J. Cryptology (1998) 11: 29–43

Journal of
CRYPTOLOGY

A Structural Comparison of the Computational Difficulty of Breaking Discrete Log Cryptosystems*

Kouichi Sakurai
Department of Computer Science and Communication Engineering,
Kyushu University, Hakozaki, Higashi-ku, Fukuoka, 812-81 Japan
sakurai@csce.kyushu-u.ac.jp

Hiroki Shizuya
ECIP & GSIS, Tohoku University,
Kawauchi, Aoba-ku, Sendai, 980-77 Japan
shizuya@ecip.tohoku.ac.jp

Communicated by Joan Feigenbaum

Received 18 January 1996 and revised 7 September 1996

Abstract. The complexity of breaking cryptosystems of which security is based on the discrete logarithm problem is explored. The cryptosystems mainly discussed are the Diffie–Hellman key exchange scheme (DH), the Bellare–Micali noninteractive oblivious transfer scheme (BM), the ElGamal public-key cryptosystem (EG), the Okamoto conference-key sharing scheme (CONF), and the Shamir 3-pass key-transmission scheme (3PASS). The obtained relation among these cryptosystems is that

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

where $\leq_m^{\mathrm{FP}}$ 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. One of such conditions suggest another advantage of the discrete logarithm associated with ordinary elliptic curves.

Key words. Cryptosystem, Computational number theory, Discrete logarithm, Elliptic curves, Key exchange, Public-key cryptography, Randomness, Security.

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

* A preliminary version of this paper was presented at Eurocrypto '95, Saint-Malo, 25 May 1995.

solve. A typical DLP is the case where $G = \mathbf{Z}_p^*$ with p prime. In 1976, Diffie and Hellman [6] first proposed a key exchange scheme that is thought to be 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., [4] and [15]). 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, great progress has been made by Maurer toward the equivalence of the DLP and breaking the Diffie–Hellman scheme [11], 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 in clarifying 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 key exchange scheme by DH, the Bellare–Micali noninteractive oblivious transfer scheme [1] by BM, the ElGamal public-key cryptosystem [7] by EG, the Okamoto conference-key sharing scheme [17] by CONF, and the Shamir 3-pass key-transmission scheme [25], [21] by 3PASS, respectively.

We first show a relationship among these cryptosystems that

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

where $\leq_m^{\mathrm{FP}}$ denotes the polynomial-time functionally many to one reducibility. We further give some condition in which these algorithms are equivalent with respect to 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 with respect to expected polynomial-time functionally Turing reduction, i.e., $\mathtt{3PASS} \equiv_T^{\mathrm{FEP}} \mathtt{CONF} \equiv_T^{\mathrm{FEP}} \mathtt{EG} \equiv_m^{\mathrm{FP}} \mathtt{BM} \equiv_m^{\mathrm{FP}} \mathtt{DH}$.
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 with respect to polynomial-time functionally many-to-one reduction, i.e., $\mathtt{3PASS} \equiv_m^{\mathrm{FP}} \mathtt{CONF} \equiv_m^{\mathrm{FP}} \mathtt{EG} \equiv_m^{\mathrm{FP}} \mathtt{BM} \equiv_m^{\mathrm{FP}} \mathtt{DH}$.

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

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

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_{3PASS} is not known to be in $\mathcal{P}$ or $\mathcal{BPP}$, we show that if L_{3PASS} is in $\mathcal{P}$, there is a probabilistic polynomial-time algorithm that reduces DH to 3PASS. Thus, if L_{3PASS} 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

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

Although L_{DH} is not known to be in $\mathcal{P}$ or $\mathcal{BPP}$ as observed in [2], we show that L_{DH} is random self-reducible in the sense of [27], 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 Functions*

Strength of complexity assumption is an important measure of the security of cryptographic protocols. Impagliazzo and Rudich [8], in fact, presented evidence that secure secret key agreement protocols require stronger complexity assumption than the existence of one-way permutations. Namely, if there exists a secure secret key agreement protocol which uses one-way permutations as a black box, then $\mathcal{P} \neq \mathcal{NP}$. While Impagliazzo and Rudich investigated the gap among several cryptographic primitives under the more general complexity assumption $\mathcal{P} \neq \mathcal{NP}$, this paper explores the relation among cryptographic primitives under a number-theoretic assumption on the hardness of computing the discrete logarithms.

A cryptographic protocol often requires a number of interactions. Rudich [22] 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 perform different functions and require a different number of interactions. Thus the results of this paper reveal relationships among computational complexity assumptions, round complexity, and functions of 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 these are based on the discrete logarithm problem (DLP). We restrict ourselves to the case where the underlying group is $\mathbf{Z}_p^*$ with p prime. It is reasonable to make this restriction because the cryptosystems involved all make this restriction. 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 simply write g^x rather than $g^x \bmod p$, etc.

