J. Cryptol. (2010) 23: 457–476
DOI: 10.1007/s00145-009-9047-0

Journal of
CRYPTOLOGY

Discrete Logarithm Problems with Auxiliary Inputs*

Jung Hee Cheon
ISaC and Dept. of Mathematics, Seoul National University, Seoul, Korea
jhcheon@snu.ac.kr

Communicated by Phong Q. Nguen

Received 10 April 2008 and revised 20 May 2009
Online publication 11 December 2009

Abstract. Let g be an element of prime order p in an abelian group, and let $\alpha \in \mathbb{Z}_p$. We show that if g, g^{α}, and g^{α^d} are given for a positive divisor d of $p-1$, the secret key α can be computed deterministically in $O(\sqrt{p/d}+\sqrt{d})$ exponentiations by using $O(\max\{\sqrt{p/d}, \sqrt{d}\})$ storage. If g^{α^i} $(i = 0, 1, 2, \ldots, 2d)$ is given for a positive divisor d of $p+1$, α can be computed in $O(\sqrt{p/d}+d)$ exponentiations by using $O(\max\{\sqrt{p/d}, \sqrt{d}\})$ storage. We also propose space-efficient but probabilistic algorithms for the same problem, which have the same computational complexities with the deterministic algorithm.

As applications of the proposed algorithms, we show that the strong Diffie–Hellman problem and related problems with public $g^{\alpha}, \ldots, g^{\alpha^d}$ have computational complexity up to $O(\sqrt{d}/\log p)$ less than the generic algorithm complexity of the discrete logarithm problem when $p-1$ (resp. $p+1$) has a divisor $d \le p^{1/2}$ (resp. $d \le p^{1/3}$). Under the same conditions for d, the algorithm is also applicable to recovering the secret key in $O(\sqrt{p/d} \cdot \log p)$ for Boldyreva's blind signature scheme and the textbook ElGamal scheme when d signature or decryption queries are allowed.

Key words. Discrete logarithm, Auxiliary inputs, Baby-step giant-step, Pollard's kangaroo algorithm, Strong Diffie–Hellman, ElGamal encryption, Blind signature

1. Introduction

Let g be an element of an abelian group G of order p, and let $\alpha \in \mathbb{Z}_p$. By d auxiliary inputs we mean a set of values $\{g^{\alpha}, \ldots, g^{\alpha^d}\}$. Then a discrete logarithm problem (DLP) with auxiliary inputs is the problem that involves the determination of α when g^{α^i} $(i = 0, 1, 2, \ldots, d)$ as well as g, g^{α} are given. For other DL-related problems, we may define their variants with auxiliary inputs similarly. For example, the ℓ-Strong Diffie–Hellman (ℓ-SDH) problem [3,5] involves the determination of $g^{\frac{1}{c+\alpha}}$ for some c if $g, g^{\alpha}, \ldots, g^{\alpha^{\ell}}$ are given. Such additional information may weaken the problem; how-

* The preliminary version of this paper appeared in the *Proceedings of Eurocrypt 2006*, Lecture Notes in Computer Science 4004, Springer-Verlag [15].

ever, this problem and its variants such as Bilinear Diffie–Hellman Inversion (BDHI) problem [3] and the Bilinear Diffie–Hellman Exponent (BDHE) problem [9] are widely applied due to its flexibility when designing cryptosystems.

In this paper, we propose an algorithm for the DLP with auxiliary inputs. Suppose that $g, g^{\alpha}, g^{\alpha^d} \in G$ are given for a divisor d of $p-1$. Let $\alpha = \zeta^{k_0+((p-1)/d)k_1}$ for $0 \le k_0 < \frac{p-1}{d}$ and $0 \le k_1 < d$ for a primitive element ζ of $\mathbb{Z}_p^*$. The proposed algorithm computes k_0 and k_1 satisfying $\alpha^d = (\zeta^d)^{k_0}$ and $\alpha\zeta^{-k_0} = (\zeta^{(p-1)/d})^{k_1}$. Since we do not know α itself, k_0 is determined by checking the equality $g^{\alpha^d} = g^{\zeta^{dk_0}}$ for all k_0, or more efficiently $g^{\alpha^d} g^{\zeta^{-du}} = g^{\zeta^{dmv}}$ for all integers u, v with $0 \le u, v \le m := \lceil (p-1)/d \rceil$ in the style of baby-step and giant-step algorithm. Once k_0 is obtained, it is used to determine k_1 through a similar method.

This technique can be generalized to the case where $p+1$ has a divisor d and $g^{\alpha}, \ldots, g^{\alpha^{2d}}$ are given. To deal with such a case, we first determine two nonconstant rational functions $f_1, f_2 \in \mathbb{Z}_p[x]$ such that $f_1(\alpha) + f_2(\alpha)\theta$ is an element of the $(p+1)$th-order subgroup H of $\mathbb{Z}_p[\theta]^* \simeq \mathbb{F}_{p^2}^*$, where θ is a root of $X^2 - a$ for a quadratic nonresidue $a \in \mathbb{Z}_p^*$. The algorithm is similar to the above, except that we use the fact that $x_1 + y_1\theta = x_2 + y_2\theta$ in H if and only if $g^{x_1} = g^{x_2}$ and $g^{y_1} = g^{y_2}$.

The proposed algorithms require $O(\max\{\sqrt{p/d}, \sqrt{d}\})$ storage and $O(\sqrt{p/d} + \sqrt{d})$ exponentiations if g, g^{α} and g^{α^d} are given for a positive divisor d of $p-1$ or $O(\sqrt{p/d} + d)$ exponentiations (under the Extended Riemann Hypothesis (ERH)) if g^{α^i} $(i = 0, 1, 2, \ldots, 2d)$ is given for a positive divisor d of $p+1$. We further propose space-efficient algorithms for the DLP with auxiliary inputs, which are variants of Pollard's kangaroo algorithms [30] and have the same computational complexities as the deterministic algorithms above.

The proposed algorithms directly imply that the strong Diffie–Hellman problem and related problems with d auxiliary inputs can be solved in $O(\sqrt{p/d} + \sqrt{d})$ (or $O(\sqrt{p/d} + d)$) when $p-1$ (resp. $p+1$) has a divisor $d \le p^{1/2}$ (resp. $d \le p^{1/3}$). These complexities are up to a factor $O(\sqrt{d}/\log p)$ less than the lower bound $\Omega(\sqrt{p})$ of previously known generic algorithms for the DLP [28,36]. Surprisingly, it matches the previously known lower bound of these problems in the generic bilinear-group model [4, 12] when d is less than $p^{1/3}$. We remark that our result could be considered as a reminder that, when designing a cryptographic scheme and making parameter selections, it is important not to take lower bounds given by security proofs lightly: if a lower bound given by the security proof is lower than expected, this might actually be a warning that a matching attack exists and may be found later.

Under the same conditions given for d, the algorithm is also applicable for recovering the secret key in $O(\sqrt{p/d} \cdot \log p)$ for Boldyreva's blind signature scheme when d signature queries are allowed. Similar results hold for the textbook ElGamal scheme [18] when a decryption oracle is available, and for the conference keying protocol by Burmester and Desmedt [14] when key issuing oracles are available to the attacker.

In order to investigate the practicality of the proposed algorithm, we analyze some well-known elliptic curve parameters and show that either $p-1$ or $p+1$ has certain small divisors for the largest prime divisor p of the order of each elliptic curve in [8, 23,32,34]. Furthermore, we show that for any $0 < \tau < 1$, the probability that a prime

p is such that $p-1$ or $p+1$ has no divisor in $(e^{(\log p)^\tau/3}, p^{1/3}]$ decreases to zero as p increases to infinity. That is, for all but negligible proportion of primes p, $p-1$ and $p+1$ have a divisor of size appropriate for our algorithm.

Related Works After the author published the preliminary version [15] of this paper, he learned that a similar result to Theorem 1 (the case that $p-1$ has a divisor of proper size) was developed independently by Brown and Gallant [13]. They used it to study the static Diffie–Hellman problem and analyze some DLP-based protocols including basic ElGamal encryption, Chaum and van Antwerpen's undeniable signature scheme, and Ford and Kaliski's key retrieval scheme. A speed-up method of the proposed algorithms using a precomputed table was proposed by Kozaki, Kutsuma, and Matsuo [24]. Using their method, our algorithm can be performed in $O(\sqrt{p/d}+\sqrt{d})$ (resp. $O(\sqrt{p/d}+d)$) *group operations* in G, rather than exponentiations in G, for the case that $d|p-1$ (resp. $d|p+1$). Satoh generalized our algorithm into more general d using an embedding to $\mathrm{GL}(k, \mathbb{Z}_p)$ [33]. His algorithm for $k=1,2$ implies ours. However, it is not efficient in the case $k>2$.

Organization This paper is organized as follows: In Sect. 2, we propose algorithms for the DL problems with auxiliary inputs and analyze their complexities. In Sect. 3, we propose space-efficient versions of our algorithms. In Sect. 4, we introduce several applications of the proposed algorithms including SDH related problems. In Sect. 5, we analyze well-known elliptic curve parameters in order to verify whether or not our algorithms are applicable for the elliptic curve parameters, and we obtain the asymptotic distribution of primes resistant to the proposed attack. In Sect. 6, we conclude this paper.

