Security Analysis of the Strong Diffie-Hellman Problem

Jung Hee Cheon

ISaC and Dept. of Mathematics, Seoul National University, Republic of Korea
jhcheon@snu.ac.kr, http://www.math.snu.ac.kr/~jhcheon

Abstract. Let g be an element of prime order p in an abelian group and $\alpha \in \mathbb{Z}_p$. We show that if g, g^{α}, and g^{α^d} are given for a positive divisor d of $p-1$, we can compute the secret α in $O(\log p \cdot (\sqrt{p/d}+\sqrt{d}))$ group operations using $O(\max\{\sqrt{p/d}, \sqrt{d}\})$ memory. If g^{α^i} $(i = 0, 1, 2, \ldots, d)$ are provided for a positive divisor d of $p + 1$, α can be computed in $O(\log p \cdot (\sqrt{p/d} + d))$ group operations using $O(\max\{\sqrt{p/d}, \sqrt{d}\})$ memory. This implies that the strong Diffie-Hellman problem and its related problems have computational complexity reduced by $O(\sqrt{d})$ from that of the discrete logarithm problem for such primes.

Further we apply this algorithm to the schemes based on the Diffie-Hellman problem on an abelian group of prime order p. As a result, we reduce the complexity of recovering the secret key from $O(\sqrt{p})$ to $O(\sqrt{p/d})$ for Boldyreva's blind signature and the original ElGamal scheme when $p - 1$ (resp. $p + 1$) has a divisor $d \leq p^{1/2}$ (resp. $d \leq p^{1/3}$) and d signature or decryption queries are allowed.

Keywords: Discrete logarithm, Diffie-Hellman, strong Diffie-Hellman, ElGamal encryption, blind signature

1 Introduction

Let g be an element of prime order p in an abelian group and $\alpha \in \mathbb{Z}_p$. The ℓ-Strong Diffie-Hellman (ℓ-SDH) problem asks to find $g^{\alpha^{\ell+1}}$ given $g, g^{\alpha}, \ldots, g^{\alpha^{\ell}}$. Recently, many cryptographic schemes including encryption, signature, and key management schemes are proposed on the basis of the Strong Diffie-Hellman (SDH) problem [MSK02,BB04e,BB04s], or its variants such as the Bilinear Diffie-Hellman problem [BBS04,DY05] and the Bilinear Diffie-Hellman Exponent (BDHE) problem [BBG05,BGW05]. A lower bound on the computational complexity of the SDH problem or its variants for generic groups are known in the sense of Shoup [Sho97], but it does not guarantee the security for specific parameters.

In this paper, we analyze the security of the SDH problem. More precisely, we show that if g, g^{α} and g^{α^d} are given for a positive divisor d of $p - 1$, the secret $\alpha \in \mathbb{Z}_p$ can be computed in $O(\log p \cdot (\sqrt{p/d} + \sqrt{d}))$ group operations using $O(\max\{\sqrt{p/d}, \sqrt{d}\})$ memory. If g^{α^i} $(i = 0, 1, 2, \ldots, d)$ are provided for a positive divisor d of $p+1$, it can be computed in $O(\log p \cdot (\sqrt{p/d}+d))$ group operations using the same size of memory. This implies that the strong Diffie-Hellman problem and its related problems have computational complexity reduced by $O(\sqrt{d})$ from that of the discrete logarithm problem for such primes. Hence it is necessary to increase by the size of d the key size of the cryptographic schemes based on the ℓ-SDH problem or its variants if the base group has such a prime as its order.

We investigate some known elliptic curve parameters and find that either $p-1$ or $p+1$ has many small divisors for the largest prime divisor p of its order for each elliptic curve in [NIST,BLS01,KM05,MIRACL]. For example, if we use the curve E^+ over $GF(3^{155})$ [BLS01] for the broadcast encryption [BGW05], the secret key can be computed in $O(2^{59})$ exponentiations (resp. $O(2^{42})$ exponentiations) when the number of users is 2^{32} (resp. 2^{64}), rather than $O(2^{76})$ group operations.

Moreover, we apply this algorithm to the schemes based on the Diffie-Hellman problem on an abelian group of prime order p. As a result, we show the complexity of recovering the secret key is reduced from $O(\sqrt{p})$ to $O(\sqrt{p/d})$ for Boldyreva's blind signatures [Bol03] when d signature or decryption queries are allowed and $p-1$ has a divisor $d \le p^{1/2}$ or $p+1$ has a divisor $d \le p^{1/3}$. Similar results hold for the original ElGamal scheme [ElG85] with decryption oracles and the conference keying protocol by Burmester-Desmedt [BD94] with key issuing oracles.

