a can be computed in Cheon time using Lemma 2, and thereafter $y = a^{-1}(ay) \bmod r$ can be immediately computed. □

Remark 1. By Theorem 6, the existence of either an efficiently-computable isomorphism $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ or an efficiently-computable isomorphism $\phi_2 : \mathbb{G}_T \to \mathbb{G}_2$ will have damaging consequences to the security of pairing-based protocols. For example, in the Boneh-Lynn-Shacham signature scheme [3], an entity's private key is an integer x selected at random from the interval $[1, r-1]$, and the corresponding public key is $X = xQ$. The entity's signature on a message $m \in \{0,1\}^*$ is $\sigma = xM$, where $M = H(m)$ and $H : \{0,1\}^* \to \mathbb{G}_1$ is a hash function. The signed message (m, σ) can be verified by computing $M = H(m)$ and checking that $e(\sigma, Q) = e(M, X)$. If an efficiently-computable isomorphism $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$ is known, then the $\mathrm{DL}_{\mathbb{G}_2}$ instance (Q, X) can be solved in Cheon time to recover the private key x. If an efficiently-computable isomorphism $\phi_2 : \mathbb{G}_T \to \mathbb{G}_2$ is known then, given a single signed message (m, σ), the $\mathrm{DL}_{\mathbb{G}_1}$ instance (M, σ) can be solved in Cheon time to determine x.

Remark 2. Let $u = -(2^{62} + 2^{55} + 1)$ and consider the elliptic curve $E : Y^2 = X^3 + 2$ defined over $\mathbb{F}_p$, where $p = 36u^4 + 36u^3 + 24u^2 + 6u + 1$. This elliptic curve is in the Barreto-Naehrig [2] family of pairing-friendly curves with embedding

degree $k = 12$, and has the property that $r = \#E(\mathbb{F}_p) = 36u^4 + 36u^3 + 18u^2 + 6u + 1$ is a 254-bit prime. Let $\mathbb{G}_1 = E(\mathbb{F}_p)$, let $\mathbb{G}_2$ be the order-r subgroup of $E(\mathbb{F}_{p^{12}})$ comprising of points P for which $\sum_{i=0}^{11} \pi^i(P) = \infty$ (π being the p-power Frobenius), and let $\mathbb{G}_T$ be the order-r subgroup of $\mathbb{F}^*_{p^{12}}$. As shown in [1], there is a very efficient implementation of a Type 3 pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. One can check that there exists an 85-bit divisor d of $r-1$ which optimizes Cheon time (2). Hence, by Theorem 6(i), if there exists an efficiently-computable isomorphism $\phi_1 : \mathbb{G}_T \to \mathbb{G}_1$, then $\mathrm{DL}_{\mathbb{G}_2}$ can be solved in roughly 2^{85} time — much faster than the 2^{127} time required by Pollard's rho algorithm.

References

[1] D. Aranha, K. Karabina, P. Longa, C. Gebotys and J. López, "Faster explicit formulas for computing pairings over ordinary curves", *Advances in Cryptology – EUROCRYPT 2011*, Lecture Notes in Computer Science, 6632 (2011), 48–68.

[2] P. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order", *Selected Areas in Cryptography – SAC 2005*, Lecture Notes in Computer Science, 3897 (2006), 319–331.

[3] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing", *Journal of Cryptology*, 17 (2004), 297–319.

[4] D. Brown and R. Gallant, "The static Diffie-Hellman problem", http://eprint.iacr.org/2004/306.

[5] J. Cheon, "Security analysis of the Strong Diffie-Hellman problem', *Advances in Cryptology – EUROCRYPT 2006*, Lecture Notes in Computer Science, 4004 (2006), 1–11.

[6] D. Coppersmith, "Fast evaluation of logarithms in fields of characteristic two", *IEEE Transactions on Information Theory*, 30 (1984), 587–594.

[7] T. ElGamal, "A subexponential-time algorithm for computing discrete logarithms over $GF(p^2)$", *IEEE Transactions on Information Theory*, 31 (1985), 473–481.

[8] G. Frey and H. Rück, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves", *Mathematics of Computation*, 62 (1994), 865–874.

[9] S. Galbraith, Pairings, Ch. IX of I. Blake, G. Seroussi, and N. Smart, eds., *Advances in Elliptic Curve Cryptography*, Vol. 2, Cambridge University Press, 2005.

[10] S. Galbraith, F. Hess and F. Vercauteren, "Aspects of pairing inversion", *IEEE Transactions on Information Theory*, 54 (2008), 5719–5728.

[11] S. Galbraith, K. Paterson and N. Smart, "Pairings for cryptographers", *Discrete Applied Mathematics*, 156 (2008), 3113–3121.

[12] N. Koblitz and A. Menezes, "Pairing-based cryptography at high security levels", *Cryptography and Coding: 10th IMA International Conference*, Lecture Notes in Computer Science, 3796 (2005), 13–36.

[13] A. Lenstra and E. Verheul, "The XTR public key system", *Advances in Cryptology – CRYPTO 2000*, Lecture Notes in Computer Science, 1880 (2000), 1–19.

[14] A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", *IEEE Transactions on Information Theory*, 39 (1993), 1639–1646.

[15] A. Menezes and S. Vanstone, "ECSTR (XTR): Elliptic Curve Singular Trace Representation", Rump Session of Crypto 2000.

[16] D. Mireles Morales, "Cheon's algorithm, pairing inversion and the discrete logarithm problem", http://eprint.iacr.org/2008/300.

[17] J. Pollard, "Monte Carlo methods for index computation mod p", *Mathematics of Computation*, 32 (1978), 918–924.

[18] E. Verheul, "Evidence that XTR is more secure than supersingular elliptic curve cryptosystems", *Journal of Cryptology*, 17 (2004), 277–296.

E-mail address: kkarabin@uwaterloo.ca

E-mail address: edward.m.knapp@gmail.com

E-mail address: ajmeneze@uwaterloo.ca

Department of Combinatorics & Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada