The Computational Square-Root Exponent Problem- Revisited *

Fangguo Zhang

School of Information Science and Technology,
Sun Yat-sen University, Guangzhou 510006, China
isszhfg@mail.sysu.edu.cn

Abstract. In this paper, we revisit the Computational Square-Root Exponent Problem (CSREP), and give a more generic condition such that CSREP is polynomial-time equivalent to the Computational Diffie-Hellman Problem (CDHP) in the group with prime order. The results obtained in this paper contain Zhang *et al.*'s results at IWCC2011. We also analyze the existence of such condition. Although primes satisfying such condition are rare (compare to all primes), it can be regarded as an evidence that CSREP may be equivalent to CDHP.

Keywords: Diffie-Hellman problem, square Diffie-Hellman problem, square-root exponent problem, equivalence, order.

1 Introduction

The discrete logarithm problem (DLP) and the Diffie-Hellman problem as cryptographic primitives play an important role in modern cryptology. For example the Diffie-Hellman key exchange [6], the ElGamal encryption [7], and the official U.S. Digital Signature Algorithm (DSA) [8], etc. Due to Pohlig and Hellman attack [17], it is restricted to groups of prime order p in this paper, where the DLP is the problem to find $x \in Z_p$ given (g, g^x), and the DHP or computational Diffie-Hellman problem(CDHP) is the problem to compute g^{ab} given (g, g^a, g^b), here $g \in G$ be a generator of group G. Maurer and Wolf [12, 14] have proved that, for every cyclic group G with prime order p, there exists polynomial time algorithm that reduces the computation of DLP in G to the computation of CDHP in G if we are able to find an elliptic curve, called *auxiliary elliptic curve*, over $\mathbb{F}_p$ with smooth order.

There are many variations of DLP and DHP, such as constrained DLP[15], P-DH problems[9], Inverse Computational Diffie-Hellman Problem (Inv-CDHP), Square Computational Diffie-Hellman Problem (Squ-CDHP), Diffie-Hellman Knowledge (DHK) problem[5], etc. For Inv-CDHP(the problem to compute $g^{a^{-1}}$ given

* This work is supported by the National Natural Science Foundation of China (No. 61070168).

(g, g^a)) and Squ-CDHP(the problem to compute g^{a^2} given (g, g^a)), due to the results of [2, 13, 19], we have the following fact: CDHP, Inv-CDHP and Squ-CDHP are polynomial time equivalent.

The computational square-root exponent problem (CSREP) is firstly proposed by Konoma *et al.* [10] in 2004, which is a problem to compute a value whose discrete logarithm is a square root of the discrete logarithm of a given value. CSREP can be regarded as a variations of CDHP. Konoma *et al.* used CSREP to analyze reduction between the discrete logarithm problem modulo a prime and the factoring problem. They also showed that CSREP is the first problem known to stay between the computational Diffie-Hellman problem and the decisional Diffie-Hellman problem with respect to the computational reduction.

However, as showed by Zhang *et al.* at [23], that under proper conditions the CSREP is polynomial-time equivalent to the Computational Diffie-Hellman Problem (CDHP). That means Konoma *et al.*'s claim is not right.

In this paper, we will give more witness that the CSREP is polynomial-time equivalent to the CDHP. The conditions showed in this paper are more generic than Zhang *et al.*'s results.

The remainder of this paper is organized as follows. In Section 2, we define certain notations and recall the computational square-root exponent problem. We discuss the P-DH function and CSREP in section 3, we provide and prove our main result in section 4 and discuss the existence of the group which satisfied the main theorem. We present an algorithm and certain example in section 5 and conclude the paper in section 6.

2 CSREP and Preliminaries

2.1 CSREP and DSREP

In [10], Konoma *et al.* defined two new problems called Computational Square-Root Exponent Problem (CSREP) and Decisional Square-Root Exponent Problem (DSREP). Konoma *et al.*'s definitions for CSREP and DSREP are over the multiplicative group modulo a prime p. We recall the definitions for CSREP and DSREP in any cyclic group with order q as follows:

Definition 1. CSREP *Let G be a cyclic group of order q and let $g \in G$ be a generator of G. Given g and g^a as input, output $g^{a^{\frac{1}{2}}}$ if a is a quadratic residue modulo q. Otherwise, output $\perp$.*

Definition 2. DSREP *Let G be a cyclic group of order q and let $g \in G$ be a generator of G. Given g, g^a and y as input, decide whether the discrete logarithm of y is a square root of the discrete logarithm of g^a. That is, output 1 if $y = g^{a^{\frac{1}{2}}}$ and 0 if $y \neq g^{a^{\frac{1}{2}}}$.*

Similar to the gap Diffie-Hellman problem, when we introduce the bilinear pairing into the group G, we can obtain the gap CSREP, i.e., in such group G, the DSREP is easy, and the CSREP is still hard.

Let G be (mutiplicative) cyclic groups of order q. Let g be a generator of G.

Definition 3. *A map $e : G \times G \to \mathbb{G}_T$ (here $\mathbb{G}_T$ is another mutiplicative cyclic group such that $|G| = |\mathbb{G}_T| = q$) is called a bilinear pairing if it satisfies the following properties:*

1. **Bilinearity:** *For all $u, v \in G$ and $a, b \in \mathbb{Z}_q$, we have $e(u^a,\ v^b) = e(u,\ v)^{ab}$.*
2. **Non-degeneracy:** *$e(g, g) \neq 1$. In other words, if g is a generator of G, then $e(g,\ g)$ generates $\mathbb{G}_T$.*
3. **Computability:** *There is an efficient algorithm to compute $e(u,\ v)$ for all $u, v \in G$.*

We say that G is a bilinear group if there exists a group $\mathbb{G}_T$, and a bilinear pairing $e : G \times G \to \mathbb{G}_T$ as above. Such groups can be found on supersingular elliptic curves or hyperelliptic curves over finite fields, and the bilinear parings can be derived from the Weil or Tate pairing.

Furthermore, Zhang *et al.* [21] designed a new signature scheme without random oracles from bilinear pairings and the CSREP. Zhang *et al.* [22] also proposed a new designated confirmer signature scheme from bilinear pairings and the hardness of the CSREP.

Due to the method of Pohlig and Hellman [17], the hardness of DLP on the group G of order q can be reduced to DLP on the group with the order of the largest prime factor of q. Therefore, for the following sections we focus on the cases where the order of group G is a prime p.

2.2 Quadratic residue modulo p and its polynomial representation

Let p be an odd prime and a be an integer relatively prime to p. We say that the integer a is a quadratic residue of p if the congruence $x^2 \equiv a \pmod{p}$ has a solution. Otherwise, we say a is a quadratic nonresidue of p.

Definition 4 (Legendre symbol). *Suppose p is an odd prime. For any integer $a > 0$, we define the Legendre symbol $\left(\frac{a}{p}\right)$ as follows:*

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \text{ is a quadratic residue modulo } p \\ -1 & \text{if } a \text{ is a quadratic nonresidue modulo } p \end{cases}$$

Theorem 1 (Euler's Criterion). *Suppose p is an odd prime. Then*

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} (mod\ p).$$

A polynomial $f(x) \in F_p[x]$ is called a polynomial representation of the square root function mod p if it satisfies the equation $\sqrt{x} \equiv f(x) \pmod{p}$ whenever x is a quadratic residue. It is well known that the simple formula $a^{\frac{p+1}{4}} \pmod{p}$ gives a square root of a when $p \equiv 3 \pmod{4}$. Let $p - 1 = 2^s n$ with n odd. Agou *et al.*[1] proved the existence of polynomial representations $f(x)$ of the square root function of degree $deg(f) \leq (p-3)/2$ and length (the number of nonzero terms) at most 2^{s-1}. Carella[4] utilized a different method to generate specific

polynomial representations of the square root function, the formulas for the cases $s = 2, 3$ and 4 are also given.

For example, when $p = 2^3 n + 1$,

$$\sqrt{x} = f(x) = 2^{-2} x^{(n+1)/2}[z^{3n}(1 - x^{2n})(1 - x^n z^{2n}) + z^n(1 - x^{2n})(1 + x^n z^{2n}) + z^{2n}(1 + x^{2n})(1 - x^n) + (1 + x^{2n})(1 + x^n)],$$

where z is a quadratic nonresidue modulo p.

2.3 Some notations

To analyze and clarify the complexity of various cryptography primitives, a useful complexity analysis is to show reductions among these primitives. Generally, to prove the equivalence of two problems, it may be easier to show the reduction relationship between them. Therefore, before describing the mathematical problems, we need the following notations from complexity theory.

- We say problem **A** is polynomial time reducible to problem **B**, denoted by $\mathbf{B} \Rightarrow \mathbf{A}$, if there exists a polynomial time algorithm $\mathcal{R}$ for solving problem **A** that makes calls to a subroutine for problem **B**. In this case, we also say the problem **B** is *harder* than the problem **A**.
- We say that **A** and **B** are polynomial time equivalent if **A** is polynomial time reducible to **B** and **B** is polynomial time reducible to **A**.

3 CSREP and P-Diffie-Hellman Function

Kiltz [9] suggested a toolbox of cryptographic functions called P-Diffie-Hellman functions. In particular, Kiltz proved that computing P-Diffie-Hellman function is computationally equivalent to computing DHP for certain class of groups. We briefly review the necessary facts about P-Diffie-Hellman function using the same notations as [9].

Let G be a finite cyclic group whose order $|G|$ is an $n-$bit integer. Let $Z_{|G|}$ denote the ring of integer residue classes modulo $|G|$. Let $\mathbb{P}_l^k$ be the family of sets of all non-linear polynomials $P(a, b)$ over $\mathbb{Z}_{|G|}$ of the form

$$P(a, b) = \sum_{i,j \in \{0 \ldots l\}} c_{ij} a^i b^j$$

with coefficients $c_{ij} \in \mathbb{Z}_{|G|}$ and $|c_{ij}|$ bounded by k.

For a cyclic finite group G, a fixed generator g of G and a polynomial $P \in \mathbb{P} = \mathbb{P}_{|G|-1}^{\lfloor |G|/2 \rfloor}$, define the P-Diffie-Hellman function, P-DH: $G \times G \to G$ as

$$P = DH(g^a, g^b) := g^{P(a,b)}.$$

Clearly computing the P-DH function cannot be harder than computing the DH function for a polynomial $P(a, b)$ due to the computation of $P(a, b)$ can

be obtained by repeated multiplication or squaring(i.e., given g^a and g^b, any monomial $g^{c_{ij}a^ib^j}$ can be computed by repeated multiplication or squaring in the exponent), therefore, CDHP $\Rightarrow$ P-DH.

Kiltz proved the following results(Theorem 3 in [9]):

For $l \in O(\sqrt{\log n})$, and for every $P, Q \in \mathbb{P}_l^{poly(n)}$ the following relation holds:

$$P - CDH \Leftrightarrow Q - CDH (\Leftrightarrow CDH).$$

Notice that CDH is in $\mathbb{P}_l^{poly(n)}$.

The square root function mod p has a polynomial representation due to Agou *et al.*'s work, therefore, CSREP can be regarded as P-Diffie-Hellman function problem. When the polynomial P of the square root function mod p is in $\mathbb{P}_l^{poly(n)}$, here $n = \log p$, then using Kiltz's result, we have CDHP $\Leftrightarrow$ CSREP in group G with order prime p.

However, we can not get CSREP $\Rightarrow$ CDHP using Kiltz's method. Because when we regard CSREP as P-Diffie-Hellman function problem, the degree of P is about $|G|$, i.e., P is not in $\mathbb{P}_l^{poly(n)}$.

4 Relation Between CSREP and CDHP

We will describe the main result of this paper in this section. When the order of the group satisfies certain conditions, the CSREP is polynomial time equivalent to the CDHP. We first define a new integer sequence, and explain the relation of this integer sequence and CSREP, then give the main theorem. We also discuss the relation of the main result of this paper and Zhang et al.'s results in [23].

4.1 A new integer sequence

For any positive integer a, we define a sequence $\{a_i\}$ as follows:

$$a_0 = a,$$
$$a_{i+1} = \begin{cases} \frac{a_i}{2} & \text{if } a_i \text{ is even} \\ \frac{a_i - 1}{2} + a & \text{if } a_i \text{ is odd} \end{cases} \quad (1)$$

If there is an integer $T > 0$ such that $a_T = a$, we call this sequence is periodic. The **period** of the sequence is the smallest such integer T.

A basic result about the period of the above sequence is: if the period of the sequence defined by a is T, then we have $a_{T-1} = 1$. This is because if $a_T = a$, then

$$a_T = a = \frac{a_{T-1}}{2} \text{ or } a_T = a = \frac{a_{T-1} - 1}{2} + a$$

So, $a_{T-1} = 2a$ (this is impossible, since $a_i < 2a$ from the construction of the sequence), or $a_{T-1} = 1$. Therefore, $a_{T-1} = 1$.

For example:

$a = 17$, the sequence defined by 17 with period 10:

$$17, 25, 29, 31, 32, 16, 8, 4, 2, 1$$

$a = 38$, the sequence defined by 38 with period 20:

$$38, 19, 47, 61, 68, 34, 17, 46, 23, 49, 62, 31, 53, 64, 32, 16, 8, 4, 2, 1$$

4.2 CSREP is equivalent to CDHP in some groups

Assume that $x^{\frac{1}{2}} \mod p \equiv \pm x^{\lambda} \mod p$. When we repeat to solve the square root for x, we have:

$$x^{\frac{1}{2}} \equiv \pm x^{\lambda}$$

$$x^{\frac{1}{4}} \equiv (x^{\frac{1}{2}})^{\frac{1}{2}} \equiv (x^{\lambda})^{\frac{1}{2}} \equiv \begin{cases} \pm x^{\frac{\lambda}{2}} & \text{if } \lambda \text{ is even} \\ \pm x^{\frac{\lambda-1}{2}+\frac{1}{2}} \equiv \pm x^{\frac{\lambda-1}{2}+\lambda} & \text{if } \lambda \text{ is odd} \end{cases} \tag{2}$$

Therefore, $x^{\frac{1}{2^i}} \equiv \pm x^{\lambda_{i-1}}$, here λ_{i-1} is the $i-1$-th item of the integer sequence defined by λ. So the above integer sequence is related to CSREP. We have the following main result:

Theorem 2. *Let G be a cyclic group of prime order p and let $g \in G$ be generator of G. When x is a quadratic residue modulo p, if $x^{\frac{1}{2}} \equiv \pm x^{\lambda} \mod p$ holds, and if the period of the sequence constructed by λ using above method is polynomial in $\log p$, then computing CSREP in G is polynomial time equivalent to computing CDHP in G.*

Proof . Clearly we have $CDHP \Rightarrow CSREP$. Now we prove that $CSREP \Rightarrow CDHP$.

Due to the work of Maurer et al.[13], Bao et al.[2] and Sadeghi et al.[19], we have CDHP $\Leftrightarrow$ Squ-CDHP. Therefore, if we can prove that $CSREP \Rightarrow$ Squ-CDHP, then we have $CSREP \Rightarrow CDHP$.

Given a CSREP-oracle $\mathcal{A}$, on input g, g^a for $a \in \mathbb{Z}_p^*$, $\mathcal{A}$ output $g^{a^{\frac{1}{2}}}$, if a is a quadratic residue modulo p. Otherwise, output $\perp$.

The following we want to compute g^{a^2} from g and g^a for $a \in \mathbb{Z}_p^*$ through calling the oracle $\mathcal{A}$.

Assume that $x^{\frac{1}{2}} \equiv \pm x^{\lambda} (\mod p)$(For example: if $p = 4k - 1$, then $\lambda = k$), the sequence defined by λ with period T is

$$\lambda_0 = \lambda, \lambda_1, \lambda_2, ..., \lambda_{T-2} = 2, \lambda_{T-1} = 1.$$

When we call the oracle $\mathcal{A}$ one time on g and g^a, we have

$$g^{a^{\frac{1}{2}}} = g^{\pm a^{\lambda}} = g^{\pm a^{\lambda_0}} = \mathcal{A}(g,\ g^a).$$

For $\pm a^{\lambda_0}$, there is only one element is quadratic residue of p when $p = 4k - 1$, we call the oracle $\mathcal{A}$ again and obtained

$$\mathcal{A}(g,\ g^{a^\lambda}) = g^{\pm a^{\lambda_1}}\ or\ \mathcal{A}(g,\ g^{-a^\lambda}) = g^{\pm a^{\lambda_1}}$$

Continue calling the oracle $\mathcal{A}$ on the last output of $\mathcal{A}$:

$$\mathcal{A}(g,\ g^{a^{\lambda_1}}) = g^{\pm a^{\lambda_2}}\ or\ \mathcal{A}(g,\ g^{-a^{\lambda_1}}) = g^{\pm a^{\lambda_2}}$$

.....

So we can compute g^{a^2} from $(g,\ g^a)$ by iteratively calling oracle $\mathcal{A}$ $T-1$ times.

Therefore, if the period T is polynomial in $\log p$, then computing CSREP in G is polynomial time equivalent to computing CDHP in G. □

Given an integer a, how to determine the period T of the sequence defined by a using the construction of Section 4.1? We find that the period is related to the order of 2 modulo $2a-1$:

Theorem 3. *The period of the sequence defined by a is $Order(2, 2a-1)$, here $Order(2, 2a-1)$ means the order of 2 modulo $2a-1$.*

Proof . If the period of the sequence defined by a is T, then we have $a_{T-1} = 1$, $a_{T-2} = 2$, ..., $a_{T-s} = 2^{s-1}$, ..., $a_{T-j} = 2^{j-1}(\mod 2a-1)$,

$$a_{T-j-1} = 2^j = 2a_{T-j} \text{ or } 2a_{T-j-1} - 2a + 1 = 2a_{T-j}(\mod 2a-1),$$

.....

$$a_0 = a = 2^{T-1} \mod 2a-1.$$

Therefore, $2^T = 2a(\mod 2a-1) = 1(\mod 2a-1)$.

If $Order(2, 2a-1) = l$, then we have

$$2^l \equiv 1(\mod 2a-1) \equiv 2a(\mod 2a-1),$$

Since $Gcd(2, 2a-1) = 1$, so, $2^{l-1} \equiv a(\mod 2a-1)$. We have

$$2^{l-1} = u(2a-1) + a = u(2a-1) + a_0$$

u and a have same parity, i.e., if a is odd (or even), then u is odd(or even). Assume that $a_0, a_1, a_2, ...$ is the sequence defined by a.

$$2^{l-2} = u_1(2a-1) + a_1$$

$u_1 = u/2$ *or* $(u-1)/2$ has same parity with a_1.

$$2^{l-i} = u_{i-1}(2a-1) + a_{i-1}$$

$u_{i-1} = u_{i-2}/2$ *or* $(u_{i-2}-1)/2$ has same parity with a_{i-1}.

There exists a $h < l$, such that $u_h = 0$.

$$2^{l-h-1} = u_h(2a-1) + a_h = a_h$$

Therefore, $a_{l-1} = 1$ and $a_l = a$. □

4.3 Relation to Zhang et al. and Roh et al.'s results

Zhang et al. analyzed the complexity of the CSREP in [23], and firstly showed that under proper conditions the CSREP is polynomial-time equivalent to the CDHP. The following theorem is the main result in [23].

Theorem 4. *[23] Let G be a cyclic group of prime order p and let $g \in G$ be generator of G. Let $p = 4k - 1$, for some $k, i, j \in \mathbb{Z}$, and i, j are polynomial in $\log p$, if one of the following conditions is satisfied:*

1). $k^i \equiv \pm 2^j \mod p - 1$
2). $k \equiv 2^j + 1 \mod p - 1$
3). $k \equiv 2^j - 2^{j-1} + 1 \mod p - 1$
4). $k \equiv 2^{2j} \pm 2^j + 1 \mod p - 1$
5). $k \equiv 2^{2j+1} \pm 2^j + 1 \mod p - 1$

then computing CSREP in G is polynomial time equivalent to computing CDHP in G.

Now we will show that the results in Theorem 2 are more generic than Zhang *et al.*'s results which include the results in Theorem 4. That is we need to show when k satisfies the conditions in Theorem 4, then the $Order(2, 2k-1)$ is also polynomial in $\log p$, here $p = 4k - 1$.

For $k^i \equiv \pm 2^j \mod p - 1$, we have $k^i \equiv \pm 2^j \mod 4k - 2$, so, $k^i \equiv \pm 2^j \mod 2k-1$. Notice that $k \equiv 2^{-1} \mod 2k-1$. Therefore, $2^{-i} \equiv \pm 2^j \mod 2k-1$, this is equivalent to $2^{i+j} \equiv \pm 1 \mod 2k - 1$, i.e., $Order(2, 2k-1) = i + j$ or $2(i + j)$ are also polynomial in $\log p$.

For $k \equiv 2^j + 1 \mod p-1$, we have $k \equiv 2^j + 1 \mod 2k-1$. So, $2^{-1} \equiv 2^j + 1 \mod 2k - 1$, this is equivalent to $2^{j+1} + 2 \equiv 1 \mod 2k - 1$, then $2^{j+1} \equiv -1 \mod 2k-1$. Therefore, $2^{2j+2} \equiv 1 \mod 2k-1$, i.e., $Order(2, 2k-1) = 2j + 2$ is also polynomial in $\log p$.

For $k \equiv 2^j - 2^{j-1} + 1 \mod p-1$, we have $k \equiv 2^j - 2^{j-1} + 1 \mod 2k-1$. So, $2^{-1} \equiv 2^j - 2^{j-1} + 1 \mod 2k-1$, $2^{j+1} + 2 = 2(2^j + 1) \equiv 2^j + 1 \mod 2k-1$, then $2^j + 1 \equiv 1 \mod 2k-1$(this is impossible due to $2k-1$ is odd) or $2^j + 1 \equiv 0 \mod 2k - 1$. Therefore, $2^{2j} \equiv 1 \mod 2k - 1$, i.e., $Order(2, 2k-1) = 2j$ is also polynomial in $\log p$.

For $k \equiv 2^{2j} \pm 2^j + 1 \mod p-1$, we have $k \equiv 2^{2j} \pm 2^j + 1 \mod 2k-1$. So, $2^{-1} \equiv 2^{2j} \pm 2^j + 1 \mod 2k-1$, $2^{2j+1} \pm 2^{j+1} + 2 \equiv 1 \mod 2k-1$, then $2^{2j+1} \pm 2^{j+1} + 1 \equiv 0 \mod 2k-1$, $2^{2j+1} + 1 \equiv \pm 2^{j+1} \mod 2k-1$, and $(2^{2j+1}+1)^2 \equiv (\pm 2^{j+1})^2 \equiv 2^{2j+2} \mod 2k - 1$. i.e., $2^{2j+2} + 1 \equiv 0 \mod 2k - 1$. Therefore, $2^{8j+4} \equiv 1 \mod 2k - 1$, i.e., $Order(2, 2k-1) = 8j + 4$ is also polynomial in $\log p$.

For $k \equiv 2^{2j+1} \pm 2^j + 1 \mod p-1$, we have $k \equiv 2^{2j+1} \pm 2^j + 1 \mod 2k-1$. So, $2^{-1} \equiv 2^{2j+1} \pm 2^j + 1$, $2^{2j+2} \pm 2^{j+1} + 2 \equiv 1 \mod 2k-1$, then $2^{2j+2} \pm 2^{j+1} + 1 \equiv 0 \mod 2k-1$. Therefore $(2^{j+1} - 1)(2^{2j+2} + 2^{j+1} + 1) \equiv 2^{3j+3} - 1 \equiv 0 \mod 2k-1$, i.e., $2^{3j+3} \equiv 1 \mod 2k - 1$, or, $(2^{j+1} + 1)(2^{2j+2} - 2^{j+1} + 1) \equiv 2^{3j+3} + 1 \equiv 0 \mod 2k - 1$, i.e., $2^{6j+6} \equiv 1 \mod 2k - 1$. This means $Order(2, 2k-1) = 3j + 3$ or $6j + 6$ are also polynomial in $\log p$.

We point out an error in Zhang et al.'s result. Notice that, in the original Theorem 2 of [23], there is a "$\pm$" following "$\equiv$" at each case. Zhang et al. only gave the proof for all "+" cases, and for the "−" cases, Zhang et al. said: "For the case of $k^i \equiv -2^j$, $k \equiv -(2^j+1)$, $k \equiv -(2^j - 2^{j-1}+1)$, $k \equiv -(2^{2j} \pm 2^j + 1)$ and $k \equiv -(2^{2j+1} \pm 2^j + 1) \pmod{p-1}$, using above method, we can get $g^{a^{-2}}$ by iteratively calling oracle A_3, so we can get $g^{a^{-1}}$ by iteratively calling oracle A_3 one more time, this is the Inv-CDHP." However, we find that when we consider the case of "$-$", beside case 1, the claim of other cases are not right, i.e., we can not get $g^{a^{-2}}$ by iteratively calling oracle A_3 polynomial times in $\log p$. For example, when $k \equiv -(2^j+1) \mod p-1$, when we call oracle A_3(same as $\mathcal{A}$ in this paper) one time, we obtained g^{-2^j-1}. We call oracle A_3 again on g^{-2^j-1}, we have $g^{-2^{j-1}-(-2^j-1)}$. So when iteratively calling oracle A_3 j times on g^a, we have

$$g^{a^{2^j-2^{j-1}+\ldots-2+1}} \text{ when } j \text{ is even}$$

or

$$g^{a^{-(2^j-2^{j-1}+\ldots-2+1)}} \text{ when } j \text{ is odd}$$

In general, we can not get $g^{a^{-2}}$ from above two cases through polynomial of j times calling to oracle A_3. For example,

$$p = 26592048236481710707811460913 1 = 4\times 66480120591204276769528652283 - 1$$

here $k = 66480120591204276769528652283$, $k \equiv -2^{112} - 1 \mod p-1$, i.e., $j = 112$. However, $Order(2, 2k-1) = 10703610197842957295104500$ is very large compare to $j = 112$. For other cases of "$-$", they can be verified that the $Order(2, 2k-1)$ are usually very large.

Very recently, Roh et al. also analyzed the complexity of the CSREP(they called SRDHP) in [18], and proposed the following theorem:

Theorem 5. *[18] Let G be a cyclic group of prime order p and let $g \in G$ be generator of G. Let $p-1 = 2^s t$. If 2^s is of order $(logp)^{O(1)}$, then SDHP and SRDHP are polynomial time equivalent.*

The results proposed in this paper mainly focuses on the case of $p \equiv 3 \mod 4$ or $p \equiv 5 \mod 5$(We need that the polynomial of the quadratic residue function modulo p is a monomial). In such case, $s = 1$ or $s = 2$, i.e., for the case of $p \equiv 3 \mod 4$ or $p \equiv 5 \mod 5$, 2^s is of order $(logp)^{O(1)}$. So, Roh et al.'s result are more generic than the results proposed in this paper. However, the method used in this paper seams simple.

5 How to find such groups?

From the Theorem 2 and Theorem 3, for a group with prime order p, when the polynomial of the quadratic residue function modulo p is a monomial, i.e., $x^{\frac{1}{2}} \equiv \pm x^{\lambda} \mod p$, and the $Order(2, 2\lambda - 1)$ is polynomial in $\log p$, then the

CSREP is polynomial-time equivalent to the CDHP. How many primes are there satisfying Theorem 2? By Li and Pomerance [11, 16], assuming GRH, most n coprime to b have $\lambda(n)/l_b(n)$ small, where $l_b(n)$ denotes the mutliplicative order of b in $\mathbb{Z}/n\mathbb{Z}^*$ and $\lambda(n)$ denotes the order of the largest cyclic subgroup in $\mathbb{Z}/n\mathbb{Z}^*$. This means that for any random integer a, to make $Order(2, 2a-1) = l$ small is very infrequent. So, for some cryptographic group used in practical applications(for example, NIST ECC, IEEE P1363, SECG, ect.), it can not prove that CSREP=CDHP using the proposed method.

However, when we consider the pairing based cryptosystem, there are some papers [3][20] to suggest using Mersenne prime number(i.e., $p = 2^n - 1$) or generalized Mersenne prime number($p = 2^n \pm 2^m \pm 1$) for the order p of the group G. This is because the computation of $f_{p,P}(Q)$ for the Tate pairing or Weil pairing is very efficient in this case. For example, when $p = 2^{190} + 2^{95} - 1$, since $pP = (2^{95}(2^{190-95}+1)-1)P$ involves only one addition and one subtraction plus 190 doublings. For this example, $Order(2, 2k-1) = 17860$.

Although primes satisfying such condition of Theorem 2 are rare (compare to all primes), it can be regarded as an evidence that CSREP may be equivalent to CDHP. Furthermore, we can confirm that there does exist certain primes that satisfy the conditions of Theorem 2. Now we describe an algorithm to find such prime p with $p \equiv 3 \mod 4$. Notice that $p = 4k - 1$ and $2^T \equiv 1 \mod 2k - 1$, here T is polynomial in $\log p$. We assume that $T < \log^2 p$.

The equation $2^T \equiv 1 \mod 2k - 1$ is equivalent to $2^T = n(2k-1) + 1$ for some integer n. Therefore, we have

$$k = \frac{2^T - 1 + n}{2n} \quad \text{for certain integers } k, T, n.$$

Then the question of finding primes p that satisfy the conditions of Theorem 2, is equivalent to find the above integers k, n and T such that $p = 4k - 1$ is a prime. More precisely, we have the following Algorithm 1. Through setting λ and c in Algorithm 1, we can get a prime with any bits satisfies Theorem 2.

Algorithm 1 Finding primes for Theorem 2 with λ bits

Require: $T1 = \lambda - 1, T2 = \lambda - 1 + c$: the range of T.
Ensure: The prime p.

```
for T = T1 to T2 do
  for n = 2^{T-T1} to 2^{T+1-T1} do
    a ← 2^T − 1 + n,  b ← 2n
    if b divides a then
      k ← a/b, p ← 4k − 1
      if (p is prime) then
        return p
      end if
    end if
  end for
end for
```

Using Algorithm 1, we try to find a prime of 160 bits satisfies Theorem 2. Thus, we have the following example.

We set $\lambda = 160$, $c = 8$, then we get the prime as

$$p = 1120192871726680081018393165195713931233289613779$$

then

$$k = 280048217931670020254598291298928482808322403445.$$

Moreover, we have $Order(2, 2k - 1) = 166$.

When we set $\lambda = 192$, $c = 20$, then we get the prime as

$$p = 382476679521505924747246299320550947352419350787109459001 9$$

then

$$k = 956191698803764811868115748301377368381048376967773647505.$$

Moreover, we have $Order(2, 2k - 1) = 210$.

It is not hard to find the *auxiliary elliptic curve* for the groups with orders for these primes, that means the DLP, CDHP and CSREP are polynomial time equivalent with respect to the computational reduction in these groups.

6 Conclusion

Konoma *et al.* proposed a new variant of computational Diffie-Hellman problem: CSREP, and used CSREP to analyze reduction between the discrete logarithm problem modulo a prime and the factoring problem. They also showed that CSREP is the first problem known to stay between the computational Diffie-Hellman problem and the decisional Diffie-Hellman problem with respect to the computational reduction. In this paper, we studied CSREP, and give a more generic condition such that CSREP is polynomial-time equivalent to the Computational Diffie-Hellman Problem (CDHP) in the group with prime order. The results obtained in this paper contain Zhang *et al.*'s results at IWCC2011. We also analyze the existence of such condition. Although there are too few(compare with all primes) such primes satisfying the condition, this case can be regarded as an evidence that CSREP may be equivalent to CDHP. That means that CSREP maybe not the problem known to stay between the CDHP and the DDHP with respect to the computational reduction.

