

Relationships among the Computational Powers of Breaking Discrete Log Cryptosystems

Kouichi Sakurai^{1*} and Hiroki Shizuya²

¹ Dept. of Computer Science and Communication Engineering, Kyushu University,
812-81 Japan. e-mail:sakurai@csce.kyushu-u.ac.jp

² ECIP & GSIS, Tohoku University, 980-77 Japan.
e-mail:shizuya@ecip.tohoku.ac.jp

Abstract. We investigate the complexity of breaking cryptosystems of which security is based on the discrete logarithm problem. We denote the algorithms of breaking the Diffie-Hellman's key exchange scheme by DH, the Bellare-Micali's non-interactive oblivious transfer scheme by BM, the ElGamal's public-key cryptosystem by EG, the Okamoto's conference-key sharing scheme by CONF, and the Shamir's 3-pass key-transmission scheme by 3PASS, respectively. We show a relation among these cryptosystems that

$$3PASS \leq_m^{\text{FP}} \text{CONF} \leq_m^{\text{FP}} \text{EG} \equiv_m^{\text{FP}} \text{BM} \equiv_m^{\text{FP}} \text{DH},$$

where \leq_m^{FP} denotes the polynomial-time functionally many-to-one reducibility, i.e. a function version of the \leq_m^p -reducibility. We further give some condition in which these algorithms have equivalent difficulty. Namely,

1. If the complete factorization of $p - 1$ is given, i.e. if the discrete logarithm problem is a certified one, then these cryptosystems are equivalent w.r.t. expected polynomial-time functionally Turing reducibility.
2. If the underlying group is the Jacobian of an elliptic curve over \mathbf{Z}_p with a prime order, then these cryptosystems are equivalent w.r.t. polynomial-time functionally many-to-one reducibility.

We also discuss the complexity of several languages related to those computing problems.

1 Introduction

1.1 Motivation

The discrete logarithm problem, DLP for short, is the problem that on input $y, g \in G$, outputs an integer x such that $y = g^x$, where G is some finite group with efficiently computable group law. A cryptosystem based on DLP is secure if the DLP is hard to solve. A typical DLP is the case where $G = \mathbf{Z}_p^*$ with p prime.

* A part of this work was done while the first author was working for Mitsubishi Electric Corp.

In 1976, Diffie and Hellman [DH76] first proposed a key exchange scheme that is secure if the DLP over \mathbf{Z}_p^* is hard to solve. A lot of cryptosystems based on DLP have been proposed to construct a public-key cryptosystem, an oblivious transfer protocol, a key-transmission scheme, a zero-knowledge proof of possession of information, and so on. It is clear that all these cryptosystems would no longer be secure if there were an efficient algorithm to solve the DLP, but no such algorithm is known to exist (see, e.g. [COS86, Odl84]). However, it is worth noting that the converse does not generally hold, i.e. it is not known that a polynomial-time algorithm to crack one of these cryptosystems implies feasibility of the DLP. Recently, a great progress has been made by Maurer toward the equivalence of the DLP and breaking the Diffie-Hellman scheme [Ma94], but the equivalence is not known to hold without assumption. Therefore, in general, all these cryptosystems could be breakable without solving the DLP.

In this paper, instead of studying whether there exists a cracking algorithm for cryptosystems without breaking DLP, we investigate the relation among such cryptosystems. Let S_1 and S_2 be two cryptosystems both based on some DLP. Our interest is whether S_1 remains secure even if a polynomial-time algorithm to break S_2 has been found, and vice versa. Although such discussion appears to be essential to clarify the security level of the cryptosystem, we know little about that, surprisingly.

1.2 Summary of Results

Let us denote the problems of breaking the Diffie-Hellman's key exchange scheme by DH, the Bellare-Micali's non-interactive oblivious transfer scheme [BeMi89] by BM, the ElGamal's public-key cryptosystem [ElG85] by EG, the Okamoto's conference-key sharing scheme [Oka88] by CONF and the Shamir's 3-pass key-transmission scheme [SRA79, Riv90] by 3PASS, respectively.

We first show a relation among these cryptosystems that

$$3\text{PASS} \leq_m^{\text{FP}} \text{CONF} \leq_m^{\text{FP}} \text{EG} \equiv_m^{\text{FP}} \text{BM} \equiv_m^{\text{FP}} \text{DH},$$

where \leq_m^{FP} denotes the polynomial-time functionally many to one reducibility. We further gives some condition in which these algorithms have equivalent w.r.t. certain reductions. Namely,

1. If the complete factorization of $p-1$ is given, i.e. if the the discrete logarithm problem is a certified one, then these cryptosystems are equivalent w.r.t. expected polynomial-time functionally Turing reduction, i.e., $3\text{PASS} \equiv_T^{\text{FEP}} \text{CONF} \equiv_I^{\text{FEP}} \text{EG} \equiv_m^{\text{FP}} \text{BM} \equiv_m^{\text{FP}} \text{DH}$.
2. If the underlying group is the Jacobian of an ordinary elliptic curve over \mathbf{Z}_p with a prime order, then these cryptosystems are equivalent w.r.t. polynomial-time functionally many-to-one reduction, i.e. $3\text{PASS} \equiv_m^{\text{FP}} \text{CONF} \equiv_m^{\text{FP}} \text{EG} \equiv_m^{\text{FP}} \text{BM} \equiv_m^{\text{FP}} \text{DH}$.

We will also investigate the complexity of languages associated with these problems. Let $L_{3\text{PASS}}$ be the language associated with 3PASS defined as

$$L_{3\text{PASS}} = \{((A, B, C, p), s) \mid 3\text{PASS}(A, B, C, p) = s\},$$

i.e. its membership problem is to recognize that the s is a correct answer to the instance (A, B, C, p) of 3PASS . Although $L_{3\text{PASS}}$ is not known to be in \mathcal{P} or \mathcal{BPP} , we show that if $L_{3\text{PASS}}$ is in \mathcal{P} , there is a probabilistic polynomial-time algorithm that reduces DH to 3PASS . Thus, if $L_{3\text{PASS}}$ is in \mathcal{P} , all the problems to crack these cryptosystems become equivalent.

In the same way, let L_{DH} be the language associated with DH defined as

$$L_{\text{DH}} = \{((A, B, g, p), C) \mid \text{DH}(A, B, g, p) = C\}.$$

Although L_{DH} is not known to be in \mathcal{P} or \mathcal{BPP} as observed in [B93], we show that L_{DH} is random self-reducible in the sense of [TW87], and therefore L_{DH} is in \mathcal{PZK} , the class of languages that have perfect zero-knowledge proof systems.

1.3 Computational Complexity, Communication Complexity, and Cryptographic Mechanisms

Complexity assumption is an important measure of security of cryptographic protocols. In general, a protocol with sophisticated mechanism requires stronger complexity assumption.

Impagliazzo and Rudich [IR89], in fact, presented evidence that secure secret key agreement protocols require stronger complexity assumption than the existence of one-way permutations.

The protocol with sophisticated mechanism also requires a number of interactions. Rudich [Rud91] constructed an oracle relative to which secret agreement can be done in k passes, but not in $k - 1$, and showed that there exists a 3-pass system based on an assumption which seems to be weaker than the existence of trapdoor functions.

We should note that the schemes discussed in this paper achieve different mechanisms and require different number of interactions. Thus, the results of this paper reveal relationships among computational complexity assumption, round complexity, and mechanisms in cryptographic protocols based on the discrete logarithms.

2 Preliminaries

2.1 Cryptosystems based on DLP

We give a brief review of the cryptosystems considered in this paper. All those are based on the discrete logarithm problem (DLP). To avoid complicated generalization of DLP defined over an generic finite group, we restrict ourselves to the case where the underlying group is \mathbf{Z}_p^* with p prime. Thus, the DLP is now the problem that on input y, g, p , outputs x such that $y \equiv g^x \pmod{p}$. Here g does not necessarily generate \mathbf{Z}_p^* . For notational convenience, we will write simply g^x rather than $g^x \pmod{p}$, etc.

We will refer to Alice and Bob as two parties, respectively, that follow the scheme and communicate with each other.

Diffie-Hellman's Key Exchange Scheme [DH76]

Alice and Bob agree on p and the base $g \in \mathbf{Z}_p^*$ before starting their communication. Alice picks a randomly from \mathbf{Z}_{p-1} , computes $A = g^a$, and sends A to Bob. Bob picks b randomly from \mathbf{Z}_{p-1} , computes $B = g^b$, and sends B to Alice. Alice computes $C = B^a$ and Bob computes $C = A^b$.

Bellare-Micali's Non-Interactive Oblivious Transfer Scheme [BeMi89]

Alice and Bob agree on p and the base $g \in \mathbf{Z}_p^*$ and some $C \in \mathbf{Z}_p^*$. Bob picks $i \in \{0, 1\}$ at random, $x_i \in \mathbf{Z}_{p-1}$, and sets $\beta_i = g^{x_i}$ and $\beta_{1-i} = C \cdot (g^{x_i})^{-1}$. Bob publishes (β_0, β_1) as his public key whereas he keeps (i, x_i) as his secret key. Suppose Alice wants to send Bob one of the strings (s_0, s_1) in an oblivious transfer manner. Alice picks at random $y_0, y_1 \in \mathbf{Z}_{p-1}$ and sends $\alpha_0 = g^{y_0}$, $\alpha_1 = g^{y_1}$ to Bob. Alice then computes $\gamma_0 = \beta_0^{y_0}$ and $\gamma_1 = \beta_1^{y_1}$, and sends $r_0 = s_0 \oplus \gamma_0$ and $r_1 = s_1 \oplus \gamma_1$ to Bob, where \oplus designates the bitwise addition mod 2.

On receiving α_0 and α_1 , Bob uses his secret key to compute $\alpha_i^{x_i} = \gamma_i$. He then computes $\gamma_i \oplus r_i = s_i$.

ElGamal's Public-Key Cryptosystem [ElG85]

Bob sets $g \in \mathbf{Z}_p^*$ as the base, picks $x \in \mathbf{Z}_{p-1}$ at random, and computes $y = g^x$. Bob publishes y, g, p as his public key whereas he keeps x as his secret key. Suppose Alice wants to send a string m to Bob. Alice picks $r \in \mathbf{Z}_{p-1}$ at random, computes $C_1 = g^r$, $C_2 = my^r$ and sends (C_1, C_2) to Bob. On receiving (C_1, C_2) , Bob uses his secret key to compute $m = C_2 / (C_1)^x$.

Okamoto's Conference-Key Sharing Scheme [Oka88]

Alice and Bob agree on p and the base $g \in \mathbf{Z}_p^*$ before starting their communication. Alice picks a randomly from \mathbf{Z}_{p-1}^* , computes $A = g^a$, and sends A to Bob. Bob picks b randomly from \mathbf{Z}_{p-1} , computes $B = A^b$, and sends B to Alice. Alice computes $C = B^{a^{-1}}$ and Bob computes $C = g^b$.

We shall note that the established key depends only on Bob's randomness b . Thus Bob can decide the value of the key g^b by himself although Bob can not send directly a message. This property has an advantage over Diffie-Hellman's key exchange scheme in the case of a conference-key sharing scheme for multiple users [Oka88].

Shamir's 3-Pass Message Transmission Scheme [SRA79]

This is also called the Massey-Omura's cryptosystem (see, e.g. [Kob87b]), and originally proposed as a tool for mental poker by Shamir et al. [SRA79, Riv90]. Alice and Bob agree on p before their communication. Suppose Alice wants to send a string (message) s to Bob. Alice picks $a \in \mathbf{Z}_{p-1}^*$ at random, computes $A = s^a$, and sends A to Bob. On receiving A , Bob picks $b \in \mathbf{Z}_{p-1}^*$ at random, computes $C = A^b$, and sends C to Alice. On receiving C , Alice uses her secret a to compute $B = C^{a^{-1}}$ and sends B to Bob. On receiving B , Bob uses his secret b to compute $s = B^{b^{-1}}$.

Remark.

Shamir's 3-pass key transmission scheme is useful not only for secret message transferring but also for an oblivious transfer [Ra81]. An oblivious transfer is a protocol satisfying the following three conditions.

1. Alice can send any message m_0 or m_1 .
2. Bob gets only one of message m_0 or m_1 .
3. Alice cannot know which message, m_0 or m_1 Bob obtains.

However, certain attacks (on the third condition above) were pointed out (e.g. [Co85]). Shamir et al. [SRA79, Ra81] applied the protocol above into shuffling cards together among two parties in an electronic poker game. Thus, we consider that the Shamir's 3-pass has a more sophisticated mechanism than an oblivious transfer. The protocol is as follows:

Before starting the protocol, A (Alice) and B (Bob) agree on a prime p .

1. For two message m_0 and m_1 , A randomly picks $a \in \mathbf{Z}_{p-1}^*$, computes $\alpha_0 = m_0^a$ and $\alpha_1 = m_1^a$, and sends (α_0, α_1) to B .
2. B picks $e \in \{0, 1\}$ and randomly selects $b \in \mathbf{Z}_{p-1}^*$, then computes $\beta_0 = \alpha_0^b$, and sends β to A .
3. A computes $\gamma = \beta^{a^{-1}}$, and sends it to B .
4. B obtains m_e by computing $\gamma^{b^{-1}}$.

2.2 Definitions of Problems

We give the formal definitions of the problems to crack the cryptosystems considered in this paper. These problems will be formalized as something like functions from some tuple of Σ^* 's to Σ^* , where Σ^* is the set of all possible strings over the finite alphabet $\Sigma = \{0, 1\}$.

$\text{DLP}(y, g, p)$ is the problem that on input p prime and $y, g \in \mathbf{Z}_p^*$, outputs $x \in \mathbf{Z}_{p-1}$ such that $y = g^x$ if such an x exists.

$\text{DH}(A, B, g, p)$ is the problem that on input p prime and $A, B, g \in \mathbf{Z}_p^*$, outputs $C \in \mathbf{Z}_p^*$ such that $C = g^{ab}$, $A = g^a$ and $B = g^b$ if such a C exists.

$\text{BM}((\alpha_0, \alpha_1), (r_0, r_1), C, (\beta_0, \beta_1), g, p)$ is the problem that on input p prime and $\alpha_0, \alpha_1, r_0, r_1, C, \beta_0, \beta_1, g \in \mathbf{Z}_p^*$ with $\beta_0\beta_1 = C$, outputs one of (s_0, s_1) such that $s_i = \gamma_i \oplus r_i$, $\gamma_i = g^{x_i y_i}$, $\alpha_i = g^{y_i}$, $\beta_i = g^{x_i}$ if such an s_i exists ($i = 0$ or 1).

$\text{EG}(C_1, C_2, y, g, p)$ is the problem that on input p prime and $C_1, C_2, y, g \in \mathbf{Z}_p^*$, outputs $m \in \mathbf{Z}_p^*$ such that $C_2 = mg^{x^r}$, $y = g^x$, $C_1 = g^r$ if such an m exists.

$\text{CONF}(A, B, g, p)$ is the problem that on input p prime and $A, B, g \in \mathbf{Z}_p^*$, outputs $C \in \mathbf{Z}_p^*$ such that $A = g^a$ where $a \in \mathbf{Z}_{p-1}^*$, $B = A^b$ where $b \in \mathbf{Z}_{p-1}$ and $C = g^b$ if such an C exists.

$\text{3PASS}(A, B, C, p)$ is the problem that on input p prime and $A, B, C \in \mathbf{Z}_p^*$, outputs s such that $A = s^a$, $B = s^b$, $C = s^{ab}$ and $a, b \in \mathbf{Z}_{p-1}^*$ if such an s exists.

The functions above always returns a correct answer if there is a solution to the query. However, there are no mentions of the behavior in the case when there is no solution to the query. However, we consider stronger functions which output \perp if there are no solutions, where \perp is the special string to designate the status that the function has no returnable value (Theorem 5).

2.3 Reducibility

In order to compare the relative complexity of different functions, we use the concept of *reducibility*. Intuitively a function f is reducible to another function g if the value of the first function f is computed by an algorithm which uses an algorithm for the second function g as a subroutine. We will consider three types of such reducibilities based on the types of subroutines.

Definition 1. A function f is polynomial-time functionally Turing reducible to a function g (in symbols $f \leq_T^{\text{FP}} g$) if a polynomial-time oracle Turing machine with access to values of g can compute f . Regarding the complexity of such an algorithm we suppose that the cost of one calling the oracle B is just one step.

Definition 2. A function f is *expected* polynomial-time functionally Turing reducible to a function g (in symbols $f \leq_T^{\text{FEP}} g$) if an expected polynomial-time oracle Turing machine with access to values of g can compute f . (NOTE: This paper says that a machine M is *expected polynomial-time* if there exists an $\epsilon > 0$ such that, for all $x \in \{0, 1\}^*$, the expectation, taken over the infinite bit sequences r , of $(t_M(x, r))^\epsilon$ is bounded above by $|x|$ (i.e., $E((t_M(x, r))^\epsilon) \leq |x|$).

Definition 3. A function f is polynomial-time functionally many-one reducible to a function g (in symbols $f \leq_m^{\text{FP}} g$) if there exists a pair of polynomial-time computable functions h_1, h_2 such that for every input string x , $f(x) = h_2(g(h_1(x)))$.

3 Main Results

3.1 Relationships among the Cryptosystems

We first show the following relation among these cryptosystems.

Theorem 4. $3\text{PASS} \leq_m^{\text{FP}} \text{CONF} \leq_m^{\text{FP}} \text{EG} \equiv_m^{\text{FP}} \text{BM} \equiv_m^{\text{FP}} \text{DH} \leq_m^{\text{FP}} \text{DLP}$.

Proof. Since it is clear that $\text{DH} \leq_m^{\text{FP}} \text{DLP}$, we show that $3\text{PASS} \leq_m^{\text{FP}} \text{CONF}$, $\text{CONF} \leq_m^{\text{FP}} \text{EG}$, $\text{EG} \equiv_m^{\text{FP}} \text{DH}$, and $\text{BM} \equiv_m^{\text{FP}} \text{DH}$.

$3\text{PASS} \leq_m^{\text{FP}} \text{CONF}$:

Let $(A, B, C, p) = (s^a, s^b, s^{ab}, p)$ be an instance of 3PASS.

$3\text{PASS}(A, B, C, p) = \text{CONF}(C, A, B, p) = \text{CONF}((s^b)^a, (s^b)^{b^{-1}a}, s^b, p) = (s^b)^{b^{-1}} = s$.

$\text{CONF} \leq_m^{\text{FP}} \text{EG}$:

Let $(A, B, C, p) = (g^a, g^{ab}, g, p)$ be an instance of CONF.

$$\text{CONF}(A, B, C, p) = 1/\text{EG}(g, 1, g^{ab}, g^b, p).$$

$\text{3PASS} \leq_m^{\text{FP}} \text{EG}$:

Let $(A, B, C, p) = (s^a, s^b, s^{ab}, p)$ be an instance of 3PASS. First we compute (C_1, C_2, y, g, p) , an instance of EG, by (CA, ABC, CB, C, p) . Here,

$$\begin{aligned} (C_1, C_2, y, g, p) &= (CA, ABC, CB, C, p) \\ &= (s^{a(b+1)}, s^{ab+a+b}, s^{b(a+1)}, s^{ab}, p) \\ &= ((s^{ab})^{b^{-1}(b+1)}, s^{ab+a+b}, (s^{ab})^{a^{-1}(a+1)}, s^{ab}, p). \end{aligned}$$

Thus,

$$\begin{aligned} m &= \text{EG}(CA, ABC, CB, C, p) \\ &= s^{ab+a+b}/(s^{ab})^{a^{-1}b^{-1}(a+1)(b+1)} \\ &= s^{ab+a+b}/s^{ab+a+b+1} = s^{-1}. \end{aligned}$$

This implies that the oracle returns $m = s^{-1}$. Therefore, we get $s = m^{-1}$, which is computed in time polynomial in $|p|$.

$\text{EG} \leq_m^{\text{FP}} \text{DH}$:

This is a trivial reduction. Let $(C_1, C_2, y, g, p) = (g^r, mg^{xr}, g^x, g, p)$ be an instance of EG. Since the oracle DH returns g^{xr} to the query (C_1, y, g, p) , m is immediately computed by $m = C_2/g^{xr}$.

$\text{DH} \leq_m^{\text{FP}} \text{EG}$ [Oka94]:

Let $(A, B, g, p) = (g^a, g^b, g, p)$ be an instance of DH. g^{ab} is the inverse of the answer of the oracle EG to the query $(A, 1, B, g, p)$.

$\text{BM} \equiv_m^{\text{FP}} \text{DH}$:

It is not hard to see that $\text{BM} \leq_m^{\text{FP}} \text{DH}$ because DH returns $\gamma_i = g^{x_i y_i}$ to the query $(\alpha_i, \beta_i, g, p)$, and s_i is computed by $s_i = \gamma_i \oplus r_i$. Conversely, for $(A, B, g, p) = (g^a, g^b, g, p)$, an instance of DH, we let

$$((\alpha_0, \alpha_1), (r_0, r_1), C, (\beta_0, \beta_1), g, p) = ((A, g^u), (0, 0), t, (B, tB^{-1}), g, p),$$

where u, t are picked randomly from \mathbf{Z}_{p-1} and \mathbf{Z}_p^* , respectively. Since we set $r_0 = 0$, the oracle $\text{BM}((A, g^u), (0, 0), t, (B, tB^{-1}), g, p)$ returns $s_0 = r_0 \oplus \gamma_0 = 0 \oplus g^{ab} = g^{ab}$.

This completes the proof. ■

We give a simple alternative proof for $\text{3PASS} \leq_m^{\text{FP}} \text{DH}$. It holds that for an instance $(A, B, C, p) = (s^a, s^b, s^{ab}, p)$ of 3PASS,

$$\text{DH}(A, B, C, p) = \text{DH}((s^{ab})^{b^{-1}}, (s^{ab})^{a^{-1}}, s^{ab}, p) = (s^{ab})^{a^{-1}b^{-1}} = s.$$

We do not know if $\text{EG} \leq_m^{\text{FP}} \text{3PASS}$. However, if we consider more stronger cracking algorithms which answer the special symbol “ \perp ” when there is no solution to the instance, we obtain a further result. Consider the following function:

$3\text{PASS}^*(A, B, C, p)$ is the problem that on input p prime and $A, B, C \in \mathbf{Z}_p^*$, outputs s such that $A = s^a$, $B = s^b$, $C = s^{ab}$ and $a, b \in \mathbf{Z}_{p-1}^*$ if such an s exists. Otherwise it outputs \perp .

Theorem 5. $\text{DH} \stackrel{\text{FEF}}{\leq_T} 3\text{PASS}^*$.

Proof. Let $(A, B, g, p) = (g^a, g^b, g, p)$ be an instance of DH. For randomly picked $u, v \in \mathbf{Z}_{p-1}$, an instance of 3PASS^* is computed by

$$(Ag^u, Bg^v, g, p) = (g^{a+u}, g^{b+v}, g, p).$$

If both $a+u$ and $b+v$ happen to be in \mathbf{Z}_{p-1}^* , the oracle $3\text{PASS}^*(g^{a+u}, g^{b+v}, g, p)$ returns $g^{(a+u)(b+v)}$ because

$$\begin{aligned} g^{a+u} &= (g^{(a+u)(b+v)})^{(b+v)^{-1}}, \\ g^{b+v} &= (g^{(a+u)(b+v)})^{(a+u)^{-1}}, \\ g &= (g^{(a+u)(b+v)})^{(a+u)^{-1}(b+v)^{-1}}. \end{aligned}$$

Thus $g^{ab} = g^{(a+u)(b+v)} / (A^v B^u g^{uv})$.

However, if either $a+u$ or $b+v$ is not in \mathbf{Z}_{p-1}^* , the oracle $3\text{PASS}^*(g^{a+u}, g^{b+v}, g, p)$ returns some s or \perp . We show that if $3\text{PASS}^*(g^{a+u}, g^{b+v}, g, p)$ returns s , then $s = g^{(a+u)(b+v)}$. If the oracle returns s , it satisfies that for some $\alpha, \beta \in \mathbf{Z}_{p-1}^*$,

$$s^\alpha = g^{a+u}, \quad s^\beta = g^{b+v}, \quad s^{\alpha\beta} = g.$$

Thus, over $\mathbf{Z}_{\text{ord}(g)}$,

$$r\alpha = a+u, \quad r\beta = b+v, \quad r\alpha\beta = 1,$$

where $\text{ord}(g)$ designates the order of g , and r is an element in $\mathbf{Z}_{\text{ord}(g)}$ such that $s = g^r \pmod p$. Then, we have that $(a+u)(b+v) = r^2\alpha\beta = r(r\alpha\beta) = r$. Thus, $s = g^r = g^{(a+u)(b+v)}$.

Conversely, if no such r, α, β exist, the oracle returns \perp . Therefore, another $u, v \in \mathbf{Z}_{p-1}$ should be picked, and this is repeated until the oracle returns a string other than \perp .

To summarize, the Algorithm 1 named DHto3PASS solves DH using the oracle 3PASS^* .

Now we estimate how many times the while-statement is repeated. The probability ρ that the oracle returns a string other than \perp to a query is greater than the probability that both $a+u$ and $b+v$ are in \mathbf{Z}_{p-1}^* . Thus, $\rho \geq (\varphi(p-1)/(p-1))^2$, where φ is the Euler's totient function. Since $\varphi(n) \leq \ln(2) \cdot n/\ln(2n)$ for a positive integer n [Rib88], the expected number of repetition of the while-statement is less than $(\ln(2(p-1))/\ln(2))^2$, which is bounded by a polynomial in $|p|$. Thus, DH reduces to 3PASS^* in probabilistic polynomial-time.

This completes the proof. ■

% Algorithm 1

% DHto3PASS

input A, B, g, p

$s := \perp$

while ($s = \perp$) do

 pick $u, v \in \mathbf{Z}_{p-1}$ at random

$A' := Ag^u; B' := Bg^v$

$s := 3\text{PASS}^*(A', B', g, p)$

end while

$C := s / (A^v B^u g^{uv})$

output C

end

Remark. The Algorithm 1 above does not give the answer “ \perp ” even when the input of DH has no solution. So, we do not know if $\text{DH}^* \leq_T^{\text{FEP}} \text{3PASS}^*$. However, we can obtain a polynomial-time reduction from DH^* to 3PASS^* with one-sided error by terminating the algorithm DHto3PASS within a suitable step, as shown in Algorithm 2.

```

% Algorithm 2
% DHto3PASS with one-sided error
input  $A, B, g, p$ 
 $s := \perp; C := \perp; i := 1$ 
 $T := q(|p|)$  % some polynomial in  $|p|$ 
while ( $[s = \perp] \wedge [i \leq T]$ ) do
    pick  $u, v \in \mathbf{Z}_{p-1}$  at random
     $A' := Ag^u; B' := Bg^v$ 
     $s := \text{3PASS}^*(A', B', g, p)$ 
     $i := i + 1$ 
end while
if  $s \neq \perp$  then  $C := s / (A^v B^u g^{uv})$ 
output  $C$ 
end

```

We do not know if $\text{DH} \equiv \text{DH}^*$ nor $\text{3PASS} \equiv \text{3PASS}^*$ because there are no known efficient algorithms to check the answers of these cracking algorithms DH and 3PASS. Nevertheless, we show that DH is reducible to 3PASS over some special discrete logarithms.

3.2 The Case of Certified Discrete Logarithms

First we show that if the complete factorization of $p - 1$ is given and the base is a generator of \mathbf{Z}_p^* , i.e. if the discrete logarithm problem is a certified one, there is a probabilistic polynomial-time algorithm that solves DH using 3PASS as an oracle. This reduces DH to 3PASS, and the above reductions become equivalent.

Theorem 6. *If the complete factorization of $p - 1$ with p prime is given and the base g is a generator of \mathbf{Z}_p^* ,*

$$\text{DH} \leq_T^{\text{FEP}} \text{3PASS}.$$

Proof. In the proof of Theorem 5, we have shown that DH^* reduces to 3PASS^* , where 3PASS^* is an algorithm which returns a special symbol “ \perp ” if and only if there is no solution. Now we consider a weaker algorithm which returns any polynomially bounded string instead of \perp . However, this happens if either $a + u$ or $b + v$ is not in \mathbf{Z}_{p-1}^* . Thus, if we restrict ourselves to the query such that both $a + u$ and $b + v$ are in \mathbf{Z}_{p-1}^* , and if the instance of DH is appropriate, then the answer from the oracle is always correct. Therefore, we modify the algorithm DHto3PASS as shown in Algorithm 3.

Here, $d = \text{true}$ if and only if both X and Y are generators of \mathbf{Z}_p^* , which implies that both $a + u$ and $b + v$ are in \mathbf{Z}_{p-1}^* . The expected number of repetition of the while-statement is bounded by $(\ln(2(p-1))/\ln(2))^2$, which is also bounded by a polynomial in $|p|$. ■

```

% Algorithm 3
% DHto3PASS for Certified DLP
input A, B, g, p = p_0^{e_0} p_1^{e_1} \cdots p_k^{e_k} + 1
d := false
while (d = false) do
  pick u, v \in \mathbf{Z}_{p-1} at random
  X := Ag^u; Y := Bg^v
  d := [\bigwedge_{i=0}^k X^{(p-1)/p_i} \neq 1] \wedge
       [\bigwedge_{i=0}^k Y^{(p-1)/p_i} \neq 1]
end while
s := 3PASS(X, Y, g, p)
C := s / (A^v B^u g^{uv})
output C
end

```

den Boer [dB88] showed that the Diffie-Hellman problem is as strong as the discrete logarithms for certain primes. It is remarkable that Maurer [Ma94] made this result stronger to cover generic cyclic groups. Let $\varphi(N)$ be the order of the group \mathbf{Z}_N^* .

Theorem 7 [dB88], see also [Ma94]. *If $\varphi(p-1)$ is smooth, i.e., it consists of small prime factors with respect to a fixed polynomial in $q(|p|)$, then $\text{DLP} \leq_T^{\text{FEP}} \text{DH}$.*

We should note that our reductions keep the modulus, then the following is induced.

Corollary 8. *Suppose that $\varphi(p-1)$ is smooth, i.e., it consists of small prime factors with respect to a fixed polynomial in $q(|p|)$. If the complete factorization of $p-1$ with p prime is given and the base g is a generator of \mathbf{Z}_p^* , then*

$$3\text{PASS} \equiv_T^{\text{FEP}} \text{CONF} \equiv_T^{\text{FEP}} \text{EG} \equiv_T^{\text{FEP}} \text{BM} \equiv_T^{\text{FEP}} \text{DH} \equiv_T^{\text{FEP}} \text{DLP}.$$

3.3 The Case of Elliptic Discrete Logarithms

Next we consider these cryptosystems based on the elliptic-curve discrete logarithm problem [Kob87a, Mil85], denoted by EDLP.

Here we briefly review the EDLP. Let $C(a, b)_p$ be an elliptic curve defined over \mathbf{Z}_p , where p prime $\neq 2, 3$, with parameters $a, b \in \mathbf{Z}_p$, that is

$$C(a, b)_p = \{(x, y) \in \mathbf{Z}_p \times \mathbf{Z}_p \mid [y = x^3 + ax + b] \wedge [a, b \in \mathbf{Z}_p] \wedge [4a^3 + 27b^2 \not\equiv 0 \pmod{p}]\} \cup \{O\},$$

where O is the point at infinity. The Jacobian of $C(a, b)_p$, which happens to be the same as $C(a, b)_p$, forms an abelian group. The EDLP is the problem that on input a point $Q \in C(a, b)_p$ and the base point $P \in C(a, b)_p$, outputs m such that

$Q = mP$ if such an m exists. Here, we denote by mP the m -time addition of the point P . The order of $C(a, b)_p$, denoted by $\#C$, is computed in time polynomial in $|p|$ [Sch85]. The order is bounded as $-2\sqrt{p} \leq \#C(a, b)_p - (p + 1) \leq 2\sqrt{p}$.

The elliptic curve $C(a, b)_p$ defined over \mathbf{Z}_p is said to be supersingular if and only if $\#C(a, b) = p + 1$. Nonsupersingular elliptic curves are called ordinary. Thus an elliptic curve group with prime order is ordinary and simple, where by a simple group we mean that there is no non-trivial normal subgroup in $C(a, b)_p$. If $C(a, b)_p$ is supersingular, the EDLP reduces in probabilistic polynomial-time to a discrete logarithm problem over the multiplicative group of a certain extension field of \mathbf{Z}_p [MOV91]. However, no such reduction algorithm is known to exist for elliptic-curve groups with prime order [Miy91].

It is not hard to see all the cryptosystems considered in this paper can actually be constructed over $C(a, b)_p$ as analogues of those over \mathbf{Z}_p^* , and the reductions shown in Theorem 4 also hold for the EDLP-based systems. Let DH_E (resp. $\text{BM}_E, \text{EG}_E, \text{CONF}_E, \text{3PASS}_E$) designate the EDLP-based DH (resp. BM, EG, CONF, 3PASS) problem. We have the following theorem.

Theorem 9. *If the cryptosystems are based on the discrete logarithm problem whose underlying group is the Jacobian of an elliptic curve defined over \mathbf{Z}_p with prime order, then*

$$\text{3PASS}_E \equiv_m^{\text{FP}} \text{CONF}_E \equiv_m^{\text{FP}} \text{EG}_E \equiv_m^{\text{FP}} \text{BM}_E \equiv_m^{\text{FP}} \text{DH}_E.$$

Proof. As Theorem 4, it is easily seen that $\text{3PASS}_E \leq_m^{\text{FP}} \text{CONF}_E \leq_m^{\text{FP}} \text{EG}_E \equiv_m^{\text{FP}} \text{BM}_E \equiv_m^{\text{FP}} \text{DH}_E$. Thus, it suffices to show that $\text{DH}_E \leq_m^{\text{FP}} \text{3PASS}_E$. Let E be an elliptic curve defined over \mathbf{Z}_p with p prime $\neq 2, 3$, and let $\#E = q$ with q prime. For an instance $(A, B, P, E, p) = (aP, bP, P, E, p)$ of DH_E , if $A \neq O$ and $B \neq O$, then both a and b are units in \mathbf{Z}_q . This is because E is simple. Thus, the oracle 3PASS_E always returns the correct answer to a query (A, B, P, E, p) . Hence, $\text{DH}_E \leq_m^{\text{FP}} \text{3PASS}_E$. ■

There are few known research on the distribution of the prime-order elliptic curves over all elliptic curves. A construction of the prime-order elliptic curves is studied also in [Miy91], and finding more efficient algorithms to construct such ordinary elliptic curves is an interesting future topic. Thus, the previously known merit of ordinary elliptic curves over \mathbf{Z}_p is just that it is immune from the attack by [MOV91]. Our theorem above is based on another interesting property of ordinary prime-order elliptic curves over \mathbf{Z}_p that any non-zero element has the inverse.

3.4 Languages Associated with the Cryptosystems

We return to the cryptosystems based on DLP defined over \mathbf{Z}_p^* .

Associated with the problems Q , we define the language L_Q by

$$L_Q = \{(x, y) \mid Q(x) = y\},$$

where Q is one of DLP, DH, BM, EG, CONF, or 3PASS. The problem to decide membership in L_Q is to recognize that y is an answer to the instance x of Q . Clearly,

these languages are in $\mathcal{NP} \cap \text{co-}\mathcal{NP}$. Indeed, L_{DLP} is in \mathcal{P} . However, it is not known that one of L_{DH} , L_{BM} , L_{EG} , L_{CONF} , or L_{3PASS} is in \mathcal{P} or \mathcal{BPP} . The same observation on L_{DH} can also be found in [B93]. Thus, there may be a reduction sequence among these languages which is different from the reductions given in Theorem 4, though, at the moment, no reductions among L_{DH} , L_{BM} , L_{EG} , L_{CONF} , and L_{3PASS} are known.