The rest of the paper is organized as follows: In Section 2, we introduce the SDH related problems and some schemes based on them. In Section 3, we present our algorithms. In Section 4, we exploit our algorithms to attack several protocols based on the Diffie-Hellman problem. In Section 5, we investigate some known elliptic curve parameters in order to check if our algorithms are applicable for these parameters. We conclude in Section 6.

2 Strong Diffie-Hellman Problems and Their Variants

Let G be an abelian group of prime order p and g a generator of G. The **Discrete Logarithm (DL) Problem** in G asks to find $a \in \mathbb{Z}_p$ given g and g^a in G. Many cryptosystems are designed on the basis of the DL problem, but most of them have the security equivalent to a weaker variant of the DL problem rather than the DL problem itself. Two most important weaker variants are as follows:

The Computation Diffie-Hellman (CDH) Problem. Given (g, g^a, g^b), compute g^{ab}.
The Decisional Diffie-Hellman (DDH) Problem. Given (g, g^a, g^b, g^c), decide whether $c = ab$ in $\mathbb{Z}_p$.

Recently, some weakened variants of the CDH problem are introduced and being used to construct cryptosystems for various functionalities or security without random oracles. One characteristic of these problems is to disclose $g, g^{\alpha}, \ldots, g^{\alpha^{\ell}}$ for the secret α and some integer ℓ.

The ℓ-weak Diffie-Hellman (ℓ-wDH) Problem. Given g and g^{α^i} in G for $i = 1, 2, \ldots, \ell$, compute $g^{1/\alpha}$. This problem was introduced by Mitsunari, Sakai, and Kasahara for a traitor tracing scheme [MSK02].

The ℓ-Strong Diffie-Hellman (ℓ-SDH) Problem. Given g and g^{α^i} in G for $i = 1, 2, \ldots, \ell$, compute $g^{\alpha^{\ell+1}}$. This problem is considered as a weaker version of ℓ-wDH problem. It was first introduced by Boneh and Boyen to construct a short signature scheme, that is provably secure in the standard model (without random oracles) [BB04s], and later a short group signature scheme [BBS04].

The SDH problem is generalized into a group with bilinear maps. We further assume that $e : G \times G \to G'$ is an admissible bilinear map between two abelian groups G and G' with prime order p.

The ℓ-Bilinear Diffie-Hellman Inversion (ℓ-BDHI) Problem. Given g and g^{α^i} in G for $i = 1, 2, \ldots, \ell$, compute $e(g, g)^{1/\alpha} \in G'$. This problem was introduced by Boneh and Boyen to construct an identity-based encryption that is secure in the standard model [BB04e]. It is also used to construct verifiable random functions [DY05].

The ℓ-Bilinear Diffie-Hellman Exponent (ℓ-BDHE) Problem. Given g, h, and g^{α^i} $(i = 1, 2, \ldots, \ell-1, \ell+1, \ldots, 2\ell)$ in G, compute $e(g, h)^{\alpha^\ell} \in G'$. This problem was introduced by Boneh, Boyen, and Goh [BBG05] to construct a hierarchical identity-based encryption scheme with constant size ciphertext, and later used for a public key broadcast encryption scheme with constant size transmission overhead [BGW05].

Given two problem instances A and B, we denote by $A \geq B$ if the problem B can be solved in polynomial time with polynomially many queries to the oracle to solve the problem A. Then we can easily deduce the following relations among the DL related problems [BBG05]:

$$\text{DL} \geq \text{CDH} \geq \text{DDH} \geq \ell\text{-wDH} \geq \ell\text{-SDH} \geq \ell\text{-BDHI},\ (\ell+1)\text{-BDHE}.$$

3 Main Results

Theorem 1. *Let g be an element of prime order p in an abelian group. 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 in $O(\log p \cdot (\sqrt{(p-1)/d} + \sqrt{d}))$ group operations using $O(\max\{\sqrt{(p-1)/d}, \sqrt{d}\})$ memory.*

Proof. Note that $\mathbb{Z}_p^*$ is a cyclic group with $\phi(p-1)$ generators, where $\phi(\cdot)$ is the Euler totient function. Since a random element in $\mathbb{Z}_p^*$ is a generator with probability

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

which is large enough [MOV, p.162], we can easily take a generator of $\mathbb{Z}_p^*$. Let ζ_0 be a generator of $\mathbb{Z}_p^*$. Then we compute $\zeta = \zeta_0^d$ that is an element of order $(p-1)/d$ in $\mathbb{Z}_p^*$.

Since $(\alpha^d)^{(p-1)/d} = 1$ and ζ generates all $(p-1)/d$-th roots of unity in $\mathbb{Z}_p^*$, there exists a non-negative integer i less than $(p-1)/d$ such that $\alpha^d = \zeta^i$. If we take $d_1 = \lceil \sqrt{(p-1)/d} \rceil$, we must have

$$(\alpha^d)\zeta^{-u} = \zeta^{d_1 v}$$

for some $0 \le u, v < d_1$. It is equivalent to

$$g_d^{\zeta^{-u}} = g^{\zeta^{d_1 v}}. \tag{1}$$

We compute and store the left-hand side terms and compare them with each of right-hand side terms in Baby-Step Giant-Step style. Note that each of terms in both sides can be computed by repeated exponentiations by either ζ^{-1} or ζ^{d_1}. Thus we can find all non-negative integers u and v less than d_1 satisfying (1) in $O(d_1 \cdot \log p)$ group operations using $O(d_1)$ memory. For u and v which satisfies (1) and $u + d_1 v$ is smallest, we put $k_0 = u + d_1 v$. Then k_0 is a non-negative integer less than $(p-1)/d$.

Let $\alpha = \zeta_0^k$ for $0 \le k < p-1$. Then we have $dk \equiv dk_0 \mod (p-1)$ and so $k \equiv k_0 \mod (p-1)/d$. There exists a non-negative integer j less than d such that $k = k_0 + j(p-1)/d$. If we take $d_2 = \lceil \sqrt{d} \rceil$, we must have

$$\alpha \zeta_0^{-u'(p-1)/d} = \zeta_0^{k_0 + d_2 v'(p-1)/d}$$

for some $0 \le u', v' < d_2$. It is equivalent to

$$g_1^{\zeta_0^{-u'(p-1)/d}} = g^{\zeta_0^{k_0 + d_2 v'(p-1)/d}}. \tag{2}$$

By the same method as above, we can find non-negative integers u' and v' less than d_2 satisfying (2) in $O(d_2 \cdot \log p)$ group operations and $O(d_2)$ memory. This completes the proof. □

We remark that the memory requirement of the above algorithm can be reduced by using Pollard's lambda techniques [Pol78]. We use the notation of Theorem 1 to sketch the idea: First we consider a function $F : \mathbb{Z}_p \to \mathbb{Z}_p$ with $F(x) = x\zeta^{f(g^x)}$ for a pseudo-random function $f : \langle g \rangle \to \mathbb{Z}_{(p-1)/d}$. For $\beta \in \mathbb{Z}_p$ and $t \ge 1$, $g^{F^t(\beta)}$ can be computed from g and g^β in $O(t \log p)$ group operations by using

$$g^{F(\beta)} = \left(g^\beta\right)^{\zeta^{f(g^\beta)}} \text{ and } g^{F^i(\beta)} = \left(g^{F^{i-1}(\beta)}\right)^{\zeta^{f(g^{F^{i-1}(\beta)})}} \text{ if } i \ge 2.$$

If we find u, v such that $g^{F^u(\alpha^d)} = g^{F^v(1)}$, we have $F^u(\alpha^d) = F^v(1)$ in $\mathbb{Z}_p$ and so

$$\alpha^d \zeta^{\sum_{i=1}^u f(g^{F^{i-1}(\beta)})} = \zeta^{\sum_{j=1}^v f(g^{F^{i-1}(1)})}.$$

Hence if we store only distinguished points [Tes98], α^d can be computed in $O(\sqrt{(p-1)/d})$ exponentiations using small memory with some probability. The second part to compute α from g^α and α^d can be done using similar technique.

If we know $g^{\alpha^{(p-1)/d}}$ for many small d, we can do even better:

Corollary 1. *Let g be an element of prime order p in an abelian group. Suppose that $p-1 = d_1 d_2 \cdots d_t$ for pairwise 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 $O(\log p \cdot \sum_{i=1}^{t} \sqrt{d_i})$ group operations using $O(\max_{1\le i\le t} \sqrt{d_i})$ memory.*

Proof. Let ζ be a generator of $\mathbb{Z}_p^*$ and $\alpha = \zeta^k$. Since $(\alpha^{(p-1)/d_i})^{d_i} = 1$, there must be a non-negative integer k_i less than d_i satisfying $\alpha^{(p-1)/d_i} = (\zeta^{(p-1)/d_i})^{k_i}$. Hence by checking

$$g_{(p-1)/d_i} = g^{(\zeta^{(p-1)/d_i})^{k_i}} \text{ for } 0 \le k_i < d_i$$

or

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

we can compute k_i in $O(\log p \cdot \sqrt{d_i})$ group operations using $O(\sqrt{d_i})$ memory. 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 using Chinese Remainder Theorem. The total complexity is $O(\log p \cdot \sum_{i=1}^{t} \sqrt{d_i})$ using $O(\max_{1\le i\le t} \sqrt{d_i})$ memory. □

Next, we use an imbedding of $\mathbb{Z}_p$ into $\mathbb{F}_{p^2}$ to generalize Theorem 1.

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

Proof. Let a be a quadratic non-residue in $\mathbb{Z}_p$ and θ be a root of $x^2 = a$ in an algebraically closed field of $\mathbb{Z}_p$. Then $\mathbb{Z}_p[\theta] \cong \mathbb{F}_{p^2}$. Let H be a subgroup of order $p+1$ of $\mathbb{F}_{p^2}$. Since $\beta \in H$ is equivalent to $\beta^{p+1} = 1$, we see that $\beta_0 + \beta_1\theta$ is an element of H for $\beta_0 = (1 + a\alpha^2)/(1 - a\alpha^2)$ and $\beta_1 = 2\alpha/(1 - a\alpha^2)$ from $\theta^p = -\theta$ and

$$\beta^{p+1} = (\beta_0 + \beta_1\theta)(\beta_0 + \beta_1\theta^p) = \beta_0^2 - a\beta_1^2. \tag{3}$$

Let ζ_0 be a generator of H (for example, the $(p+1)$-th power of a generator of $\mathbb{F}_{p^2}^*$). Then $\zeta := \zeta_0^d$ generates all the $(p+1)/d$-th roots of unity and so there must be some $k \in \mathbb{Z}$ such that $\beta^d = \zeta^k$ and $0 \le k < (p+1)/d$. For convenience, we denote $\zeta^i = s_i + t_i\theta$ for some $s_i, t_i \in \mathbb{Z}_p$ where the index i is defined modulo $(p+1)/d$. Also we denote

$$\beta^d = (\beta_0 + \beta_1\theta)^d = \frac{1}{(1 - a\alpha^2)^d}(f_0(\alpha) + f_1(\alpha)\theta),$$

where f_i's are polynomials of degree $2d$. Then we must have

$$\beta^d \zeta^{-u} = \zeta^{d_1 v} \tag{4}$$

for some $0 \le u, v < d_1 := \lceil \sqrt{(p+1)/d} \rceil$. It is equivalent to

$$(f_0(\alpha)s_{-u} + af_1(\alpha)t_{-u}) + (f_0(\alpha)t_{-u} + f_1(\alpha)s_{-u})\theta = (1 - a\alpha^2)^d(s_{d_1 v} + t_{d_1 v}\theta). \tag{5}$$

Hence we compute $(g^{f_0(\alpha)s_{-u} + af_1(\alpha)t_{-u}}, g^{f_0(\alpha)t_{-u} + f_1(\alpha)s_{-u}})$ for all $0 \le u < d_1$ and store them. By comparing them with $(g^{(1-a\alpha^2)^d s_{d_1 v}}, g^{(1-a\alpha^2)^d t_{d_1 v}})$ for each $0 \le v <$

d_1, we can find the (unique) non-negative integers u and v less than d_1 satisfying (4) and $u+d_1v < (p+1)/d$. We put $k_0 = u+d_1v$. Note that $g^{f_0(\alpha)}, g^{f_1(\alpha)}$ and $g^{(1-a\alpha^2)^d}$ can be computed from $g, g_1, \ldots, g_{2d}$ in $6d$ exponentiations. Hence k_0 can be found in $O(\log p \cdot (6d + \sqrt{(p+1)/d}))$ group operations with $O(\sqrt{(p+1)/d})$ memory.

Let $\beta = \zeta_0^k$ for $0 \le k < p+1$. Then we have $k \equiv k_0 \mod (p+1)/d$. There exists a non-negative integer j less than d such that $k = k_0 + j(p+1)/d$. If we take $d_2 = \lceil \sqrt{d} \rceil$, there must exist non-negative integers u', v' less than d_2 such that

$$\beta \zeta_0^{-u'(p+1)/d} = \zeta_0^{k_0 + d_2 v'(p+1)/d}. \tag{6}$$

We denote $\zeta_0^{-i(p+1)/d} = s'_i + t'_i\theta$ and $\zeta_0^{k_0+d_2i(p+1)/d} = s''_i + t''_i$ for some $s'_i, t'_i, s''_i, t''_i \in \mathbb{Z}_p$ where the index i is defined modulo $(p+1)$. Then (6) is equivalent to

$$((1+a\alpha^2)s_{u'} + 2a\alpha t_{u'}) + ((1+a\alpha^2)t_{u'} + 2\alpha s_{u'})\theta = (1-a\alpha^2)(s_{v'} + t_{v'}\theta). \tag{7}$$

Hence we compute $(g^{(1+a\alpha^2)s_{u'}+2a\alpha t_{u'}}, g^{(1+a\alpha^2)t_{u'}+2\alpha s_{u'}})$ for all $0 \le u' < d_2$ and store them. By comparing them with $(g^{(1-a\alpha^2)s_{v'}}, g^{(1-a\alpha^2)t_{v'}})$ for each $0 \le v' < d_2$, we can find non-negative integers u' and v' satisfying (6). That is, $\beta = \zeta_0^{k_0+(u'+d_2v')(p+1)/d}$ can be found in $O(\log p \cdot \sqrt{d})$ group operations and $O(\sqrt{d})$ memory. This completes the proof. □

We remark that if $d \le p^{1/3}$, then Theorem 2 says that the secret can be computed in $O(\log p \cdot \sqrt{p/d})$ group operations using $O(\sqrt{p/d})$ memory.

Remark 1. We may consider that our proof utilizes Diffie-Hellman oracles in a very restricted way [Boe88,MW99]. That is, in our situations we can use the Diffie-Hellman oracle $DH(g^x, g^y) = g^{xy}$ only when x is fixed and $y = x^\ell$ for some small ℓ. This restriction is an obstacle when we try to generalize the proposed algorithm into other extension fields of $\mathbb{F}_p$ or elliptic or hyperelliptic curves over $\mathbb{F}_p$.

4 Analysis of Cryptographic Schemes based on the Diffie-Hellman Problem

4.1 Blind Signature based on the GDH Assumption

The Gap-Diffie-Hellman (GDH) group is an abelian group on which there is an polynomial time algorithm to solve the decisional Diffie-Hellman problem and there is no polynomial time algorithm to solve the computation Diffie-Hellman problem.

Boldyreva proposed a blind signature scheme on a Gap-Diffie-Hellman group [Bol03]. 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 [BLS01]. 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 picks 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 [Bol03], 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 so 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 $g_2 \cdot g^{\gamma_1}$. 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 Theorem 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 $O(\sqrt{d})$ from that of the GDH assumption.

We note that the attack does not imply that the security proof of the scheme is wrong, but that more quantitative analysis on security reduction is required. In fact, the security proof of BLS signatures 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 [BLS01].

This method can be applied similarly to schemes which respond by its secret key power for an unknown message. For example, the conference keying protocol by Burmester-Desmedt has this property [BD94]. Thus, in this case, we need to take the order carefully or raise the security parameter.

4.2 Original ElGamal Encryption Scheme

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

The ElGamal encryption is known not to satisfy non-malleability under chosen ciphertext attacks (Refer to the appendix in [ABR98]). That is, given a 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 in the previous subsection, first a chosen ciphertext attacker $\mathcal{A}$ takes 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 he knows k, k' and g^x, $\mathcal{A}$ can compute $g_2 := g^{x^2}$. By taking different random pairs (k, k') and replacing y by g_2, $\mathcal{A}$ can obtain $g_3 := g^{x^3}$ similarly. 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

Theorem 1 and 2, if $p-1$ has a divisor $d \leq \min\{\ell, p^{1/2}\}$ or $p+1$ has a divisor $d \leq \min\{\ell/2, p^{1/3}\}$, the secret key x can be computed in $O(\sqrt{p/d})$.