Acknowledgements

I thank Prof. Steven Galbraith for bringing the paper of Roh et al to my attention.

References

1. S.J. Agou, M Deleglise, J.L. Nicolas, Short Polynomial Representations for Square Roots Modulo p, Designs, Codes, Cryptography, 28, 33-44, 2003.
2. F. Bao, R. Deng and H. Zhu, Variations of Diffie-Hellman problem, In Proceedings of ICICS 2003, LNCS 2836, pp. 301-312, Springer-Verlag, 2003.
3. P.S.L.M. Barreto, H.Y. Kim, B. Lynn, M. Scott. Efficient Algorithms for Pairing-Based Cryptosystems. Crypto 2002, LNCS 2442, Springer-Verlag (2002), pp. 354-368.
4. N. A. Carella, Formulas for the Square Roots Mod p, http://arxiv.org/abs/1101.4605.
5. A. W. Dent, The Hardness of the DHK Problem in the Generic Group Model, Cryptology ePrint Archive, Report 2006/156, 2006. http://eprint.iacr.org/.
6. W. Diffie and M. Hellman, New Directions in cryptography, IEEE Transactions on Information Theory, volume 22, pp. 644-654, 1976.
7. T. ElGamal, A public-key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory, volume 31, pp. 469-472, 1985.
8. FIPS 186-2, Digital signature standard, Federal Information Processing Standards Publication 186-2, February 2000.
9. E. Kiltz, A tool box of cryptographic functions related to the Diffie-Hellman function, Indocrypt'01, LNCS 2247, pp. 339C349. Springer-Verlag, 2001.
10. C. Konoma, M. Mambo, and H. Shizuya, Complexity analysis of the cryptographic primitive problems through square-root exponent, IEICE Trans. Fundamentals, vol.E87-A, no.5, pp. 1083-1091, May 2004.
11. S. Li and C. Pomerance, On the Artin-Carmichael primitive root problem on average, Mathematika 55 (2009), 167-176.
12. U. M. Maurer, Towards the equivalence of breaking hte Diffie-Hellman protocol and computing discrete logatithms, CRYPTO94, LNCS 839, pp. 271-281, Springer-Verlag, 1994.
13. U. M. Maurer and S. Wolf, Diffie-Hellman oracles, CRYPTO96, LNCS 1109, pp. 268-282. Springer Verlag, Berlin Germany, 1996.
14. U. M. Maurer and S. Wolf, The Relationship Between Breaking the Diffie-Hellman Protocol and Computing Discrete Logarithms, SIAM J. Comput. 28(5): 1689-1721, 1999.
15. I. Mironov, A. Mityagin and K. Nissim, Hard instances of the constrained discrete logarithm problem, 7th Algorithmic Number Theory Symposium (ANTS VII), LNCS 4076, pp. 582-598, 2006.
16. C. Pomerance, Order and chaos, 2011, http://www.math.dartmouth.edu/∼carlp/ordertalkmsri.pdf.
17. S. C. Pohlig and M. E. Hellman, An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, IEEE-Transactions on Information Theory 24, pp. 106-110, 1978.
18. D. Roh and S. G. Hahn, The square root DiffieCHellman problem, to appear at Designs, Codes and Cryptography, Online 11 April, 2011 http://www.springerlink.com/content/Author=Sang+Geun+Hahn
19. A. R. Sadeghi and M. Steiner, Assumptions related to discrete logarithms: Why subtleties make a real difference, EUROCRYPT 2001, LNCS 2045, pp. 244-261, Springer-Verlag, 2001.

20. Michael Scott: Faster Pairings Using an Elliptic Curve with an Efficient Endomorphism. INDOCRYPT 2005, LNCS 3797,pp. 258-269, 2005.
21. F. Zhang, X. Chen, W. Susilo, and Y. Mu, A new signature scheme without random oracles from bilinear pairings. VietCrypt 2006, LNCS 4341, pp. 67-80. Springer-Verlag, 2006.
22. F. Zhang, X. Chen, B. Wei, Efficient designated confirmer signature from bilinear pairings, ASIACCS 2008, pp. 363-368, 2008.
23. F. Zhang and P. Wang, On Relationship of Computational Diffie-Hellman Problem and Computational Square-Root Exponent Problem, IWCC 2011, LNCS 6639, pp. 283–293. Springer, Heidelberg, 2011.