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

Diffie–Hellman Key Exchange Scheme [6]

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 Noninteractive Oblivious Transfer Scheme [1]

Alice and Bob agree on p and the base $g \in \mathbf{Z}_p^*$ and some $C \in \mathbf{Z}_p^*$. Bob randomly picks $i \in \{0, 1\}$ and $x_i \in \mathbf{Z}_{p-1}$, 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 Public-Key Cryptosystem [7]

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 Conference-Key Sharing Scheme [17]

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 will 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 cannot send a message directly. This property has an advantage over the Diffie–Hellman key exchange scheme in the case of a conference-key sharing scheme for multiple users [17].

Shamir 3*-Pass Message Transmission Scheme* [25]

This is also called the Massey–Omura cryptosystem (see, e.g., [10]), and originally proposed as a tool for mental poker by Shamir *et al*. [25], [21]. 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 2.1. Shamir's 3-pass key transmission scheme is useful not only for secret message transferring but also for an oblivious transfer [19]. 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; and
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., [3]). Shamir *et al*. [25], [19] 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 is a more functional protocol 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 $\beta = \alpha_e^b$, 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 $\Sigma = \{0, 1\}$.

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.

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.

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).

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 $C_2 = mg^{xr}$, $y = g^x$, $C_1 = g^r$ if such an m exists.

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 a C exists.

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 return a correct answer if there is a solution to the query. However, there is no mention 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 3.3).

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

Definition 2.3. A function f is *expected* polynomial-time functionally Turing reducible to a function g (in symbols $f \leq_T^{\mathrm{FEP}} g$) if an expected polynomial-time oracle Turing machine with access to values of g can compute f. (*Note*. We say that a machine M is *expected polynomial-time* if there exists an $e > 0$ such that, for all $x \in \{0,1\}^*$, the expectation, taken over the infinite bit sequences r, of $(t_M(x,r))^e$ is bounded above by $|x|$ (i.e., $E((t_M(x,r))^e) \leq |x|$).)

Definition 2.4. A function f is polynomial-time functionally many-to-one reducible to a function g (in symbols $f \leq_m^{\mathrm{FP}} g$) 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 3.1. $\mathtt{3PASS} \leq_m^{\mathrm{FP}} \mathtt{CONF} \leq_m^{\mathrm{FP}} \mathtt{EG} \equiv_m^{\mathrm{FP}} \mathtt{BM} \equiv_m^{\mathrm{FP}} \mathtt{DH} \leq_m^{\mathrm{FP}} \mathtt{DLP}$.

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

$\mathtt{3PASS} \leq_m^{\mathrm{FP}} \mathtt{CONF}$:
Let $(A, B, C, p) = (s^a, s^b, s^{ab}, p)$ be an instance of 3PASS.

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

$\mathtt{CONF} \leq_m^{\mathrm{FP}} \mathtt{EG}$:
Let $(A, B, g, p) = (g^a, g^{ab}, g, p)$ be an instance of CONF.

$$\mathtt{CONF}(A, B, g, p) = \frac{1}{\mathtt{EG}(g, 1, g^{ab}, g^a, p)}.$$

$\mathsf{EG} \leq_m^{\mathrm{FP}} \mathsf{DH}$:
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 $m = C_2/g^{xr}$.

$\mathsf{DH} \leq_m^{\mathrm{FP}} \mathsf{EG}$ [18]:
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 $(A, 1, B, g, p)$.

$\mathsf{BM} \equiv_m^{\mathrm{FP}} \mathsf{DH}$ [16]:
It is not hard to see that $\mathsf{BM} \leq_m^{\mathrm{FP}} \mathsf{DH}$ 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

$$((\alpha_0, \alpha_1), (r_0, r_1), C, (\beta_0, \beta_1), g, p) = ((A, A), (0, 0), B^2, (B, B), g, p).$$

Since we set $r_0 = r_1 = 0$, the oracle $\mathsf{BM}((A, A), (0, 0), B^2, (B, B), g, p)$ returns $s_i = r_i \oplus \gamma_i = 0 \oplus g^{ab} = g^{ab}$, no matter which value i takes.
This completes the proof. □

Remark 3.2. The recent published textbook by Stinson [26] gives the theorem on "$\mathsf{EG} \equiv_m^{\mathrm{FP}} \mathsf{DH}$" with a proof.