2. The Proposed Algorithm

The following notation is used throughout this paper:

$\mathbb{Z}_p$	The finite field $\{0, 1\ldots, p-1\}$ of prime p elements
$\mathbb{Z}_p^*$	The multiplicative subgroup $\mathbb{Z}_p \setminus \{0\}$ of $\mathbb{Z}_p$
$a \bmod z$	The smallest nonnegative integer congruent modulo z
$\phi(\cdot)$	The Euler totient function
G	An abelian group (written multiplicatively) of order p wherein the equality of two elements is verified more efficiently than a multiplication in G
Mul_G	The cost of one multiplication in G
Inv_G	The cost of one inversion in G
$\mathrm{Exp}_G(n)$	The cost of one exponentiation of an element in G by a positive integer less than n
Comp_G	The cost to determine if two elements of G are identical including writing and reading an element of G

The Discrete Logarithm Problem (DLP) in G is defined as follows: For a given input $(g, g^\alpha) \in G^2$, compute $\alpha \in \mathbb{Z}_p$. In this section, we consider the DL problem with auxiliary inputs such as g^{α^i} for some integer i.

2.1. $p-1$ *Cases*

Lemma 1. *Let G be an abelian group of prime order p with generator g. Suppose that a generator ζ of $\mathbb{Z}_p^*$ and a positive divisor d of $p-1$ are given. If g, $g_1 := g^{\alpha}$ and $g_d := g^{\alpha^d}$ are given, α can be computed in time*

$$2\left(\left\lceil\sqrt{\frac{p-1}{d}}\right\rceil+\lceil\sqrt{d}\rceil\right)\cdot\left(\mathrm{Exp}_G(p)+\log p\cdot\mathrm{Comp}_G\right)+\left(5+o(1)\right)\cdot\mathrm{Exp}_{\mathbb{Z}_p}(p)$$

by using storage for $\max\{\lceil\sqrt{(p-1)/d}\rceil,\lceil\sqrt{d}\rceil\}$ *elements of G.*

Proof. Since $\alpha\in\mathbb{Z}_p^*$, there must exist nonnegative integers $k_0<\frac{p-1}{d}$ and $k_1<d$ such that

$$\alpha=\zeta^{k_0+k_1((p-1)/d)}.$$

Let $\hat{\zeta}:=\zeta^d$. Then $\alpha^d=\hat{\zeta}^{k_0}$. If we take $m=\lceil\sqrt{\frac{p-1}{d}}\rceil$ and $\hat{m}=\lfloor\frac{p-1}{md}\rfloor$ ($\hat{m}\le m$), k_0 can be expressed in a unique manner as $k_0=u+mv$, where $0\le u<m$ and $0\le v\le\hat{m}$. This implies that

$$(\alpha^d)\hat{\zeta}^{-u}=\hat{\zeta}^{mv}$$

or equivalently

$$g_d^{\hat{\zeta}^{-u}}=g^{\hat{\zeta}^{mv}}. \tag{1}$$

We construct a lookup table (baby-steps) which contains all the pairs $(g_d^{\hat{\zeta}^{-u}},u)$ for $0\le u<m$, and we sort the table by the first component. We compute $g^{\hat{\zeta}^{mv}}$ for each $0\le v\le\hat{m}$ (giant-steps) and compare with the lookup table in order to identify coincidences. Note that the terms in both sides of (1) can be computed by repeated exponentiations by either $\hat{\zeta}^{-1}$ or $\hat{\zeta}^m$. Thus, we can determine a pair of (u,v) that satisfies (1) in $2m\cdot(\mathrm{Exp}_G(p)+\log p\cdot\mathrm{Comp}_G)+2\cdot\mathrm{Exp}_{\mathbb{Z}_p}(p)$ by using storage for m elements of G.

Now we apply a similar method to find a nonnegative integer $k_1<d$ from $\alpha\zeta^{-k_0}=\zeta^{\frac{p-1}{d}k_1}$. Let $\check{\zeta}=\zeta^{\frac{p-1}{d}}$. Then, there exist nonnegative integers u',v' such that

$$\left(\alpha\zeta^{-k_0}\right)\check{\zeta}^{-u'}=\check{\zeta}^{m'v'}\quad(0\le u'<m',\ 0\le v'\le\hat{m}'), \tag{2}$$

where $m':=\lceil\sqrt{d}\rceil$ and $\hat{m}':=\lfloor\frac{d}{m'}\rfloor$. Equation (2) is equivalent to

$$\left(g_1^{\zeta^{-k_0}}\right)^{\check{\zeta}^{-u'}}=g^{\check{\zeta}^{m'v'}}. \tag{3}$$

Using the same method described above, we can find integers u' and v' that satisfy (3) in $2m'\cdot(\mathrm{Exp}_G(p)+\log p\cdot\mathrm{Comp}_G)+(3+o(1))\mathrm{Exp}_{\mathbb{Z}_p}(p)$ by using storage for m' elements of G. □

We require a generator of $\mathbb{Z}_p^*$ for our attacks. Since $\mathbb{Z}_p^*$ is a cyclic group with $\phi(p-1)$ generators, a random element in $\mathbb{Z}_p^*$ is a generator [26, p. 162] with probability

$$\frac{\phi(p-1)}{(p-1)} > \frac{1}{6\log\log(p-1)},$$

which is sufficiently large. However, the best-known deterministic algorithm [42] for determining a generator of $\mathbb{Z}_p^*$ requires $p^{1/4+o(1)}$ when the factorization of $p-1$ is known. Since the integer factorization is performed with constant probability of success deterministically in $O(p^{1/4})$ (by heuristic analysis) with using Pollard's rho algorithm, a generator of $\mathbb{Z}_p^*$ is computed in $p^{1/4+o(1)}$ [38]. Therefore, since we assumed that Exp_G dominates Comp_G, we can conclude that the computation of the discrete logarithm with d auxiliary inputs requires a deterministic running time given by $O(\sqrt{(p-1)/d}+\sqrt{d})\cdot \mathrm{Exp}_G(p)$. It proves the following theorem. We note that a similar result to the following theorem was developed independently by Brown and Gallant [13].

Theorem 1. *Let G be an abelian group of prime order p with generator g. Suppose that d is a positive divisor of $p-1$. If g, $g_1 := g^{\alpha}$ and $g_d := g^{\alpha^d}$ are given, α can be computed deterministically in time*

$$O\left(\sqrt{\frac{p-1}{d}}+\sqrt{d}\right)\cdot \mathrm{Exp}_G(p)$$

by using storage for $\max\{\lceil\sqrt{(p-1)/d}\rceil, \lceil\sqrt{d}\rceil\}$ *elements of G.*

If $p-1$ is factorized into a product of small prime divisors $d_1, \ldots, d_t$ and $g^{\alpha^{(p-1)/d_i}}$ is known for each i, we obtain a very efficient algorithm for computing α as the following corollary.

Corollary 1. *Let G be an abelian group of prime order p with generator g. Suppose that a factorization of $p-1$ is given as $p-1 = d_1 d_2 \cdots d_t$ for pairwise relatively prime d_i's. If g and $g_{(p-1)/d_i} := g^{\alpha^{(p-1)/d_i}}$ for $1 \le i \le t$ are given, α can be computed in time*

$$O\left(\sum_{i=1}^{t}\sqrt{d_i}\right)\cdot \mathrm{Exp}_G(p)$$

by using storage for $\max_{1\le i\le t}\lceil\sqrt{d_i}\rceil$ *elements of G.*

Proof. Let $\alpha = \zeta^k$ for a generator ζ of $\mathbb{Z}_p^*$ and $\zeta_i := \zeta^{(p-1)/d_i}$ be an element of order d_i in $\mathbb{Z}_p^*$. Since $(\alpha^{(p-1)/d_i})^{d_i} = 1$ and ζ_i generates all the d_ith roots of unity, there must be a unique nonnegative integer k_i less than d_i that satisfies $\alpha^{(p-1)/d_i} = \zeta_i^{k_i}$, which is equivalent to

$$g^{(p-1)/d_i} = g^{\zeta_i^{k_i}}.$$

That is,

$$(g_{(p-1)/d_i})^{\zeta_i^{-u_i}} = g^{\zeta_i^{\lceil\sqrt{d_i}\rceil v_i}} \quad \text{for } 0 \le u_i, v_i < \lceil\sqrt{d_i}\rceil.$$

By using the baby-step giant-step algorithm, we can compute k_i in $\lceil\sqrt{d_i}\rceil\ \mathrm{Exp}_G(p)$ by using storage for $O(\sqrt{d_i})$ G elements. Since k satisfies $k \equiv k_i \mod d_i$, we can compute k by performing the above step for $1 \le i \le t$ and by using the Chinese Remainder Theorem. The total complexity is $O(\sum_{i=1}^{t}\sqrt{d_i})\ \mathrm{Exp}_G(p)$ when using storage for $\max_{1\le i\le t}\lceil\sqrt{d_i}\rceil$ G elements. Finally, ζ_i can be computed in $O(\sqrt{d_i})$ for each i. This completes the proof. □

2.2. $p+1$ *Cases*

In this subsection, we consider the case that d is a divisor of $p+1$ and $g^{\alpha}, \ldots, g^{\alpha^{2d}}$ are given.

Let a be a quadratic nonresidue of $\mathbb{Z}_p$. If we denote by θ a root of $X^2 - a$ in an algebraically closed field of $\mathbb{Z}_p$, then $\mathbb{Z}_p[\theta] \cong \mathbb{Z}_p[X]/(X^2 - a)$ is a finite field of order p^2 and $\theta^p = -\theta$. We first determine two nonconstant rational functions $f_1, f_2 \in \mathbb{Z}_p[x]$ such that $\beta := f_1(\alpha) + f_2(\alpha)\theta$ is an element of the $(p+1)$th-order subgroup H of $\mathbb{Z}_p[\theta]^*$. Then we choose a generator ζ of H and let $\beta = \zeta^{k_0 + k_1\frac{p+1}{d}}$ for $0 \le k_0 < \frac{p+1}{d}$ and $0 \le k_1 < d$.

First, we compute k_0 with $\hat{\zeta}^{k_0} = \beta^d$ for $\hat{\zeta} = \zeta^d$. It is done by checking the equality

$$\beta^d \hat{\zeta}^{-mv} = \hat{\zeta}^u$$

for all integers u, v with $0 \le u, v \le m := \lceil\frac{p+1}{d}\rceil$ as in the baby-step and giant-step method. Since we do not know the powers of β themselves, the equality is checked by using the fact that $x_1 + y_1\theta = x_2 + y_2\theta$ in H if and only if $g^{x_1} = g^{x_2}$ and $g^{y_1} = g^{y_2}$. Once k is obtained, we apply a similar method to find k in the baby-step giant-step style from

$$\beta\zeta^{-k_0} = \zeta^{((p+1)/d)k_1}.$$

Finally, α is computed by solving the equation $f_1(\alpha) + f_2(\alpha)\theta = \zeta^{k_0 + k_1((p+1)/d)}$.

Lemma 2. *Let G be an abelian group of prime order p with generator g. Suppose that a positive divisor d of $p+1$ and a generator of $\mathbb{Z}_p[\theta]$ are given, where θ is a root of $X^2 - a$ for a quadratic nonresidue element a of $\mathbb{Z}_p$. If $g_i := g^{\alpha^i}$ for $i = 1, 2, \ldots, 2d$ is given, α can be computed in time*

$$(3d + 6m + 6m' + 6)\cdot \mathrm{Exp}_G(p) + (3d + 2m + 2m' + 2)\cdot \mathrm{Mul}_G + 4\cdot \mathrm{Exp}_{\mathbb{Z}_p}(p)$$
$$+ (7d + 8m + 8m')\mathrm{Mul}_{\mathbb{Z}_p} + 3d\cdot \mathrm{Inv}_{\mathbb{Z}_p} + 2(m + m')\log p \cdot \mathrm{Comp}_G$$

by using storage for $\max\{\sqrt{(p+1)/d}, \sqrt{d}\}$ elements of G, where $m = \lceil\sqrt{\frac{p+1}{d}}\rceil$ and $m' = \lceil\sqrt{d}\rceil$.

Proof. Let H be the subgroup of order $p+1$ of $\mathbb{F}_p[\theta]^*$ with generator ζ, which can be obtained by raising the $(p-1)$-st power to a generator of $\mathbb{F}_p[\theta]^*$, and let $\beta = \beta_0 + \beta_1\theta$ for $\beta_0 = (1+a\alpha^2)/(1-a\alpha^2) \in \mathbb{F}_p$ and $\beta_1 = 2\alpha/(1-a\alpha^2) \in \mathbb{F}_p$. We have $\beta \in H$ since

$$\beta^{p+1} = \beta \cdot \beta^p = (\beta_0 + \beta_1\theta)\big(\beta_0 + \beta_1\theta^p\big) = \beta_0^2 - a\beta_1^2 = 1. \tag{4}$$

Hence there exist nonnegative integers $k_0 < \frac{p+1}{d}$ and $k_1 < d$ such that $\beta = \zeta^{k_0+k_1((p+1)/d)}$.

Let $\hat{\zeta} := \zeta^d$. Then we have $\beta^d = \hat{\zeta}^{k_0}$. For convenience, we denote $\hat{\zeta}^i = s_i + t_i\theta$ for some $s_i, t_i \in \mathbb{Z}_p$, where the index i is defined modulo $\frac{p+1}{d}$. Further we denote

$$\beta^d = \frac{(1+\alpha\theta)^{2d}}{(1-a\alpha^2)^d} = \frac{P_1(\alpha) + P_2(\alpha)\theta}{P_3(\alpha)} \tag{5}$$

for

$$P_1(x) = \sum_{i=0}^{d} \binom{2d}{2i} a^i x^{2i}, \qquad P_2(x) = \sum_{i=0}^{d-1} \binom{2d}{2i+1} a^i x^{2i+1}, \quad \text{and}$$

$$P_3(x) = \sum_{i=0}^{d} \binom{d}{i} (-a)^i x^{2i} \quad \text{in } \mathbb{Z}_p[x],$$

where the first equality comes from $1 + a\alpha^2 + 2\alpha\theta = (1+\alpha\theta)^2$.

Then, there must exist u and v such that

$$\beta^d \hat{\zeta}^{-u} = \hat{\zeta}^{mv} \quad (0 \le u < m,\ 0 \le v \le \hat{m}), \tag{6}$$

where $m := \lceil \sqrt{\frac{p+1}{d}} \rceil$ and $\hat{m} := \lfloor \frac{p+1}{md} \rfloor \le m$. Equation (6) is equivalent to

$$\big(P_1(\alpha)s_{-u} + aP_2(\alpha)t_{-u}\big) + \big(P_1(\alpha)t_{-u} + P_2(\alpha)s_{-u}\big)\theta = P_3(\alpha)(s_{mv} + t_{mv}\theta).$$

We construct a lookup table (baby-steps) which contains all the triples $(g^{P_1(\alpha)s_{-u}+aP_2(\alpha)t_{-u}}, g^{P_1(\alpha)t_{-u}+P_2(\alpha)s_{-u}}, u)$ for all $0 \le u < m$, and we sort the table by first component. We compute $(g^{P_3(\alpha)s_{mv}}, g^{P_3(\alpha)t_{mv}})$ for each $0 \le v \le \hat{m}$ (giant-steps) and refer to the lookup table in order to find a unique pair (u, v) that satisfies (6). Then, $k = u + mv$. The detailed procedure is listed as follows:

- Step 1: Compute $\hat{\zeta} = \zeta^d$ and $\hat{\zeta}^{-1}$ in $2 \cdot \mathrm{Exp}_{\mathbb{Z}_p}(p)$
- Step 2: Compute (s_{-u}, t_{-u}) and (s_{mv}, t_{mv}) for $0 \le u < m$ and $0 \le v \le \hat{m}$ in $8m \cdot \mathrm{Mul}_{\mathbb{Z}_p}$
- Step 3: Compute $a^i \bmod p$, $\binom{2d}{2i} \bmod p$, $\binom{2d}{2i+1} \bmod p$, and $\binom{d}{i} \bmod p$ for $1 \le i \le d$, in order to obtain the coefficients of $P_1, P_2, P_3 \in \mathbb{Z}_p[x]$, in $7d \cdot \mathrm{Mul}_{\mathbb{Z}_p} + 3d \cdot \mathrm{Inv}_{\mathbb{Z}_p}$.
- Step 4: Compute $g^{P_1(\alpha)}, g^{P_2(\alpha)}$ and $g^{P_3(\alpha)}$ from $g, g_1, \ldots, g_{2d}$ in $(3d+2) \cdot (\mathrm{Exp}_G(p) + \mathrm{Mul}_G)$.

Step 5: Compute s_i- or t_i-powers of $g^{P_j(\alpha)}$ for some necessary i and j to form a table and perform the comparisons in $m \cdot (6\mathrm{Exp}_G(p) + 2\mathrm{Mul}_G) + 2m \log p \cdot \mathrm{Comp}_G$.

It requires storage for $\max\{\lceil\sqrt{(p+1)/d}\rceil, \lceil\sqrt{d}\rceil\}$ elements of G.

Now we apply a similar method to find a nonnegative integer $k_1 < d$ from $\beta\zeta^{-k_0} = \zeta^{\frac{p+1}{d}k_1}$. Let $\check{\zeta} = \zeta^{\frac{p+1}{d}}$. Then, there must exist nonnegative integers u', v' such that

$$\beta\zeta^{-k_0}\check{\zeta}^{-u'} = \check{\zeta}^{m'v'} \quad (0 \le u' < m',\ 0 \le v' \le \hat{m}'), \tag{7}$$

where $m' := \lceil\sqrt{d}\rceil$ and $\hat{m}' := \lfloor\frac{d}{m'}\rfloor$. We denote $\zeta^{-k_0}\check{\zeta}^{-i} = s'_i + t'_i\theta$ and $\check{\zeta}^{m'i} = s''_i + t''_i\theta$ for some $s'_i, t'_i, s''_i, t''_i \in \mathbb{Z}_p$, where the index i is defined modulo d. Then (7) is equivalent to

$$\big((1+a\alpha^2)s'_{u'} + 2a\alpha t'_{u'}\big) + \big((1+a\alpha^2)t'_{u'} + 2\alpha s'_{u'}\big)\theta = \big(1-a\alpha^2\big)\big(s''_{v'} + t''_{v'}\theta\big). \tag{8}$$

We construct a lookup table that contains all the triples

$$\big(g^{(1+a\alpha^2)s'_{u'}+2a\alpha' t_{u'}}, g^{(1+a\alpha^2)t'_{u'}+2\alpha s'_{u'}}, u\big)$$

for all $0 \le u' < m'$ and sort the table by the first component. Then, compute $(g^{(1-a\alpha^2)s''_{v'}}, g^{(1-a\alpha^2)t''_{v'}})$ for each $0 \le v' < \hat{m}'$ in order to find a pair (u', v') that satisfies (2). The detailed procedure is listed as follows:

Step 1: Compute ζ^{k_0} and ζ^{-1} in $2 \cdot \mathrm{Exp}_{\mathbb{Z}_p}(p)$.