One connection to the reductions among the cracking problems is shown in the following.

Theorem 10. L_{3PASS} is not in \mathcal{P} unless $\text{DH} \leq_T^{\text{FEP}} \text{3PASS}$.

Proof. We show the contraposition. That is, if L_{3PASS} is in \mathcal{P} , $\text{DH} \leq_T^{\text{FEP}} \text{3PASS}$. The algorithm DHto3PASS in the proof of Theorem 5 can be modified, if L_{3PASS} is in \mathcal{P} , as shown in Algorithm 4. This completes the proof. ■

```

% Algorithm 4
input  $A, B, g, p$ 
 $d := \text{false}$ 
while ( $d = \text{false}$ ) do
    pick  $u, v \in \mathbf{Z}_{p-1}$  at random
     $A' := Ag^u; B' := Bg^v$ 
     $s := \text{3PASS}(A', B', g, p)$ 
     $d := [((A', B', g, p), s) \in L_{\text{3PASS}}]$ 
end while
 $C := s / (A^v B^u g^{uv})$ 
output  $C$ 
end

```

The theorem above gives a characterization of the complexity of L_{3PASS} . Also we obtain

Corollary 11. If L_{3PASS} is in \mathcal{P} , then $\text{3PASS} \equiv_T^{\text{FEP}} \text{CONF} \equiv_T^{\text{FEP}} \text{EG} \equiv_m^{\text{FP}} \text{BM} \equiv_m^{\text{FP}} \text{DH}$.

Theorem 12. The language L_{DH} is random self-reducible in the sense of [TW87].

Proof. For an instance $((A, B, g, p), C)$, let $A' = Ag^r$, $B' = Bg^s$, and $C' = CA^s B^r g^{rs}$ to make another instance $((A', B', g, p), C')$, where r and s are randomly picked from \mathbf{Z}_{p-1} . Note that if $A = g^a$, $B = g^b$ and $C = g^{ab}$, then $A' = g^{a+r}$, $B' = g^{b+s}$, and $C' = g^{(a+r)(b+s)}$. Hence, the distribution of A' (resp. B', C') is exactly the same as that of A (resp. B, C). It is clear that $((A', B', g, p), C')$ is in L_{DH} , so is $((A, B, g, p), C)$. This implies L_{DH} is random self-reducible. ■

The theorem above implies that L_{DH} has a perfect zero-knowledge interactive proof.

3.5 Single-Use versus Multiple-Use in Cryptosystems

Consider the situation that we use the Shamir's 3-pass scheme for transferring the same message s polynomially many times. In such a case, an adversary can get much information than single use. We discuss the relative security between single-use and multiple-use in the cryptosystem. So, we formulate the following k -3PASS problem.

k -3PASS is the problem that on input p prime and $A_1, B_1, C_1, \dots, A_k, B_k, C_k, \in \mathbf{Z}_p^*$, s such that $A_j = s^{a_j}$, $B_j = s^{b_j}$, $C_j = s^{a_j b_j}$ and $a_j, b_j \in \mathbf{Z}_{p-1}^*$ ($j = 1, \dots, k$) if such an s exists.

We show that multiple-use is as secure as single use.

Theorem 13. $1\text{-3PASS} (= 3\text{PASS}) \leq_m^{\text{FP}} k\text{-3PASS}$

Proof. Let (A, B, C, p) be an instance of 1-3PASS. Pick $(u_1, v_1), \dots, (u_k, v_k) \in \mathbf{Z}_{p-1}^* \times \mathbf{Z}_{p-1}^*$ at random. Put

$$A_i = A^{u_i}, B_i = B^{v_i}, C_i = C^{u_i v_i} (1 \leq i \leq k).$$

Then, $((A_1, B_1, C_1, p), \dots, (A_k, B_k, C_k, p))$ is an instance of k -3PASS. ■

The theorem above suggests a role of the randomness of each party in the scheme. The same property holds in some other cryptosystems, namely k -EG and k -CONF defined as follows.

k -EG is the problem that on input p prime and $C_{11}, C_{21}, \dots, C_{1k}, C_{2k}, y, g \in \mathbf{Z}_p^*$, outputs $m \in \mathbf{Z}_p^*$ such that $C_{2j} = mg^{xr}$, $y = g^x$, $C_{1j} = g^{rj}$ ($j = 1, \dots, k$) if such an m exists.

k -CONF is the problem that on input p prime and $A_1, \dots, A_k, B, g \in \mathbf{Z}_p^*$, outputs $C \in \mathbf{Z}_p^*$ such that $A = g^{a_j}$ where $a_j \in \mathbf{Z}_{p-1}^*$, $B = A_j^b$ where $b \in \mathbf{Z}_{p-1}$ ($j = 1, \dots, k$) and $C = g^b$ if such an C exists.

Theorem 14. $1\text{-EG}(= \text{EG}) \leq_m^{\text{FP}} k\text{-EG}$

Proof. Let (C_1, C_2, y, g, p) be an instance of 1-EG. We show that for any $k \leq q(|p|)$ with q polynomial, this can be transformed into an instance of k -EG in polynomial-time. First, pick $u_1, \dots, u_k \in \mathbf{Z}_{p-1}$ at random. Then, put

$$C_{1i} = C_1 g^{u_i}, C_{2i} = C_2 y^{u_i} (1 \leq i \leq k).$$

Since $C_{1i} = g^{r+u_i}$ and $C_{2i} = mg^{x(r+u_i)}$, we now have an instance of k -EG as $((C_{11}, C_{21}, y, g, p), \dots, (C_{1k}, C_{2k}, y, g, p))$. ■

Okamoto [Oka88] observed such a property in his scheme.

Theorem 15 [Oka88]. $1\text{-CONF}(= \text{CONF}) \leq_m^{\text{FP}} k\text{-CONF}$

4 Concluding Remarks

We have given the reductions among the problems to break some cryptosystems based on the discrete logarithms over \mathbf{Z}_p^* (Theorem 4). Specifically, we have shown that these problems are equivalent under the stronger function model (Theorem 5), although none of them is known to be equivalent to the discrete logarithm problem itself.