We might imagine a situation that this attack is harmful: One uses the original ElGamal encryption scheme, to encrypt not so important messages, with another cryptosystem having the same secret key. Then the secret key may be revealed from the original ElGamal encryption scheme and so the other system can be insecure. This shows that the original ElGamal scheme must not share the same secret key with another system.

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 security of the ℓ-SDH problem on an abelian group of order p can be reduced up to $O(\log p \cdot p^{1/4})$ (resp. $O(\log p \cdot p^{1/3})$) 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 security reduction due to our algorithm yields a serious security problem.

Example 1. We consider the situation that $E^+(\mathbb{F}_{3^{97}})$ [BLS01] is used for the broadcast encryption scheme [BGW05]. $E^+(\mathbb{F}_{3^{97}})$ has a subgroup G of 151 bit prime order p. Let g be a generator of G and $\alpha \in \mathbb{Z}_p$ be the system secret. The scheme assuming n users publishes g and $g_i := g^{\alpha^i}$ for $0 \leq i \leq 2n$, $i \neq n$. Using a non-degenerate bilinear map e on G, we can compute $e(g,g)^{\alpha^i}$ for all non-negative integers $i \leq 4n$. Using Pollard ρ method [Pol78], the secret key can be found in $O(2^{76})$ group operations. But if we apply the proposed algorithm, it is reduced to about $O(2^{59})$ exponentiations or $O(2^{67})$ group operations for $n = 2^{32}$. Furthermore, if we use $n = 2^{64}$ as in the file sharing application [BGW05], the complexity is reduced 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 take the group of about 220 bit prime order unless p is of a special form.

Most cryptosystems based on SDH-related problems make use of 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 [NIST,BLS01,KM05,MIRACL].

NIST curves. NIST suggested several elliptic curves for federal government use [NIST]. 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 most of them, the largest prime divisor p has the property that either $p-1$ or $p+1$ has enough small divisors. We present some of them:

- 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.

We note that P-192 gives the smallest security loss, that is 8 bits, if the parameter ℓ in the SDH problem is less than 83 bits. Otherwise, however, the security loss for P-192 can be more than 40 bits.

Elliptic curves with embedding degree 6. Boneh, Lynn and Shacham suggested two families of elliptic curves with embedding degree 6 for short signatures [BLS01]: $E^+ : y^2 = x^3 + 2x + 1$ 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 [KM05] 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 divisor 2^i for $i \geq 60$ in all cases except one. The exceptional case is $p = 2^{160} + 2^3 - 1$. 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. MIRACL library [MIRACL] 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).

We can see that our algorithm can be applied for all the examples above. We note that our algorithm is more plausible for pairing-friendly curves including Koblitz-Menezes curves and MIRACL library curves because a curve with an order of small Hamming weights in signed binary form admits efficient implementation of Weil or Tate pairing. In most cases, however, it is necessary and seems hard to find a prime p such that both of $p-1$ and $p+1$ have no small divisor greater than $(\log p)^2$. We may consider Gordon's algorithm [Gor84] to generate strong primes which resist against the proposed algorithms. Basically, the algorithm is to find a prime of the form $p = 2(p_1^{p_2-2} \mod p_2)p_1 - 1 + p_1 p_2 k$ where p_1 and p_2 are primes of equal size and k is an integer. Then we have $p_1 | p+1$ and $p_2 | p-1$. But this algorithm usually yields a prime much larger than p_1 and p_2. It would be an interesting problem to find elliptic curve parameters for which the security loss of the SDH is minimized.

6 Conclusion and Further Studies

In this paper, we proposed a novel algorithm to solve the SDH-related problems. More precisely, given an element g of prime order p in an abelian group and a secret $\alpha \in \mathbb{Z}_p$, if g^{α^i} $(0 \le i \le \ell)$ are published for the secret α, the complexity to recover α can be reduced by a factor of $\sqrt{d}$ from that of the DLP, where d is the maximum of the largest divisor of $p-1$ not exceeding $\min\{\ell, p^{1/2}\}$ and the largest divisor of $p+1$ not exceeding $\min\{\ell/2, p^{1/3}\}$. This algorithm can be used to attack cryptographic schemes that admit an oracle to return its secret key power 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 or use a prime p such that both of $p+1$ and $p-1$ have no small divisor greater than $(\log p)^2$. However, we have no idea about the distribution of such primes.