Step 2: Compute $(s'_{u'}, t'_{u'})$ and $(s''_{v'}, t''_{v'})$ for $0 \le u' < m'$ and $0 \le v' \le \hat{m}'$ in $(m+\hat{m}) \cdot \mathrm{Mul}_H$ or equivalently $8m' \cdot \mathrm{Mul}_{\mathbb{Z}_p}$.

Step 3: Compute $g^{(1+a\alpha^2)}$, $g^{2a\alpha}$, $g^{1-a\alpha^2}$ and then compute their powers by appropriate s', t', s'', or t'' to form a table and perform comparisons. This can be performed in $(3+6m') \cdot \mathrm{Exp}_G(p) + 2m \cdot \mathrm{Mul}_G + (2m' \log p) \cdot \mathrm{Comp}_G$.

It requires storage for m' elements of G.

Finally, $\alpha = \frac{\beta_1}{\beta_0+1}$ can be computed from $\beta = \beta_0 + \beta_1\theta$ with one additional division or exponentiation in G. This completes the proof. □

The previous lemma requires a quadratic nonresidue element of $\mathbb{F}_p$, which is easy to obtain since a half of elements of $\mathbb{F}_p^*$ are quadratic nonresidues and quadratic residuosity is easily checked by raising a power of $(p-1)/2$ to the element. Moreover, it is known [2,37] that the least positive integer that is not a quadratic residue modulo p is at most $2\log p$ so that we can take a quadratic nonresidue in most $2\log p$ steps. Currently, the best-known deterministic algorithm [35] for determining a generator of $\mathbb{F}_{p^2}^*$ requires $p^{1/2+o(1)}$ when the factorization of p^2-1 is known. However, assuming the Extended Riemann-Hypothesis (ERH), this can be performed by $O((\log p)^{O(1)})$ [35]. Assuming that a comparison operation is faster than an exponentiation, we have the following theorem.

Theorem 2. *Let G be an abelian group of prime order p with generator g. Suppose that a positive divisor d of $p+1$ and $g_i := g^{\alpha^i}$ for $i = 1, 2, \ldots, 2d$ are given. Then, α can be computed deterministically in time*

$$O\left(\sqrt{\frac{p+1}{d}} + d\right) \cdot \mathrm{Exp}_G(p) + O(d + \log p) \cdot \mathrm{Mul}_{\mathbb{Z}_p}$$

by using storage for $\max\{\sqrt{\frac{p+1}{d}}, \sqrt{d}\}$ *elements of G, assuming the ERH.*

Note that if $d \leq p^{1/3}$, then Theorem 2 states that the secret key can be computed in $O(\sqrt{p/d})$ exponentiations by using $O(\sqrt{p/d})$ storage.

Remark 1. We may compare our proof with the proof of the equivalence between the DLP and DHP in some circumstances [16,25]. While the latter assumes that Diffie–Hellman oracles $DH(g^x, g^y) = g^{xy}$ are accessible, for the situations in our study, the Diffie–Hellman oracle can be used only when x is fixed and $y = x^\ell$ for some small ℓ. This restriction becomes an obstacle when we attempt to generalize the proposed algorithm to the case that d divides $p^k - 1$ for $k > 2$ or the order of an (hyper)elliptic curve over $\mathbb{F}_p$ as in [16,25]. Satoh generalized our algorithm to the case that d is a divisor of $\Phi_k(p)$ by using an embedding of $\mathbb{Z}_p$ into $\mathrm{GL}(k, \mathbb{Z}_p)$, where Φ_k is the kth cyclotomic polynomial [33]. His algorithm for $k = 1, 2$ includes our algorithms. However, it is not efficient for $k > 2$.

3. Probabilistic Algorithms

In this section, we propose a variant of Pollard's kangaroo algorithm [30,31] for DLP with auxiliary inputs.

First, we introduce Pollard's kangaroo algorithm. Let E be an abelian group of order n, and let $\zeta, \xi \in E$. Suppose that $\xi = \zeta^x$ for $a \leq x \leq b$ with $a, b \in \mathbb{Z}$. Let $S = \{g^{s_1}, \ldots, g^{s_r}\}$ be a set of jumps, where the s_i's are small integers with mean size $\approx \sqrt{b-a}$, and let $\nu : E \to \{1, \ldots, r\}$ be a fixed random function. We define a random walk

$$F : E \to E, \quad y \mapsto y\zeta^{s(y)},$$

where $s : E \to \mathbb{Z}$ with $s(y) = s_{\nu(y)}$. The tame kangaroo starts from ζ^c for some known integer c and follows the path

$$t_0 = \zeta^c, \qquad t_{i+1} = F(t_i) \quad (i \geq 0). \tag{9}$$

The wild kangaroo starts from ξ and follows the path

$$w_0 = \xi, \qquad w_{i+1} = F(w_i) \quad (i \geq 0). \tag{10}$$

If t_i or w_i satisfies some condition for distinguished points, it is compared with the list of distinguished points and, if there is no match, stored in the list with its travel length

$\sum_{j=0}^{i} s(t_j)$ or $\sum_{j=0}^{i} s(w_j)$. If a match is found between two kangaroos, say, $t_{i'} = w_i$, then we can compute

$$x \equiv c + \sum_{j=0}^{i-1} s(t_j) - \sum_{j=0}^{i'-1} s(w_j) \bmod n.$$

If no match is found until the predetermined length of travels, another wild kangaroo is launched with $\xi\zeta^{c'}$ for small c'. When appropriate parameters are chosen, the expected number of operations is approximately $2\sqrt{b-a} + 1/\Theta$, where Θ is the proportion of distinguished points in a cyclic group generated by ζ.

3.1. $p-1$ *Cases*

Let us apply this for the DLP with auxiliary inputs. Let G be an abelian group of prime order p, and let $g, g^{\alpha}, g^{\alpha^d} \in G$ be given for a positive divisor d of $p-1$. Let ζ be a primitive element of $\mathbb{F}_p^*$, and let E be the cyclic group generated by $\hat{\zeta} = \zeta^d$. We apply Pollard's kangaroo algorithm for $\hat{\zeta}$ and $\hat{\alpha} = \alpha^d \in E$ to obtain $0 \le k_0 < \frac{p-1}{d}$ with $\hat{\alpha} = \hat{\zeta}^{k_0}$.

Since we do not know $\hat{\alpha}$, we cannot launch a wild kangaroo. Hence we need to modify the algorithm: We define $\bar{\nu} : g^E := \{g^h | h \in E\} \to \{1, \ldots, r\}$ be a fixed random function and

$$\nu : E \to \{1, \ldots, r\}; \quad h \mapsto \bar{\nu}\big(g^h\big).$$

Note that ν runs through all functions from E to $\{1, \ldots, r\}$ when $\bar{\nu}$ runs through all functions from g^E to $\{1, \ldots, r\}$ since a function $E \to g^E; x \mapsto g^x$ is bijective. The index function s and the random walk is defined as above.

For a wild kangaroo, we have $g^{w_0} = g^{\hat{\alpha}}$ and $g^{w_{i+1}}$ is computed from g^{w_i} as $(g^{w_i})^{\zeta^{s(w_i)}}$ for $i \ge 0$. The path of a tame kangaroo is computed in a similar manner. Since $g^{w_i} = g^{v_j}$ in G is equivalent to $w_i \equiv v_j \bmod p$, we can find a match in the list of distinguished points in the same running time with the original Pollard's kangaroo in E. If ζ^{s_i} is precomputed, each jump requires one exponentiation in G. Hence the discrete logarithm $k_0 = \log_{\hat{\zeta}} \hat{\alpha}$ is computed approximately in $(2\sqrt{(p-1)/d} + 1/\Theta)$ exponentiations in G with storage $O(\Theta\sqrt{(p-1)/d})$.

Let $\alpha = \zeta^k$ for $0 \le k < p-1$. Then there exists a nonnegative integer $k_1 < d$ such that $k = k_0 + k_1 \frac{p-1}{d}$. If we let $\check{\zeta} = \zeta^{(p-1)/d}$ and $\check{\alpha} = \alpha\zeta^{-k_0}$, we have $\check{\alpha} = \check{\zeta}^{k_1}$. We apply the above algorithm again for $\check{\zeta}$ and $\check{\alpha}$ to obtain k_1, which requires approximately $2\sqrt{d}$ exponentiations in G.

Therefore the proposed algorithm computes α approximately in

$$O\big(\sqrt{(p-1)/d} + \sqrt{d} + \Theta^{-1}\big)$$

exponentiations in G with storage $O(\Theta \cdot \max\{\sqrt{(p-1)/d}, \sqrt{d}\})$.

3.2. $p+1$ *Cases*

Let us consider the case that d is a divisor of $p+1$ and $g, g^{\alpha}, \ldots, g^{\alpha^{2d}}$ are given. As in Lemma 2, suppose that a is a quadratic nonresidue in $\mathbb{Z}_p$ and θ is a root of

$X^2 - a$. Let ζ be an element of order $(p+1)$ in $\mathbb{Z}_p[\theta]$, and let E be the cyclic group generated by $\hat{\zeta} = \zeta^{(d+1)/d}$. We define

$$\nu : E \to \{1, \ldots, r\}; \quad h_1 + h_2\theta \mapsto \bar{\nu}\big(g^{h_1}, g^{h_2}\big),$$

where $\bar{\nu} : g^E := \{(g^{h_1}, g^{h_2}) | h_1 + h_2\theta \in E\} \to \{1, \ldots, r\}$ is a fixed random function. The index function s and the random walk is defined as above.

