数理解析研究所講究録 1325 巻 2003 年 134-139

Short Exponent DDH

小柴健史 (Takeshi Koshiba)
富士通研究所セキュアコンピューティング研究部
Secure Computing Lab., Fujitsu Laboratories Ltd.

黒澤 馨 (Kaoru Kurosawa)
茨城大学工学部情報工学科
Dept. Computer and Information Science, Ibaraki Univ.

1 Introduction

The discrete log (DL) assumption and the decision Diffie-Hellman (DDH) assumption are basis of many applications in modern cryptography.

Blum and Micali [1] presented the first cryptographically secure pseudo-random bit generators (PRBG) under the DL assumption over Z_p^*, where p is a prime. Long and Wigderson [5], and Peralta [8] showed that up to $O(\log\log p)$ pseudo-bits can be extracted by a single modular exponentiation of the Blum-Micali generator.

Recently, a stronger (but reasonable) variant of the DL assumption was used to improve the efficiency of the PRBG. It claims that the DL problem is still hard even if the exponent is small. Under this assumption, Patel and Sundaram [7] showed that it is possible to extract up to $n - \omega(\log n)$ bits from one iteration of the Blum-Micali generator, where n is the bit length of p. Gennaro [4] further improved this result in such a way that each full modular exponentiation can be replaced with a short modular exponentiation. This variant of the DL assumption is called the discrete log with short exponent (DLSE) assumption.

On the other hand, the DDH assumption is described as follows. Let G be a finite Abelian group of order q, where q is a prime. Let g be a generator, that is, $G = \langle g \rangle$. Then (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^c) are indistinguishable, where a, b and c are uniformly and randomly chosen from Z_q.

Under this assumption, ElGamal encryption scheme [3] is secure in the sense of indistinguishability against chosen plaintext attack (IND-CPA) and Cramer-Shoup scheme [2] is secure in the sense of indistinguishability against chosen ciphertext attack (IND-CCA).

In this paper, we investigate a variant of the DDH assumption which claims that (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^c) are still indistinguishable even if a is small. We call it the DDH with short exponent assumption (DDH-SE assumption). If the DDH-SE assumption is true, then we can improve the efficiency of ElGamal scheme and Cramer-Shoup scheme directly. However, this assumption has not been studied so far.

For this problem, we show that the DDH-SE assumption is equivalent to two *widely accepted* assumptions, the DDH assumption and the DLSE assumption. To prove this, we first show that short exponents $\{g^s \mid s \text{ is small}\}$ and full exponents $\{g^x \mid x \in Z_q\}$ are indistinguishable under the DLSE assumption. We then show that the DDH assumption and the DLSE assumption implies the DDH-SE assumption. We further show that the DDH-SE assumption implies the DDH assumption and the DLSE assumption.

	Original assumption only	Original assumption + DLSE
DL over Z_p^*	PRBG of Blum-Micali [1]	Improved PRBG [7, 4]
DDH over G	ElGamal [3], Cramer-Shoup [2] and etc.	DDH-SE (This paper)

Table 1. Comparison with the previous works

We finally present variants of ElGamal encryption scheme and Cramer-Soup scheme. Each encryption algorithm uses a short random exponent r. Hence they are much faster than the original encryption algorithms.

It is clear that these variants are provably secure under the DDH-SE assumption. However, more than that, we can immediately see that they are provably secure under the two widely accepted assumptions, the DDH assumption and the DLSE assumption, from our result. That is, our variant of ElGamal scheme is IND-CPA under the DDH assumption and the DLSE assumption, and our variant of Cramer-Shoup scheme is IND-CCA under the same assumptions.

We believe that there will be many other applications of our result.

2 Preliminaries

$|x|$ denotes the bit length of x. $x \in_R X$ means that x is randomly chosen from a set X. For a set X, we sometimes mean the uniform distribution over X.

Let $lsb_k(z)$ be the function that returns the least significant k bits of z and $msb_k(z)$ the function that returns the most significant k bits of z. If we write

$b = msb_k(z)$, we sometimes mean that the binary representation of b is $msb_k(z)$.

Let G be a finite Abelian group of order q, where q is a prime. G can be constructed as a subgroup of Z_p^*, where p is a large prime. It can also be obtained from elliptic curves. Let g be a generator of G, that is, $G = \langle g \rangle$. Note that any $g \neq 1$ is a generator of G because the order q is a prime.

- Let n denote the bit length of q and
- let $c = \omega(\log n)$. It means that c grows faster than $a \log n$ for any constant a. That is, 2^c grows faster than any polynomial in n.

2.1 Discrete Log with Short Exponent (DLSE) Assumption

Let

$$f(z, g) = (g, g^z).$$

The discrete log (DL) problem is to compute the inverse of f. The DL assumption says that the DL problem is hard.

Assumption 2. 1 *(DL assumption)* There exists no probabilistic polynomial time Turing machine which solves the DL problem with non-negligible probability.

Next let

$$f^{se}(u||v, g) = (g, g^{u||0^{n-c}}),$$

where $|u| = c$ and $||$ denotes concatenation. Then the DLSE problem is to compute the inverse of f^{se}. The DLSE assumption says that the DLSE problem is hard.

Assumption 2. 2 *(DLSE assumption)* There exists no probabilistic polynomial time Turing machine which solves the DLSE problem with non-negligible probability.

The running time of the index-calculus method depends only on n, the size of G. The baby-step giant-step algorithm by Shanks or the *rho* and *lambda* algorithms by Pollard can solve the DLSE problem in $O(2^{c/2})$. Schnorr proved that this is the best that can be done for generic algorithms [9].

The DLSE assumption was first introduced to a key agreement protocol in order to decrease the computational cost of exponentiation [6]. Patel and Sundaram [7] then proposed a cryptographically secure PRBG under the DLSE assumption over Z_p^*, where p is a prime. (Note that the order of Z_p^* is not a prime.) Gennaro improved the efficiency [4].

2.2 Security of Public Key Cryptosystem

A public key cryptosystem consists of three probabilistic polynomial time algorithms, a key generation algorithm, an encryption algorithm and a decryption algorithm. The key generation algorithm outputs a public key pk and a secret key sk.

Consider the following model of adversaries. In the find stage, the adversary chooses two messages m_0, m_1 on input pk. She then sends these to an encryption oracle. The encryption oracle chooses a random bit b, and encrypts m_b. In the guess stage, the ciphertext C_b is given to the adversary. The adversary outputs a bit b'.

We say that the public key cryptosystem is secure in the sense of indistinguishability against chosen plaintext attack (IND-CPA) if $|\Pr(b' = b) - 1/2|$ is negligibly small (as a function of the security parameter).

The security against chosen-ciphertext attack (IND-CCA) is defined similarly except for that the adversary gets the decryption oracle and is allowed to query any ciphertext C, where it must be $C \neq C_b$ in the guess stage.

3 Short EXP $\approx$ Full EXP

For $0 \le i \le n - c$, let

$$R_i = \{2^i u \mid 0 \le 2^i u < q\}.$$

Define

$$\begin{aligned} A_0 &= \{(g, g^x) \mid x \in R_0\} \quad (= \{(g, g^x) \mid x \in Z_q\}), \\ A_{n-c} &= \{(g, g^x) \mid x \in R_{n-c}\}. \end{aligned}$$

Note that the exponent of g^x in A_0 is n bits long. On the other hand, the exponent of g^x in A_{n-c} is essentially only c bits long.

In this section, we prove that A_0 and A_{n-c} are indistinguishable under the DLSE assumption.

3.1 Overview

For $1 \le i \le n - c$, let

$$A_i = \{(g, g^x) \mid x \in R_i\}.$$

Suppose that there exists a distinguisher D which can distinguish A_{n-c} from A_0. Then by using a hybrid argument, there exists j such that A_j and A_{j+1} are distinguishable.

We will show that the DLSE problem can be solved by using the (D, j). Note that the DLSE problem is to find x of g^x in A_{n-c}.

3.2 Indistinguishability

Theorem 3. 1 *A_0 and A_{n-c} are indistinguishable under the DLSE assumption.*

Before showing Theorem 3. 1 , we show some technical lemmas. Let $c = \omega(\log n)$.

Lemma 3. 1 Suppose that there exits a probabilistic polynomial-time Turing machine D that on input $(g, g^z) \in_R A_{n-c}$, outputs the i-th bit of z with probability $1/2 + \epsilon$, where $0 < \epsilon \le 1/2$.

Then there exists a probabilistic polynomial-time Turing machine D_i that on input $g^z \in_R \{g^x \mid x \in R_{n-c}\}$, outputs the i-th bit of z with probability $1/2+\epsilon$ for any fixed $g \in G$.

It is clear that computing z from (g, g^z) is the same as computing z from (g^r, g^{rz}). Lemma 3. 1 is easily obtained from this random self reducible property.

Next let g be a generator of G.

Lemma 3. 2 For $i \ge c$, suppose that there exists a probabilistic polynomial-time Turing machine D that on input $g^{u||0^{n-i}} \in_R \{g^x \mid x \in R_{n-i}\}$, outputs the least significant bit of u with probability $1/2 + \epsilon$.

Then there exists a probabilistic polynomial-time Turing machine D' that on input $g^{v||0^{n-c}} \in_R \{g^x \mid x \in R_{n-c}\}$ and $msb_{\log t}(v)$, outputs the least significant bit of v with probability at least $1/2 + \epsilon - (4/t)$.

(Proof) Let D be a probabilistic polynomial-time machine as stated above. We construct a probabilistic polynomial-time machine D' that, given $g^{v||0^{n-c}}$ and $msb_{\log t}(v)$, outputs the least significant bit of v with probability at least $1/2 + \epsilon - (4/t)$.

Let

$$a = g^{v||0^{n-c}} \quad \text{and} \quad b = msb_{\log t}(v)$$

be the inputs to D'. That is, $v = b||v'$ for some v'. We will find the lsb of v' by using D because $lsb_1(v) = lsb_1(v')$.

(1) First, D' zeros the most significant $\log t$ bits of v by computing

$$a_1 = (a \cdot g^{-b \cdot 2^{n-\log t}}) \quad (= g^{0^{\log t}||v'||0^{n-c}}).$$

(2) Next D' computes

$$a_2 = (a_1)^{e^{i-c}},$$

where $e = 1/2 \pmod q$. (The exponent of a_1 is shifted to the right $i - c$ bits.) Let s be the integer such that $a_2 = g^s$. It is clear that

$$s = 0^{i-c+\log t}||v'||0^{n-i}.$$

(3) D' chooses $r \in R_{n-i}$ randomly and computes

$$a' = a_2 \cdot g^r \quad (= g^{s+r}).$$

Note that $r = 2^{n-i}r'$ for some r' since $r \in R_{n-i}$.
(4) D' invokes D with input (g, a').
(5) Suppose that D outputs a bit α. Then D' outputs $\beta = \alpha \oplus lsb_1(r')$.

Now suppose that $s + r < q$. Then $s + r = u||0^{n-i}$, where

$$u = (0^{i-c+\log t}||v') + r'.$$

Therefore D outputs $\alpha = lsb_1(u)$ with probability $1/2 + \epsilon$. On the other hand,

$$\alpha = lsb_1(u) = lsb_1(v') \oplus lsb_1(r') = lsb_1(v) \oplus lsb_1(r').$$

Hence D' outputs $\beta = \alpha \oplus lsb_1(r') = lsb_1(v)$. Finally we see that

$$\Pr(s + r < q) \ge 1 - 4/t. \tag{1}$$

(The proof is given in Appendix.) Consequently the success probability of D' is at least

$$(1-4/t)(1/2+\epsilon) = 1/2+\epsilon-(4/t)(1/2+\epsilon) > 1/2+\epsilon-4/t.$$

Q.E.D.

Now, we are ready to prove Theorem 3. 1 .

(Proof of Theorem 3. 1) We assume that there exists a distinguisher D between A_0 and A_{n-c}, namely,

$$|\Pr[D(A_0) = 1] - \Pr[D(A_{n-c}) = 1]| > \frac{1}{p(n)}$$

for some polynomial $p(\cdot)$. (A_0 and A_{n-c} in the above equation denote the uniform distribution over the set A_0 and A_{n-c}, respectively.) Then, for some j,

$$|\Pr[D(A_j) = 1] - \Pr[D(A_{j+1}) = 1]| > \frac{1}{np(n)}.$$

Let $p_i = \Pr[D(A_i) = 1]$. We estimate each p_i by the sampling method and denote by $\hat{p}_i$ the corresponding estimated value based on m experiments. Using the Chernoff bounds, we have

$$\Pr[\hat{p}_i > p_i + 1/8np(n)] \le e^{-2m/64(np(n))^2} \quad \text{and}$$
$$\Pr[\hat{p}_i < p_i - 1/8np(n)] \le e^{-2m/64(np(n))^2}.$$

Similarly, we can estimate all p_i with accuracy $\pm 1/8np(n)$ with high probability. Thus, we have

$$|\hat{p}_{j+1} - \hat{p}_j| > 1/np(n) - 2/8np(n) = 3/4np(n)$$

for some j. Using the estimated values, it is not hard to find i such that $|p_{i+1} - p_i| > 1/2np(n)$ with high probability.

By simple transformation, we can say that there is a probabilistic polynomial-time machine F, which can invoke D, to find an index i such that

$$\Pr[D(g, g^{u||b||0^{n-i-1}}) = b;$$
$$u||b||0^{n-i-1} \in_R R_{n-i-1}, g \in_R G] > \frac{1}{2} + \frac{1}{2np(n)}$$

with high probability. For simplicity, let us fix i satisfying the above. It is not hard to see that the distinguisher D can work as a prediction algorithm for the

$(n-i)$-th bit of discrete exponent. Thus, we rename the distinguisher (or, the prediction algorithm) P_{n-i} in order to avoid confusion. Let (g, g^z) be an input to P_{n-i}. We note that P_{n-i} works only when the least significant $n-i-1$ bits of z are all 0. By Lemma 3. 1 , we can construct the $(n-i)$-th bit predictor P'_{n-i} that works for fixed g with the same probability.

Using P'_{n-i}, we construct a probabilistic polynomial-time machine A, given $(g, g^{u||0^{n-c}})$, computes u with high probability. To this end, since $i < n-c$ from the assumption, we have only to construct the least significant bit predictor B by using Lemma 3. 2 . (Note that "the least significant bit" means not the least significant bit of $u||0^{n-c}$ but the least significant bit of u.) Then, B has to work when inputs to B are of the form

$$(g, g^{0^{i-c+1}||u||0^{n-i-1}}).$$

Let w be the exponent such that

$$(g, g^w) = (g, g^{0^{i-c+1}||u||0^{n-i-1}}).$$

Set $t = 4n^5(p(n))^2$ and guess the most significant $\log t$ bits of w. For each guessing value a of $\log t$ bits, let

$$g^{w'} = g^w \cdot g^{-a2^{n-\log t}}.$$

For $(g, g^{w'})$, we can use Lemma 3. 2 , because the most significant $\log t$ bits of w' are all 0 and P'_{n-i} is the $(n-i)$-th bit predictor that satisfies the condition of Lemma 3. 2 . We construct an algorithm E, given $(g, g^{w'})$, outputs the $(n-i)$-th bit of w' with high probability. In randomization process, the error (i.e., carrying-over by addition) occurs with probability at most $4/t$. If we take m' sampling points randomly then the probability that every point does not meet carrying-over is at least $(1-4/t)^{m'}$. Using these m' sampling points, we perform the majority voting (after neutralizing the effect of randomization). By setting $m' = n^3(p(n))^2$, we can neglect the failure probability of the majority voting. (Recall that the bias is $1/2np(n)$ and use the Chernoff bounds.) It is not hard to see that the success probability of E is at least $(1-4/t)^{m'} \leq 1 - 1/n^2$.

Let us complete the construction of B. We are still assuming that the guessing value a is correct. After zeroing the $(n-i)$-th bit of w', invoke E with $(g, (g^{w''})^e)$, where w'' is the resulting string after zeroing the $(n-i)$-th bit of w' and $e = 1/2 \pmod q$. Repeating this process c times, we can recover all bits of w' with probability at least $1 - 1/n$. Let $\hat{w}$ be the resulting candidate for w'. Check whether $g^{\hat{w}+a2^{n-\log t}} = g^w$ or not and output w if the equation holds. Since one of the guessing values is correct and the number of the guessing values is bounded by $t = 4n^5(p(n))^2$, B can compute w with probability at least $1 - 1/n$.

Thus, we can construct a probabilistic polynomial-time machine A, given $(g, g^{u||0^{n-c}})$, computes u with probability at least $1 - 1/n$. Considering that we need to find the distinguishing index i by using F before invoking A, we can say that the discrete logarithm problem with short exponents can be solved with probability at least $1 - 2/n$.

Q.E.D.

4 Equivalence: DDH + DLSE = DDH-SE

Let

$$\begin{aligned} B_0 &= \{(g, g^x, g^y, g^{xy}) \mid x \in Z_q, y \in Z_q\}, \\ C_0 &= \{(g, g^x, g^y, g^z) \mid x \in Z_q, y \in Z_q, z \in Z_q\}. \end{aligned}$$

The DDH assumption says that B_0 and C_0 are indistinguishable.

Assumption 4. 1 *(DDH assumption)* There exists no polynomial time distinguisher which can distinguish B_0 and C_0 with non-negligible probability.

Next define

$$\begin{aligned} B_{n-c} &= \{(g, g^x, g^y, g^{xy}) \mid x \in R_{n-c}, y \in Z_q\}, \\ C_{n-c} &= \{(g, g^x, g^y, g^z) \mid x \in R_{n-c}, y \in Z_q, z \in Z_q\}. \end{aligned}$$

Note that x is essentially only c bits long. Then the DDH with short exponent assumption (DDH-SE assumption) claims that B_{n-c} and C_{n-c} are still indistinguishable. ($c = \omega(\log n)$, where $n = |q|$.)

Assumption 4. 2 *(DDH-SE assumption)* There exists no polynomial time distinguisher which can distinguish B_{n-c} and C_{n-c} with non-negligible probability.

In this section, we prove that the DDH-SE assumption is equivalent to the DDH assumption and the DLSE assumption. We first show that the DDH assumption and the DLSE assumption implies the DDH-SE assumption.

Theorem 4. 1 *Suppose that the DDH assumption and the DLSE assumption are true. Then the DDH-SE assumption is true.*

(Proof) From Theorem 3. 1 , A_0 and A_{n-c} are indistinguishable under the DLSE assumption, where

$$\begin{aligned} A_0 &= \{(g, g^x) \mid x \in R_0\}, \\ A_{n-c} &= \{(g, g^x) \mid x \in R_{n-c}\}. \end{aligned}$$

First it is clear that C_0 and C_{n-c} are indistinguishable because y and z are random independently of x.

Next we prove that B_0 and B_{n-c} are indistinguishable. Suppose that there exists a distinguisher D which distinguishes B_{n-c} from B_0. Then we show that there exists a distinguisher D' which distinguishes A_{n-c} from A_0.

On input (g, g^x), D' chooses $y \in Z_q$ at random and computes g^y and $(g^x)^y$. D' then gives $(g, g^x, g^y, (g^x)^y)$ to D. Note that

$$(g, g^x, g^y, (g^x)^y) \in_R \begin{cases} B_0 & \textit{if} \quad (g, g^x) \in_R A_0, \\ B_{n-c} & \textit{if} \quad (g, g^x) \in_R A_{n-c}. \end{cases}$$

D' finally outputs the output bit of D. Then it is clear that D' can distinguish A_{n-c} from A_0.

However, this is against Theorem 3. 1 . Hence B_0 and B_{n-c} are indistinguishable. Consequently we obtain that

$$B_{n-c} \approx B_0 \approx C_0 \approx C_{n-c},$$

where $\approx$ means indistinguishable. ($B_0 \approx C_0$ comes from the DDH assumption.) Therefore, B_{n-c} and C_{n-c} are indistinguishable.

Q.E.D.

We next show that the DDH-SE assumption implies the DDH assumption and the DLSE assumption.

Theorem 4. 2 *Suppose that the DDH-SE assumption is true. Then the DDH assumption and the DLSE assumption are true.*

(Proof) First suppose that there exists a probabilistic polynomial time Turing machine M which can solve the DLSE problem with some non-negligible probability ϵ. Then we show that there exists a distinguisher D for B_{n-c} and C_{n-c}.

On input (g, g^x, g^y, α), D gives g^x to M. If M does not output x correctly, then D outputs a random bit b. Suppose that M outputs x correctly. Then D outputs b such that

$$b = \begin{cases} 1 & \textit{if} \quad \alpha = (g^y)^x \\ 0 & \textit{if} \quad \alpha \neq (g^y)^x. \end{cases}$$

Then it is easy to see that D distinguishes between B_{n-c} and C_{n-c}.

Next suppose that there exists a distinguisher D_0 which breaks the DDH assumption. Then we show that there exists a distinguisher D_1 which breaks the DDH-SE assumption.