We have also shown that the equivalence occurs if the discrete logarithm problem is a certified one over \mathbf{Z}_p^* (Theorem 6), or if it is the elliptic-curve discrete logarithm problem associated with an ordinary elliptic curve defined over \mathbf{Z}_p (Theorem 9). Therefore, if one cryptosystem is breakable, so are the others. This means that if one wants to crack one of the cryptosystems, there are several possible approaches to the algorithm for breaking the target cryptosystem. However, this also implies that one cryptosystem is as secure as the others, namely, the provable security of the cryptosystems. Although those theorems can be interpreted in two ways as above, it is true that they give an interesting aspect of the cryptosystems based on the certified discrete logarithm or the ordinary elliptic-curve discrete logarithm.

Further, we have defined some languages associated with those problems. We have pointed out that each language to recognize the correct answer of the problem is not known to be in \mathcal{P} , whereas the language corresponding to the discrete logarithm problem is in \mathcal{P} . Some questions remain open:

- Does $L_{3\text{PASS}}$ reduce to L_{DH} with respect to \leq_T^p -reducibility?
- Does L_{DH} reduce to $L_{3\text{PASS}}$ with respect to \leq_T^p -reducibility?
- Does $L_{3\text{PASS}}$ have a perfect zero-knowledge interactive proof?

Acknowledgments

We would like to thank the following people. Toshiya Itoh pointed out a flaw of a mathematical formula in an earlier version of this paper. Kojiro Kobayashi gave us invaluable comments on the (non-)transitivity of randomized reducibilities. Tatsuaki Okamoto informed us of his conference-key sharing scheme discussed in his Ph.D thesis.

References

- [B93] Brands, S., "An efficient off-line electronic cash system based on the representation problem," CWI Technical Report CS-R9323 (Apr. 1993).
- [BeMi89] Bellare, M. and S. Micali, "Non-interactive oblivious transfer and applications," in *Advances in Cryptology – Crypto'89*, Lecture Notes in Computer Science 435, pp.547-557, *Springer-Verlag*, Berlin (1990).
- [Co85] Coppersmith, D. "Cheating at mental poker," *Advances in Cryptology – Crypto'85*, Lecture Notes in Computer Science 218, *Springer-Verlag*, Berlin, pp.104-107 (1986).
- [COS86] Coppersmith, D., A. M. Odlyzko, and R. Schroepel, "Discrete logarithms in $GF(p)$," *Algorithmica* 1, pp.1-15 (1986).

- [dB88] den Boer, B., "Diffie-Hellman is as strong as discrete log for certain primes," *Advances in Cryptology – Eurocrypt'88*, Lecture Notes in Computer Science 403, *Springer-Verlag*, Berlin, pp.530-539 (1990).
- [DH76] Diffie, W. and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, IT-22, No.6, pp.644-654, (Nov. 1976).
- [ElG85] ElGamal, T., "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inform. Theory*, IT-31, No.4, pp.469-472, (July 1985).
- [IR89] Impagliazzo, R. and Rudich, S., "Limits on the provable consequences of one-way permutations," *Proc. of 21st STOC*, pp.44-61 (1989).
- [Kob87a] Koblitz, N., "Elliptic curve cryptosystems," *Math. Comp.*, 48, pp.203-209 (1987).
- [Kob87b] Koblitz, N., "A Course in Number Theory and Cryptography," GTM 114, *Springer-Verlag* (1987).
- [Ma94] Maurer, U. M., "Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms," *Advances in Cryptology – Crypto'94*, Lecture Notes in Computer Science 839, *Springer-Verlag*, Berlin, pp.271-281 (1994).
- [Mil85] Miller, V., "Uses of elliptic curves in cryptography," *Advances in Cryptology – Crypto'85*, Lecture Notes in Computer Science 218, *Springer-Verlag*, Berlin, pp.417-426 (1986).
- [Miy91] Miyaji, A., "On ordinary elliptic curve cryptosystems," in *Advances in Cryptology – Asiacrypt'91*, Lecture Notes in Computer Science 739, *Springer-Verlag*.
- [MOV91] Menezes, A., T. Okamoto, and S. A. Vanstone, "Reducing elliptic logarithms to logarithms in a finite field," *Proc. of 23rd STOC*, pp.80-89 (1991).
- [Odl84] Odlyzko, A. M., "Discrete logarithms in finite fields and their cryptographic significance," *Advances in Cryptology – Eurocrypt'84*, Lecture Notes in Computer Science 209, *Springer-Verlag*, Berlin, pp.224-314 (1985).
- [Oka88] Okamoto, T., "Encryption and authentication schemes based on public-key systems" Ph.D. Thesis, *The University of Tokyo* (1988).
- [Oka94] Okamoto, T., Personal communication via email (1994).
- [Ra81] Rabin, M., "How to exchange secrets by oblivious transfer," *Tech. Memo TR-81*, Aiken Computation Laboratory, Harvard University, (1981).
- [Rib88] Ribenboim, P., "The Book of Prime Number Records," *Springer-Verlag* (1988).
- [Riv90] Rivest, R. L., "Cryptography," Chapter 13 of *Handbook of Theoretical Computer Science*, Vol. A, Algorithms and Complexity, edited by Jan van Leeuwen, *The MIT*, pp.717-755 (1990).
- [Rud91] Rudich, S., "The use of interaction in public cryptosystems," *Advances in Cryptology – Crypto'91*, Lecture Notes in Computer Science 576, *Springer-Verlag*, Berlin, pp.242-251 (1992).
- [SRA79] Shamir, A., R. L. Rivest, and L. Adleman, "Mental Poker," *MIT/LCS*, TM-125, (Feb. 1979).
- [Sch85] Schoof, R., "Elliptic curves over finite field and the computation of square roots mod p ," *Math. Comp.*, 44, pp.483-494 (1985).
- [TW87] Tompa, M. and H. Woll, "Random Self-Reducibility and Zero-Knowledge Interactive Proofs of Possession of Information," *Proc. of 28th FOCS*, pp.472-482 (1987).