We take

$$\beta := \frac{1 + a\alpha^2}{1 - a\alpha^2} + \frac{2\alpha}{1 - a\alpha^2}\theta, \qquad \hat{\beta} := \beta^d = \frac{P_1(\alpha) + P_2(\alpha)\theta}{P_3(\alpha)} \in E,$$

for three polynomials P_1, P_2, P_3 of degree $\le 2d$. Let $\beta = \zeta^k$ for $0 \le k < p+1$. Then there exist $0 \le k_0 < \frac{p+1}{d}$ and $0 \le k_1 < d$ such that $k = k_0 + k_1 \frac{p+1}{d}$. First, we compute k_0 satisfying $\hat{\beta} = \hat{\zeta}^{k_0}$.

Take w_i as in the (10) starting from $w_0 = \hat{\beta}$. Let us denote by $w_i = w_i' + w_i''\theta$ for $w_i', w_i'' \in \mathbb{F}_p$ and $W_i' = g^{P_3(\alpha)w_i'}$ and $W_i'' = g^{P_3(\alpha)w_i''}$. First, we compute

$$W_0' = g^{P_1(\alpha)}, \qquad W_0'' = g^{P_2(\alpha)}.$$

If we denote $\hat{\zeta}^{s(w_i)} = U_i + V_i\theta$ for $U_i, V_i \in \mathbb{F}_p$, then we have

$$w_{i+1} = \big(w_i' + w_i''\theta\big)(U_i + V_i\theta) = \big(w_i'U_i - aw_i''V_i\big) + \big(w_i'V_i + w_i''U_i\big)\theta.$$

Hence we can compute

$$W_{i+1}' = \big(W_i'\big)^{U_i}\big(W_i''\big)^{-aV_i}, \qquad W_{i+1}'' = \big(W_i'\big)^{V_i}\big(W_i''\big)^{U_i}.$$

The path of a tame kangaroo is computed in a similar manner starting from $t_0 = \hat{\zeta}^c$ for some integer c. Then each jump requires four exponentiations, and so the discrete logarithm k_0 of β^d to the base $\hat{\zeta}$ is computed approximately in $(8\sqrt{(p+1)/d} + 4(d + o(1)) + \Theta^{-1})$ exponentiations in G with storage $O(\Theta\sqrt{(p+1)/d})$.

Once k_0 is obtained, we take $\check{\zeta} = \zeta^{(p+1)/d}$ and $\check{\xi} = \beta\zeta_0^{-k_0}$ so that $\check{\xi} = \check{\zeta}^{k_1}$. Then we can obtain k_1 in a similar manner with approximately $8\sqrt{d}$ exponentiations in G.

Therefore the proposed algorithm computes α approximately in

$$O\big(\sqrt{(p+1)/d} + d + \Theta^{-1}\big)$$

exponentiations in G with storage $O(\Theta \cdot \max\{\sqrt{(p+1)/d}, \sqrt{d}\})$.

4. Applications

4.1. *Strong Diffie–Hellman Problems and Their Variants*

Many cryptosystems are designed on the basis of the DL problem; however, in most of these systems, their security is equivalent to a weaker variant of the DL problem rather than the DL problem itself. Two of the most popular weaker variants are given as follows:

The Computational Diffie–Hellman (CDH) Problem. For a given input $(g, g^x, g^y) \in G^3$, compute $g^{xy} \in G$.

The Decisional Diffie–Hellman (DDH) Problem. For a given input $(g, g^x, g^y, g^z) \in G^4$, decide whether $z = xy$ in $\mathbb{Z}_p$.

Recently, some weakened variants of the CDH problem have been introduced, and they are being used to construct cryptosystems with various functionalities or to prove some cryptosystem's security without random oracles. One characteristic of such problems is to disclose $g, g^x, \ldots, g^{x^\ell}$ for the secret key x and some integer ℓ. Thus our attacks is applicable to these problems. We introduce such problems.

The ℓ-weak Diffie–Hellman (ℓ-wDH) Problem. For a given input $(g, g^x, \ldots, g^{x^\ell}) \in G^{\ell+1}$, compute $g^{1/x} \in G$. This problem was introduced by Mitsunari, Sakai, and Kasahara for a traitor tracing scheme [27]. It is called also the ℓ-Diffie–Hellman Inversion (ℓ-DHI) problem [4].

Another class of problems are defined on a group with bilinear maps. We further assume that we have an efficiently computable bilinear map $e : G \times G \to G_T$ for a cyclic group G_T of order p.

The ℓ-Strong Diffie–Hellman (ℓ-SDH) Problem. For a given input $(g, g^x, \ldots, g^{x^\ell}) \in G^{\ell+1}$, compute a pair $(c, g^{1/(x+c)}) \in \mathbb{Z}_p \times G$ for a freely selected value $c \in \mathbb{Z}_p \setminus \{-x\}$. This problem was first introduced by Boneh and Boyen for the construction of a short signature scheme, which is provably secure in the standard model (without random oracles) [4], and it was used later for a short group signature scheme [7].

The ℓ-Bilinear Diffie–Hellman Inversion (ℓ-BDHI) Problem. For a given input $(g, g^x, \ldots, g^{x^\ell}) \in G^{\ell+1}$, compute $e(g, g)^{1/x} \in G_T$. This problem was introduced by Boneh and Boyen for the construction of an identity-based encryption that is secure in the standard model [3]. It is also used to construct verifiable random functions [17] and a hierarchical identity-based encryption scheme with constant size ciphertext [9].

The ℓ-Bilinear Diffie–Hellman Exponent (ℓ-BDHE) Problem. For a given input $(g, g^x, \ldots, g^{x^{\ell-1}}, g^{x^{\ell+1}}, \ldots, g^{x^{2\ell}}) \in G^{2\ell+1}$, compute $e(g, g)^{x^\ell} \in G'$. This problem was introduced by Boneh, Boyen, and Goh [9] for the initial construction of a hierarchical identity-based encryption scheme with constant size ciphertext, and it was used later for a public-key broadcast encryption scheme with constant size transmission overhead [10]. A blind and partially blind signature was proposed by Okamoto [29] based on both ℓ-BDHE and ℓ-BDHI.

A more general (and slightly weaker) version of the SDH problem was introduced in [5]. It is defined on a bilinear group pair (G_1, G_2), on which there exist a group G_T and a nondegenerate bilinear map $e : G_1 \times G_2 \to G_T$ such that the group order $p = |G_1| = |G_2| = |G_T|$ is prime, and the pairing e and the group operations in G_1,

G_2, and G_T are all efficiently computable [5,20]. The problem is stated as follows: For a given input $(g_1, g_1^x, \ldots, g_1^{x^\ell}, g_2, g_2^x) \in G_1^{\ell+1} \times G_2^2$, compute a pair $(c, g_1^{1/(x+c)}) \in \mathbb{Z}_p \times G_1$ for a freely selected value $c \in \mathbb{Z}_p \setminus \{-x\}$. In this situation, our algorithm can be applied to compute x from $(g_1, g_1^x, \ldots, g_1^{x^\ell}) \in G_1^{\ell+1}$ or $(h, h^x, \ldots, h^{x^{\ell+1}}) \in G_T^{\ell+2}$ for $h = e(g_1, g_2)$, which gives a pair $(c, g_1^{\frac{1}{x+c}})$ for any $c \in \mathbb{Z}_p$.

Recently, Jao, and Yoshida [22,43] applied the proposed attack to recover the secret key of the Boneh–Boyen signature scheme [4] using the d signature queries. Since the reduction takes only at most $O(d^2)$ exponentiations in G, the security of the scheme is equivalent to the SDH problem for small $d \leq p^{1/5}$.

4.2. *Blind Signature Scheme Based on the GDH Assumption*

A Gap–Diffie–Hellman (GDH) group is an abelian group in which there exists a polynomial time algorithm for solving the decisional Diffie–Hellman problem and it is assumed that no polynomial-time algorithm exists for solving the computation Diffie–Hellman problem.

Boldyreva proposed a blind signature scheme on a Gap–Diffie–Hellman group [11]. The scheme is as follows: Let G be a GDH group of prime order p and g a generator of G. Let $H : \{0,1\}^* \to G$ be a full domain hash function [8]. A signer has a private key $x \in \mathbb{Z}_p$ and the corresponding public key $y = g^x$. In order to blindly sign a message $M \in \{0,1\}^*$, a user selects a random $k \in \mathbb{Z}_p^*$, computes $M' = H(M)g^k$, and sends it to the signer. The signer computes $\sigma' = (M')^x$ and sends it back to the user. Then the user computes the signature $\sigma = \sigma'/y^k (= H(M)^x)$ of the message M.