Let (g, g^x, g^y, g^a) be the input to D_1, where $a = xy \bmod q$ or random. D_1 chooses $r \neq 0$ at random and gives $(g, (g^x)^r, g^y, (g^a)^r)$ to D_0. It is easy to see that

$$(g, (g^x)^r, g^y, (g^a)^r) \in_R \begin{cases} B_0 & \textit{if} \ (g, g^x, g^y, g^a) \in_R B_{n-c}, \\ C_0 & \textit{if} \ (g, g^x, g^y, g^a) \in_R C_{n-c}. \end{cases}$$

Finally D_1 outputs the output bit of D_0. Then it is clear that D_1 distinguishes between B_{n-c} and C_{n-c}.

Q.E.D.

From Theorem 4. 1 and Theorem 4. 2 , we obtain the following corollary.

Corollary 4. 1 *The DDH-SE assumption is equivalent to both the DDH assumption and the DLSE assumption.*

5 Applications

In this section, we present variants of ElGamal encryption scheme and Cramer-Soup scheme. Each encryption algorithm uses a short random exponent r such that r is essentially c bits long, where $c = \omega(\log |q|)$ and q is the order of the underlying group.

Note that computing g^r requires at most $2c$ modulo multiplications in our variants while it requires at most $2n$ modulo multiplications in the original algorithms. Hence our variants are much faster than the original encryption algorithms.

It is clear that these variants are provably secure under the DDH-SE assumption. However, more than that, we can immediately see that they are provably secure under the two widely accepted assumptions, the DDH assumption and the DLSE assumption, from Corollary 4. 1 .

5.1 Application to ElGamal Encryption Scheme

It is well-known that ElGamal encryption scheme is IND-CPA under the DDH assumption. Now our variant of ElGamal encryption scheme is described as follows.

(Key generation) Choose a generator g at random. and let $\hat{g} = g^{n-c}$. Choose $x \in Z_q$ randomly and let $\hat{y} = \hat{g}^x$. The public key is $(\hat{g}, \hat{y})$ and the secret key is x.

(Encryption) Given a message $m \in G$, first choose r such that

$$2^{n-c} \leq 2^{n-c} r < q$$

randomly. Next compute $c_1 = \hat{g}^r, c_2 = m\hat{y}^r$. The ciphertext is (c_1, c_2).

(Decryption) Given a ciphertext (c_1, c_2), compute

$$c_2 / c_1^x = m.$$

Theorem 5. 1 *The above scheme is still IND-CPA under the DDH assumption and the DLSE assumption.*

(Proof) From Corollary 4. 1 , we can assume the DDH-SE assumption. Then $\hat{y}^r$ could be replaced by a random group element without changing significantly the behavior of the attacker. However, if we perform this substitution, the message m is perfectly hidden, which implies the security.

Q.E.D.

We can also prove the converse of Theorem 5. 1 easily.

5.2 Application to Cramer-Shoup Encryption Scheme

Similarly, we show our variant of Cramer-Shoup scheme. Cramer-Shoup scheme is IND-CCA under the DDH assumption [2].

(Key generation) Choose two generator g_1 and g_2 at random. Also Choose $x_1 x_2, y_1, y_2, z \in Z_q$ randomly. Let $\hat{g}_1 = g_1^{n-c}$ and $\hat{g}_2 = g_2^{n-c}$. Also let $\hat{c} = \hat{g}_1^{x_1}\hat{g}_2^{x_2}, \hat{d} = \hat{g}_1^{y_1}\hat{g}_2^{y_2}, \hat{h} = \hat{g}_1^{z}$.

Let H be a randomly chosen universal one-way hash function.

The public key is $(\hat{g}_1, \hat{g}_2, \hat{c}, \hat{d}, \hat{h}, H)$ and the secret key is $(x_1 x_2, y_1, y_2, z)$.

(Encryption) Given a message $m \in G$, first choose r such that

$$2^{n-c} \leq 2^{n-c} r < q$$

randomly. Next compute

$$u_1 = \hat{g}_1^r, u_2 = \hat{g}_2^r, e = \hat{h}^r m, \alpha = H(u_1, u_2, e), v = (\hat{c}\hat{d}^{\alpha})^r.$$

The ciphertext is (u_1, u_2, e, v).