We may try to generalize the proposed algorithms as in [MW99]. One problem is to find an embedding of $\mathbb{F}_p$ to some other groups including extension fields of $\mathbb{F}_p$ and elliptic or hyperelliptic curves over $\mathbb{F}_p$.

Acknowledgement I am grateful to Dong Hoon Lee and Taekyoung Kwon for helpful discussions and JaeHong Seo for his implementation. I would also like to thank the anonymous reviewers for their valuable suggestions.

References

[ABR98] M. Abdalla, M. Bellare, and 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`.

[BB04e] D. Boneh and X. Boyen,"Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles," Eurocrypt 2004, LNCS 3027, Springer-Verlag, pp. 223-238, 2004.

[BB04s] D. Boneh and X. Boyen, "Short Signatures Without Random Oracles," Eurocrypt 2004, LNCS 3027, Springer-Verlag, pp. 56-73, 2004.

[BBG05] D. Boneh, X. Boyen, and E. Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext," Eurocrypt 2005, LNCS 3494, Springer-Verlag, pp. 440-456, 2005.

[BBS04] D. Boneh, X. Boyen, and H. Shacham, "Short Group Signatures," Crypto 2004, LNCS 3152, Springer-Verlag, pp. 41-55, 2004.

[BD94] M. Burmester and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System (Extended Abstract)," Eurocrypt 1994, LNCS 950, Springer-Verlag, pp. 275-286, 1994.

[BGW05] D. Boneh, C. Gentry, and B. Waters. "Collution Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," Crypto 2005, LNCS 3621, Springer-Verlag, pp. 258-275, 2005.

[BLS01] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," J. of Cryptology, Vol. 17, No. 4, pp. 297-319, 2004. Extended abstract in proceedings of Asiacrypt '01, LNCS 2248, Springer-Verlag, pp. 514-532, 2001.

[Boe88] B. den Boer, "Diffie-Hellman is as Strong as Discrete Log for Certain Primes," Crypto '88, LNCS 403, Springer-Verlag, pp. 530-539, 1989.

[Bol03] A. Boldyreva, "Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme," Public Key Cryptography 2003, LNCS 2567, pp. 31-46, 2003.

[DY05] Y. Dodis and A. Yampolskiy, "A Verifiable Random Function with Short Proofs and Keys," Public Key Cryptography 2005, LNCS 3386, pp. 416-431, 2005.

[ElG85] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms," IEEE Transactions on Information Theory, Vol. 31, no 4, pp. 469-472, 1985.

[Gor84] J. Gordon, "Strong Primes are Easy to Find," Eurocrypt '84, LNCS 209, Springer-Verlag, pp. 216-223, 1984.

[KM05] N. Koblitz and A. Menezes, "Pairing-based Cryptography at High Security Levels," IMA Conference of Cryptography and Coding 2005, pp. 13-36, 2005.

[MIRACL] M. Scott, *Multiprecision Integer and Rational Arithmetic C/C++ Library*, Available at `http://indigo.ie/~mscott/`.

[MOV] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[MSK02] S. Mitsunari, R. Sakai, and M. Kasahara, "A New Traitor Tracing," IEICE Trans. Fundamentals, Vol. E85-A, no. 2, pp. 481-484, 2002.

[MW99] U. Maurer and S. Wolf, "The Relationship Between Breaking the Diffie-Hellman Protocol and Computing Discrete Logarithms," SIAM J. Comput., Vol. 28, no. 5, pp. 1689-1721, 1999.

[NIST] *Recommended Elliptic Curves for Federal Government Use*, Available at `http://csrc.nist.gov/CryptoToolkit/dss/ecdsa/NISTReCur.pdf`, 1999.

[Pol78] J. Pollard, "Monte Carlo Methods for Index Computation (mod p)," Mathematics of Computation, Vol. 32, pp. 918-924, 1978.

[Sho97] V. Shoup, "Lower bounds for Discrete Logarithms and Related Problems," Eurocrypt '97, LNCS 1233, Springer-Verlag, pp. 256-66, 1997.

[Tes98] E. Teske, "Speeding up Pollard's Rho Method for Computing Discrete Logarithms," Algorithmic Number Theory Symposium III, LNCS 1423, pp.541-554, 1998.