This scheme is shown to be secure against one-more forgery under chosen message attacks in the random oracle model [11], that is the standard security notion for blind signature schemes. However, since the signer does not have any information on the message to be signed, we may use this blind signing phase as a Diffie–Hellman oracle and hence reduce the security of this scheme under chosen message attacks: A chosen-message attacker $\mathcal{A}$ takes a random $\gamma_1 \in \mathbb{Z}_p$ and requests a signature on the message $y \cdot g^{\gamma_1}$. From the signature $\sigma_1 = (y \cdot g^{\gamma_1})^x$, $\mathcal{A}$ obtains $g_2 := g^{x^2} = \sigma_1/y^{\gamma_1}$. Second, $\mathcal{A}$ takes another random $\gamma_2 \in \mathbb{Z}_p$ and requests a signature on the message $g_2 \cdot g^{\gamma_2}$. From the signature $\sigma_2 = (g_2 \cdot g^{\gamma_2})^x$, $\mathcal{A}$ obtains $g_3 := g^{x^3} = \sigma_2/y^{\gamma_2}$. If ℓ signature queries are allowed, $\mathcal{A}$ repeats this procedure ℓ times to obtain $g_1, g_2, \ldots, g_{\ell+1}$ $(g_i := g^{x^i})$. By Theorems 1 and 2, if $p-1$ has a divisor $d \leq \min\{\ell+1, p^{1/2}\}$ or $p+1$ has a divisor $d \leq \min\{(\ell+1)/2, p^{1/3}\}$, the secret key x can be computed in $O(\sqrt{p/d})$. That is, the security of the scheme is reduced by $O(\sqrt{d})$ in comparison to that of the GDH assumption.

It must be noted that the attack does not imply that the security proof of the scheme is wrong, but it implies that further quantitative analysis on security reduction is required. In fact, the security proof of BLS signature schemes on which the Boldyreva's blind signature scheme is based shows that the advantage of an adversary can be increased by q_S, when q_S signature queries are allowed [8].

This method is applicable in a similar manner to schemes which respond by its secret key power for an unknown message. For example, the conference keying protocol proposed by Burmester–Desmedt possesses this property [14].

4.3. *The Textbook ElGamal Encryption Scheme*

We briefly introduce the textbook ElGamal encryption scheme in a generalized form: Let G be an abelian group of prime order p and g a generator of G. Suppose that the secret key and the public key of the recipient are $x \in \mathbb{Z}_p$ and g^x, respectively. In order to encrypt a message $m \in G$, a sender selects a random $k \in \mathbb{Z}_p$ and sends a ciphertext $(c_1, c_2) := (g^k, m(g^x)^k)$ to the recipient. The recipient recovers the message m by computing c_2/c_1^x.

The textbook ElGamal encryption scheme is not secure in many situations. If it is used to encrypt a short message, for example, a secret key for symmetric ciphers, the message can be recovered easily [6]. Also it is vulnerable to chosen ciphertext attacks (Refer to the appendix in [1]). That is, for a given decryption oracle, any target ciphertext can be decrypted without feeding itself to the decryption oracle. Here we show that the decryption oracle enables not only a decryption of any target ciphertext without the secret key but also a reduction of the complexity to compute the secret key in some cases.

As mentioned in the previous subsection, first, a chosen ciphertext attacker $\mathcal{A}$ selects random numbers $k_1, k_2 \in \mathbb{Z}_p$, requests a decryption of the ciphertext $(c_1, c_2) := (y^k, y^{k'})$ to the decryption oracle, and obtains $c_2/c_1^x = g^{xk'} \cdot g^{x^2 k}$. Since $\mathcal{A}$ knows k, k', and g^x, he can compute $g_2 := g^{x^2}$. By considering different random pairs (k, k') and replacing y by g_2, $\mathcal{A}$ can obtain $g_3 := g^{x^3}$. By repeating this procedure ℓ times, $\mathcal{A}$ can obtain $g_1, g_2, \ldots, g_\ell$ $(g_i := g^{x^i})$ when ℓ decryption queries are allowed. By Theorems 1 and 2, if $p-1$ has a divisor $d \le \min\{\ell, p^{1/2}\}$ or $p+1$ has a divisor $d \le \min\{\ell/2, p^{1/3}\}$, the secret key x can be computed in $O(\sqrt{p/d})$ group exponentiations.

Our attack gives yet another reason why the textbook ElGamal is not secure. Furthermore, although it is not so likely in practice, it gives a good lesson on the risk of sharing the secret keys: if one uses the textbook ElGamal encryption scheme along with another cryptosystem having the same secret key, the secret key can be revealed from the textbook ElGamal encryption scheme, and hence the other system can also be insecure. For more applications of our attacks, see Brown and Gallant [13].

5. Practicality of the Proposed Algorithm

In this section, we discuss the potential of the proposed algorithms. The algorithm in Theorem 1 has complexity $O(\log p \cdot (\sqrt{(p-1)/d} + \sqrt{d}))$ for a divisor d of $p-1$. The complexity achieves the minimum value $O(\log p \cdot p^{1/4})$ when $d = O(p^{1/2})$. The algorithm in Theorem 2 has complexity $O(\log p \cdot (\sqrt{(p-1)/d} + d))$ for a divisor d of $p+1$. The complexity achieves the minimum value $O(\log p \cdot p^{1/3})$ when $d = O(p^{1/3})$. Hence the d-SDH problem on an abelian group of order p can be solved up to $O(\log p \cdot p^{1/4})$ (resp. $O(\log p \cdot p^{1/3})$) times faster than with generic DLP algorithms for large ℓ if $p-1$ (resp. $p+1$) has a divisor $d = O(p^{1/2})$ (resp. $d = O(p^{1/3})$).

Now we give an example in which the security reduction of the base problem due to our algorithm yields an attack of the system. We remark that not all the cryptosystems based on d-SDH problems are vulnerable to our attacks. For example, we do not know how to obtain several powers y^{α^i} of some group element y from the Boneh–Boyen ID-based encryption scheme [3].

Example 1. We consider a situation in which $E^+(\mathbb{F}_{3^{97}})$ [8] is used for the broadcast encryption scheme [10]. $E^+(\mathbb{F}_{3^{97}})$ has a subgroup G with a 151-bit prime order p. Let g be a generator of G, and $\alpha \in \mathbb{Z}_p$ be the system secret key. The scheme, assuming n users, publishes g and $g_i := g^{\alpha^i}$ for $0 \le i \le 2n$, $i \ne n$. By using a nondegenerate bilinear map e on G, we can compute $e(g, g)^{\alpha^i}$ for all nonnegative integers $i \le 4n$. By using Pollard's ρ algorithm [30,41], the secret key can be obtained in $O(2^{76})$ group operations. However, if we apply the proposed algorithm, it is reduced to approximately $O(2^{59})$ exponentiations or $O(2^{67})$ group operations for $n = 2^{32}$. Furthermore, $n = 2^{64}$, as suggested in the file sharing application [10], though it is not so realistic in practice, can reduce the complexity to $O(2^{42})$ exponentiations or $O(2^{50})$ group operations.

We remark that in order to give 2^{80} security for the system with 2^{64} users, it is recommended to consider the group with approximately a 220-bit prime order, unless p is of a special form.

5.1. *Practical Parameters*

Most cryptosystems based on SDH-related problems utilize bilinear maps. For practice, we investigate some known elliptic curve parameters and show that either $p - 1$ or $p + 1$ has many small divisors for the largest prime divisor p of the order for each elliptic curve in [8,23,32,34].

NIST Curves NIST suggested several elliptic curves for federal government use [32]. They consist of three categories: Pseudo-random curves over a prime field, a pseudo-random curve over a binary field, and a Koblitz curve over a binary field. For all the curves, the largest prime divisor p has the property that either $p - 1$ or $p + 1$ has sufficient small divisors. We present some of the curves and the factorizations of their orders:

- B-163: $p - 1 = 2 \cdot 53 \cdot 383 \cdot 21179 \cdot$ (a 132 bit prime), which is a 163-bit integer.
- K-163: $p - 1 = 2^4 \cdot 43 \cdot 73 \cdot$ (a 16-bit prime) $\cdot$ (an 18-bit prime) $\cdot$ (a 112-bit prime), which is a 163-bit integer.
- P-192: $p - 1 = 2^4 \cdot 5 \cdot 2389 \cdot$ (an 83-bit prime) $\cdot$ (a 92-bit prime), which is a 192-bit integer.

For example, a security loss of the cryptosystem using P-192 is up to 8 bits when the parameter ℓ in the SDH problem is larger than 2^{16}.

Elliptic Curves with Embedding Degree 6 Boneh, Lynn, and Shacham suggested two families of elliptic curves with embedding degree 6 for short signature schemes [8]: $E^+ : y^2 = x^3 + 2x + 1$ and $E^- : y^2 = x^3 + 2x - 1$ over $\mathbb{F}_3$. We consider E^+ or E^- over $\mathbb{F}_{3^\lambda}$. We denote by p the largest prime factor of $E^{\pm}(\mathbb{F}_{3^\lambda})$.

- $E^+(\mathbb{F}_{3^{97}})$: $p - 1 = 2 \cdot 3^{49} \cdot 24127552321 \cdot 21523361 \cdot 76801$, which is a 151-bit integer.
- $E^+(\mathbb{F}_{3^{121}})$: $p - 1 = 2 \cdot 3 \cdot 11^2 \cdot 683 \cdot 6029 \cdot$ (a 123-bit prime), which is a 155-bit integer.

Koblitz–Menezes Curves Koblitz and Menezes [23] suggested seven supersingular elliptic curve parameters for pairing-based cryptography. If we denote by p the order of the group to be used in cryptosystems, either $p+1$ or $p-1$ has a divisor 2^i for $i \geq 60$ in all the cases except one. The exceptional case is for $p = 2^{160} + 2^3 - 1$. In this case, however, $p - 1 = 2 \cdot 29 \cdot 227 \cdot 27059 \cdot$ (a 37 bit prime) $\cdot$ (a 94-bit prime).