(Decryption) Given a ciphertext (u_1, u_2, e, v), first compute $\alpha = H(u_1, u_2, e)$ and test if

$$u_1^{x_1 + y_1\alpha} u_2^{x_2 + y_2\alpha} = v.$$

If this condition does not hold, the decryption algorithm outputs "reject". Otherwise, it outputs $m = e/u_1^z$.

Theorem 5. 2 *The above scheme is still IND-CCA under the DDH assumption and the DLSE assumption.*

(Proof) Almost the same as the proof of [2]. Use Corollary 4. 1 .

Q.E.D.

References

[1] M. Blum and S. Micali: "How to generate cryptographically strong sequences of pseudo-random bits", SIAM Journal on Computing, Vol.13, No.4, pp.850–864 (1984)

[2] R. Cramer and V. Shoup: "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack", Advances in Cryptology – Crypto'98 Proceedings, Lecture Notes in Computer Science Vol.1462, Springer Verlag, pp.13–25 (1998)

[3] T. ElGamal: " A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Trans. Information Theory, Vol.IT-31, No.4, pp.469–472 (1985)

[4] R. Gennaro: "An improved pseudo-random generator based on discrete log", Advances in Cryptology – Crypto 2000 Proceedings, Lecture Notes in Computer Science Vol.1880, Springer Verlag, pp.469–481 (2000)

[5] D. L. Long and A. Wigderson: "The discrete logarithm hides $O(\log n)$ bits", SIAM Journal on Computing, Vol.17, No.2, pp.363–372 (1988)

[6] P. van Oorschot and M. Wiener: "On Diffie-Hellman key agreement with short exponents", Advances in Cryptology – Eurocrypt'96 Proceedings, Lecture Notes in Computer Science Vol.1070, Springer Verlag, pp.332–343 (1996)

[7] S. Patel and G. S. Sundaram: "An efficient discrete log pseudo random generator", Advances in Cryptology – Crypto'98 Proceedings, Lecture Notes in Computer Science Vol.1462, Springer Verlag, pp.304–317 (1998)

[8] R. Peralta: "Simultaneous security of bits in the discrete log", Advances in Cryptology – Eurocrypt'85 Proceedings, Lecture Notes in Computer Science Vol.219, Springer-Verlag, pp.62–72 (1986)

[9] C. Schnorr: "Security of almost all discrete log bits", Electronic Colloquium on Computational Complexity. TR-98-033. http:/www.eccc.uni-trier.de.eccc/

Appendix

A Proof of Inequality (1) in Lemma 3.1

First, recall n is the bit length of q and let $b_n||b_{n-1}||\cdots||b_1$ be the binary representation of q, where each b_i is in $\{0,1\}$. Then, the most significant bit of q (i.e., b_n) is always equal to 1. Thus, q can be written as $q = 2^{n-1} + q'$ where $0 \leq q' < 2^{n-1}$. Now s is assumed to be less than $2^{n-\log t}$. Thus, r such that $s + r \geq q$ satisfies that $r > 2^{n-1} - 2^{n-\log t}$. Since r is assumed to be chosen uniformly from R_{n-i}, the probability that $s + r \geq q$ is

$$\begin{aligned}
&\Pr[s + r \geq q] \\
&\quad < \Pr[r \geq 2^{n-1} - 2^{n-\log t}] \\
&\quad = \Pr[r \geq 2^{n-1} - 2^{n-\log t} | msb_1(r) = 1] \\
&\qquad\qquad \cdot \Pr[msb_1(r) = 1] \\
&\qquad + \Pr[r \geq 2^{n-1} - 2^{n-\log t} | msb_1(r) = 0] \\
&\qquad\qquad \cdot \Pr[msb_1(r) = 0] \\
&\quad < \Pr[r \geq 2^{n-1} - 2^{n-\log t} | msb_1(r) = 1] \\
&\qquad + \Pr[r \geq 2^{n-1} - 2^{n-\log t} | msb_1(r) = 0] \\
&\quad < 2 \cdot \Pr[r \geq 2^{n-1} - 2^{n-\log t} | msb_1(r) = 0] \\
&\quad = 2 \cdot 2^{1-\log t} \\
&\quad = 4/t.
\end{aligned}$$

Thus, the probability that $s + r < q$ is $(1 - 4/t)$ at least.

Q.E.D.