Elliptic Curves in MIRACL Library The MIRACL library [34] provides a sample parameter for pairing-friendly elliptic curves. The order of the group is $p = 2^{159} + 2^{17} + 1$. Then $p - 1$ has the following prime factorization: $p - 1 = 2^{17} \cdot 5 \cdot 569 \cdot$ (a 27-bit prime) $\cdot$ (a 32-bit prime) $\cdot$ (a 32-bit prime) $\cdot$ (a 39-bit prime).

It is clear that our algorithm can be applied to all the examples above. We note that our algorithm may be more plausible for pairing-friendly curves including Koblitz–Menezes curves and MIRACL library curves, because these curves need to have some special properties, such as an order of small Hamming weights or a small "T" value for efficient implementation of Weil or Tate pairing, and so it is more difficult to have the order resistant against our attack.

5.2. *Distribution of Primes Susceptible to the Proposed Attacks*

Let $H(x, y, z; A)$ be the number of integers $n \leq x$ in A having a divisor in $(y, z]$. Ford showed in [19] that for any fixed integers a, b, λ with $\lambda \neq 0$ and $0 \leq a < b < 1$,

$$H\left(x, x^a, x^b; P_\lambda\right) = \Omega\left(\frac{x}{\ln x}\right),$$

where $P_\lambda := \{p + \lambda \mid p \text{ is a prime}\}$. In other words, there are $\Omega(x/\ln x)$ primes $p \leq x$ such that $p + \lambda$ has a prime divisor in $(x^a, x^b]$. For example, we observe that there is a positive proportion of primes p such that $p - 1$ has a prime divisor in $(x^{1/3}, x^{1/2}]$ or $p + 1$ has a prime divisor in $(x^{1/4}, x^{1/3}]$.

Conversely, we may ask how many primes can resist against the proposed attack. As an application of the fundamental lemma of the combinatorial sieve, we show in the following proposition that for any $0 < \tau < 1$, the probability of a prime p such that $p - 1$ (resp. $p + 1$) has no divisor in $(e^{(\ln p)^\tau/3}, p^{1/3}]$ reduces to zero as p increases to infinity.

Proposition 1. *For any $\lambda \neq 0$ and $0 < \tau < 1$,*

$$\#\left\{\textit{prime } p \leq x \mid p - \lambda \textit{ has no divisor in } \left(e^{(\ln p)^\tau/3}, p^{1/3}\right]\right\} = \pi(x) \cdot O\left(\ln^{1-\tau} x\right).$$

This proposition arises from the fundamental lemma of the combinatorial sieve [40, Theorem 4, p. 60]. For the convenience of readers, we introduce it.

Lemma 3 [40, Theorem 4, p. 60]. *Let $\mathcal{A}$ be a finite set of integers, and let $\mathcal{P}$ be the set of prime numbers in $[\eta, \xi]$ with $2 \leq \eta \leq \xi$ and $\mathcal{P}(y)$ the product of all elements $\leq y$ in $\mathcal{P}$. Assume that there exists a nonnegative multiplicative function ω, some real number X, and positive constants κ, C such that*

(a) $R_d := |\{a \in \mathcal{A} | a \equiv 0 \bmod d\}| - X\omega(d)/d$,
(b) $\prod_{\eta \le p \le \xi}(1 - \frac{\omega(p)}{p})^{-1} < (\frac{\ln \xi}{\ln \eta})^{\kappa}(1 + \frac{C}{\ln \eta})$.

Then we have, uniformly for $\mathcal{A}$, X, y, *and* $u \ge 1$,

$$S(\mathcal{A}, \mathcal{P}, y) = X \prod_{p \le y, p \in \mathcal{P}} \left(1 - \frac{\omega(p)}{p}\right)\left(1 + O\left(u^{-u/2}\right)\right) + O\left(\sum_{d \le y^u, d | \mathcal{P}(y)} |R_d|\right),$$

where $S(\mathcal{A}, \mathcal{P}, y) = |\{a \in \mathcal{A} | \gcd(a, \mathcal{P}(y)) = 1\}|$.

Proof of Proposition 1. Let $\omega(d) = d/\phi(d)$, $2 \le \eta^2 \le \xi$, and $\kappa > 2$. Then if ξ is sufficiently large, we have

$$\begin{aligned}\prod_{\eta \le p \le \xi}\left(1 - \frac{\omega(p)}{p}\right)^{-1} &\le 2 \prod_{\eta \le p \le \xi}\left(1 - \frac{1}{p}\right)^{-1} \\ &\le 2\frac{\ln \xi}{\ln \eta}\left(1 + O\left(\frac{1}{\ln \eta}\right)\right)\left(1 + O\left(\frac{1}{\ln \xi}\right)\right)^{-1} \\ &\le \left(\frac{\ln \xi}{\ln \eta}\right)^{\kappa}\left(1 + O\left(\frac{1}{\ln \eta}\right)\right),\end{aligned}$$

where the second inequality comes from the Mertens formula [40, Theorem 11, p. 17].

Let x be a positive integer and $\epsilon > 0$. If we consider $\mathcal{A} = \{p + \lambda \mid p \text{ a prime} \le x\}$, $X = \pi(x)$, $y = x^{1/3-\epsilon}$, and $u = 1$, then the Bombieri–Vinogradov Theorem [40, p. 262] states that for any constant $B > 0$, we have

$$\sum_{d \le y, d | \mathcal{P}(y)} |R_d| \le \sum_{d \le y}\left|\pi(x, -\lambda, d) - \frac{\pi(x)}{\phi(d)}\right| = O\left(\frac{x}{(\ln x)^B}\right),$$

where $\pi(x, -\lambda, d) = \{p \le x | p \text{ is a prime}, p \equiv -\lambda \bmod d\}$.

If we consider $\eta = e^{(\ln x)^{\tau}/3}$ $(0 < \tau < 1)$, $\xi = y = x^{1/3-\epsilon}$, and $B = 2$ in Lemma 3, then we have

$$S\left(\mathcal{A}, \mathcal{P}, x^{1/3-\epsilon}\right)/\pi(x) = O\left(\frac{\ln \eta}{\ln \xi}\right) + O\left(\frac{1}{\ln^{B-1} x}\right) = O\left(\ln^{\tau-1} x\right),$$

where $S(\mathcal{A}, \mathcal{P}, x^{1/3-\epsilon})$ is the number of primes $p \le x$ such that $p + \lambda$ has no divisor in $(e^{(\ln x)^{\tau}/3}, x^{1/3-\epsilon}]$.

Note that when $(e^{(\ln x)^{\tau}/3}, x^{1/3-\epsilon}] \subset (e^{(\ln p)^{\tau}/3}, p^{1/3}]$, $p + \lambda$ has no divisor in $(e^{(\ln x)^{\tau}/3}, x^{1/3-\epsilon}]$ if it has no divisor in $(e^{(\ln p)^{\tau}/3}, p^{1/3}]$. Hence, unless $p > x$ or $p < x^{1-3\epsilon}$, a prime $p \le x$ such that $p + \lambda$ has no divisor in $(e^{(\ln p)^{\tau}/3}, p^{1/3}]$ should be in $S(\mathcal{A}, \mathcal{P}, x^{1/3-\epsilon})$. Hence the number of primes $p \le x$ such that $p + \lambda$ has no divisor in $(e^{(\ln p)^{\tau}/3}, p^{1/3}]$ is

$$O\left(\pi(x) \ln^{\tau-1} x\right) + O\left(\pi\left(x^{1-3\epsilon}\right)\right) = \left(\pi(x) \ln^{\tau-1} x\right),$$

which proves the proposition. □

5.3. *Construction of Primes Resistant Against the Proposed Attack*

It appears difficult to find a prime p such that both of $p-1$ and $p+1$ have no small divisor greater than $(\log p)^2$ (more precisely, than $e^{(\ln p)^\tau/3}$ for any $0<\tau<1$). The prime number theorem [37] states that the probability that a positive integer p is a prime is $O(1/\log p)$ when p is sufficiently large. Given two small integers n_1, n_2, let us consider a positive integer p that satisfies three conditions: (1) p is a prime, (2) $(p-1)/n_1$ is a prime, and (3) $(p+1)/n_2$ is a prime. If the three conditions are independent, we may expect that the probability is approximately $O(1/\log^3 p)$. However, we do not know if the three conditions are independent, and hence obtaining a distribution of primes resistant to the proposed attack must be a hard problem. However, parameters of practical sizes can be generated by random selection and testing. Some sample parameters resistant to the proposed algorithms can be found in [39].

We may consider Gordon's algorithm [21] to generate primes that can resist against the proposed algorithms. Basically, the algorithm is to determine a prime of the form $p = 2(p_1^{p_2-2} \bmod p_2)p_1 - 1 + p_1 p_2 k$, where p_1 and p_2 are primes of equal size, and k is an integer. Then, we have $p_1 | p+1$ and $p_2 | p-1$. However, this algorithm usually yields a prime p significantly larger than p_1 and p_2, and so there can be other small divisors of $p-1$ and $p+1$. Hence this algorithm is not useful for our context.

6. Conclusion and Further Studies

In this paper, we proposed a novel algorithm to solve the DLP with auxiliary inputs more efficiently: For given element g of prime order p in an abelian group and g^{α^i} $(0 \le i \le \ell)$, the complexity to recover $\alpha \in \mathbb{Z}_p$ can be reduced by a factor of $O(\sqrt{d}/\log p)$ as compared to that of the DLP, where d is the largest divisor of $p-1$ not exceeding $\min\{\ell, p^{1/2}\}$ or the largest divisor of $p+1$ not exceeding $\min\{\ell/2, p^{1/3}\}$. This algorithm can be used to attack any cryptographic scheme that admits an oracle returning a power of its secret key upon an arbitrary input.

Hence, if a cryptographic scheme or protocol is based on a variant of ℓ-SDH problems or allows such an oracle by ℓ times, it is recommended to increase the key size by $\log_2 \ell$-bit, or use a prime p such that both of $p+1$ and $p-1$ have no small divisor greater than $(\log p)^2$, or more generally to rely on the generic lower bounds as they apply for the envisioned application to determine a safe key size.

It would be interesting to find an efficient generalization of the proposed algorithms to a group of arbitrary prime order p, using $\mathrm{GL}(n, \mathbb{F}_p)$, an extension field of $\mathbb{F}_p$ or an (hyper-)elliptic curve over $\mathbb{F}_p$ as in [25,33].

Acknowledgement

The author expresses his gratitude to Igor Shparlinski for informing him how to get the distribution of primes susceptible to the proposed attacks and to Brent Waters for pointing out a mistake in the definition of the Strong Diffie–Hellman problem in the preliminary version. He is also very grateful to Phong Nguyen and anonymous reviewers for their sincere reviews and helpful suggestions. This work was supported by the KOSEF grant funded by the Korea government (MEST) (No. R01-2008-000-11287-0, No. 20090058574).

References

[1] M. Abdalla, M. Bellare, P. Rogaway, DHAES: An encryption scheme based on Diffie–Hellman problem. IEEE P1363a Submission (1998). Available at http://grouper.ieee.org/groups/1363/addendum.html

[2] E. Bach, Explicit bounds for primality testing and related problems. *Math. Comput.* **55**, 355–380 (1990)

[3] D. Boneh, X. Boyen, Efficient selective-ID secure identity-based encryption without random oracles, in *Proceedings of Eurocrypt 2004*, LNCS, vol. 3027 (Springer, Berlin, 2004), pp. 223–238

[4] D. Boneh, X. Boyen, Short signatures without random oracles, in *Proceedings of Eurocrypt 2004*, LNCS, vol. 3027 (Springer, Berlin, 2004), pp. 56–73

[5] D. Boneh, X. Boyen, Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptol.* **21**(3), 149–177 (2008)

[6] D. Boneh, A. Joux, P. Nguyen, Why textbook ElGamal and RSA encryption are insecure, in *Proceedings of Asiacrypt 2000*, LNCS, vol. 1976 (Springer, Berlin, 2000), pp. 30–43

[7] D. Boneh, X. Boyen, H. Shacham, Short group signatures, in *Proceedings of Crypto 2004*, LNCS, vol. 3152 (Springer, Berlin, 2004), pp. 41–55

[8] D. Boneh, B. Lynn, H. Shacham, Short signatures from the Weil pairing. *J. Cryptol.* **17**(4), 297–319 (2004). Extended abstract in proceedings of Asiacrypt 2001, LNCS, vol. 2248 (Springer, Berlin, 2001), pp. 514–532

[9] D. Boneh, X. Boyen, E. Goh, Hierarchical identity based encryption with constant size ciphertext, in *Proceedings of Eurocrypt 2005*, LNCS, vol. 3494 (Springer, Berlin, 2005), pp. 440–456. A full paper is available in http://crypto.stanford.edu/~dabo/papers/shibe.pdf

[10] D. Boneh, C. Gentry, B. Waters, Collution resistant broadcast encryption with short ciphertexts and private keys, in *Proceedings of Crypto 2005*, LNCS, vol. 3621 (Springer, Berlin, 2005), pp. 258–275

[11] A. Boldyreva, Threshold signatures, multisignatures and blind signatures based on the Gap–Diffie–Hellman-group signature scheme, in *Proceedings of Public Key Cryptography 2003*, LNCS, vol. 2567 (Springer, Berlin, 2003), pp. 31–46

[12] X. Boyen, The uber-assumption family—a unified complexity framework for bilinear groups, in *Proceedings of Pairing 2008*, LNCS, vol. 5209 (Springer, Berlin, 2008), pp. 39–56

[13] D. Brown, R. Gallant, The static Diffie–Hellman problem. Available in http://eprint.iacr.org/2004/306

[14] M. Burmester, Y. Desmedt, A secure and efficient conference key distribution system (Extended Abstract), in *Proceedings of Eurocrypt 1994*, LNCS, vol. 950 (Springer, Berlin, 1994), pp. 275–286

[15] J. Cheon, Security analysis of the strong Diffie–Hellman problem, in *Proceedings of Eurocrypt 2006*, LNCS, vol. 4004 (Springer, Berlin, 2006), pp. 1–11

[16] B. den Boer, Diffie–Hellman is as strong as discrete log for certain primes, in *Proceedings of Crypto '88*, LNCS, vol. 403 (Springer, Berlin, 1989), pp. 530–539

[17] Y. Dodis, A. Yampolskiy, A verifiable random function with short proofs and keys, in *Proceedings of Public Key Cryptography 2005*, LNCS, vol. 3386 (Springer, Berlin, 2005), pp. 416–431

[18] T. Elgamal, A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory* **31**(4), 469–472 (1985)

[19] K. Ford, The distribution of integers with a divisor in a given interval. *Ann. Math.* (2008, to appear)

[20] S. Galbraith, K. Paterson, N. Smart, Pairings for cryptographers. *Discrete Appl. Math.* **156**(16), 3113–3121 (2008)

[21] J. Gordon, Strong primes are easy to find, in *Proceedings of Eurocrypt '84* (Springer, Berlin, 1984), pp. 216–223

[22] D. Jao, K. Yoshida, Boneh–Boyen signatures and the strong Diffie-Hellman problem, in *Proceedings of Pairing* (2009, to appear)

[23] N. Koblitz, A. Menezes, Pairing-based cryptography at high security levels, in *Proceedings of IMA Conference of Cryptography and Coding 2005*, pp. 13–36

[24] S. Kozaki, T. Kutsuma, K. Matsuo, Remarks on Cheon's algorithms for pairing-related problems, in *Proceedings of Pairing 2007*, LNCS, vol. 4575 (Springer, Berlin, 2007), pp. 302–316

[25] U. Maurer, S. Wolf, The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms. *SIAM J. Comput.* **28**(5), 1689–1721 (1999)

[26] A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography* (CRC Press, Boca Raton, 1996)

[27] S. Mitsunari, R. Sakai, M. Kasahara, A new traitor tracing. *IEICE Trans. Fundam.* **E85-A**(2), 481–484 (2002)
[28] V. Nechaev, Complexity of a deterministic algorithm for the discrete logarithm. *Math. Zamet.* **55**, 91–101 (1994). English translation in *Math. Notes* **55**(2), 165–172 (1994)
[29] T. Okamoto, Efficient blind and partially blind signatures without random oracles, in *Proceedings in TCC 2006*, LNCS, vol. 3876 (Springer, Berlin, 2006), pp. 80–99
[30] J. Pollard, Monte Carlo methods for index computation (mod p). *Math. Comput.* **32**, 918–924 (1978)
[31] J. Pollard, Kangaroos, monopoly and discrete logarithms. *J. Cryptol.* **13**(4), 437–447 (2000)
[32] *Recommended Elliptic Curves for Federal Government Use*, Available at http://csrc.nist.gov/CryptoToolkit/dss/ecdsa/NISTReCur.pdf, 1999
[33] T. Satoh, On generalization of Cheon's algorithms. Preprint, 2008
[34] M. Scott, *Multiprecision Integer and Rational Arithmetic C/C++ Library*. Available at http://indigo.ie/~mscott/
[35] V. Shoup, Searching for primitive roots in finite fields. *Math. Comput.* **58**, 369–380 (1992)
[36] V. Shoup, Lower bounds for discrete logarithms and related problems, in *Proceedings of Eurocrypt '97*, LNCS, vol. 1233 (Springer, Berlin, 1997), pp. 256–66
[37] V. Shoup, *A Computational Introduction to Number Theory and Algebra* (Cambridge University Press, Cambridge, 2005)
[38] I. Shparlinski, On finding primitive roots in finite fields. *Theor. Comput. Sci.* **157**, 273–275 (1996)
[39] D. Sun, Elliptic curves with the minimized security loss of the strong Diffie–Hellman problem, Ph.D. Dissertation, Seoul National University, 2007. Available at http://library.snu.ac.kr/DetailView.jsp?uid=4&cid=2857710
[40] G. Tenenbaum, *Introduction to Analytic and Probabilistic Number Theory* (Cambridge University Press, Cambridge, 1995)
[41] E. Teske, Speeding up Pollard's rho method for computing discrete logarithms, in *Proceedings of Algorithmic Number Theory Symposium III*, LNCS, vol. 1423 (Springer, Berlin, 1998), pp. 541–554
[42] Y. Wang, On the least primitive root of a prime. *Sci. Sin.* **10**(1), 1–14 (1961)
[43] K. Yoshida, Boneh–Boyen signatures and the strong Diffie–Hellman problem, Master Thesis, University of Waterloo, 2009. Available at http://uwspace.uwaterloo.ca/handle/10012/4219