We do not know if $\mathsf{DH} \leq_m^{\mathrm{FP}} \mathsf{3PASS}$. However, if we consider more strong 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:

$\mathsf{3PASS}^\star(A, B, C, p)$ 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 3.3. $\mathsf{DH} \leq_T^{\mathrm{FEP}} \mathsf{3PASS}^\star$.

Proof. Let $(A, B, g, p) = (g^a, g^b, g, p)$ be an instance of DH. We transform it into an instance of $\mathsf{3PASS}^\star$ by

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

where u and v are randomly picked from $\mathbf{Z}_{p-1}$. We show that if $\mathsf{3PASS}^\star(g^{a+u}, g^{b+v}, g, p)$ returns s other than $\perp$, then $s = g^{(a+u)(b+v)}$. Once this s is obtained, the output of $\mathsf{DH}(A, B, g, p)$ is computed as $g^{ab} = g^{(a+u)(b+v)}/(A^v B^u g^{uv})$.
If the oracle returns s, it satisfies that for some $\alpha, \beta \in \mathbf{Z}_{p-1}^*$,

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

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

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

where ord(g) designates the order of g, and r is an element in $\mathbf{Z}_{\mathrm{ord}(g)}$ such that $s = g^r \bmod p$. Then, we have that $(a+u)(b+v) = r^2\alpha\beta = r(r\alpha\beta) = r$. Thus, $s = g^r = g^{(a+u)(b+v)}$. Note that $r\alpha\beta = 1 \bmod \mathrm{ord}(g)$ implies that $r, \alpha, \beta \in \mathbf{Z}^*_{\mathrm{ord}(g)}$, and both $a+u$ and $b+v$ are in $\mathbf{Z}^*_{\mathrm{ord}(g)}$.

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*.

```
% Algorithm 1
% DHto3PASS
input A, B, g, p
s := ⊥
while (s = ⊥) do
      pick u, v ∈ Z_{p-1} at random
      A' := Ag^u; B' := Bg^v
      s := 3PASS*(A', B', g, p)
end while
C := s/(A^v B^u g^{uv})
output C
end
```

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 the probability that both $a+u$ and $b+v$ are in $\mathbf{Z}^*_{\mathrm{ord}(g)}$, which is greater than or equal to 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 $\varphi(n) \geq \ln(2)\cdot n/\ln(2n)$ for a positive integer n [20], 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. □

Remark 3.4. 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 DH$^\star \leq_T^{\mathrm{FEP}}$ 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 := ⊥; C := ⊥; i := 1
T := q(|p|) % some polynomial in |p|
while ([s = ⊥] ∧ [i ≤ T]) do
      pick u, v ∈ Z_{p-1} at random
      A' := Ag^u; B' := Bg^v
```

$s := \mathsf{3PASS}^\star(A', B', g, p)$
$i := i + 1$
end while
if $s \neq \perp$, then $C := s/(A^v B^u g^{uv})$
output C
end

We do not know if $\mathsf{DH} \equiv \mathsf{DH}^\star$ or $\mathsf{3PASS} \equiv \mathsf{3PASS}^\star$ 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 3.5. *If the complete factorization of $p-1$ with p prime is given and the base g is a generator of $\mathbf{Z}_p^*$,*

$$\mathsf{DH} \leq_T^{\mathrm{FEP}} \mathsf{3PASS}.$$

Proof. In the proof of Theorem 3.3, we have shown that DH reduces to $\mathsf{3PASS}^\star$, where $\mathsf{3PASS}^\star$ 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.

% 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
  $X := Ag^u$; $Y := Bg^v$
  $$d := \left[\bigwedge_{i=0}^{k} X^{(p-1)/p_i} \neq 1\right] \wedge \left[\bigwedge_{i=0}^{k} Y^{(p-1)/p_i} \neq 1\right]$$
end while
$s := \mathsf{3PASS}(X, Y, g, p)$
$C := s/(A^v B^u g^{uv})$
output C
end

Here, $d = \texttt{true}$ 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 $|p|$. □

den Boer [5] showed that the Diffie–Hellman problem is as strong as the discrete logarithms for certain primes. It is remarkable that Maurer [11] made this result stronger to cover generic cyclic groups. Let $\varphi(N)$ be the order of the group $\mathbf{Z}_N^*$.

Theorem 3.6 [5] (see also [11]). *If $\varphi(p-1)$ is smooth, i.e., it consists of small prime factors with respect to a fixed polynomial in $q(|p|)$, then* $\texttt{DLP} \leq_T^{\text{FEP}} \texttt{DH}$.

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

Corollary 3.7. *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*

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

3.3. *The Case of Elliptic Discrete Logarithms*

Next we consider these cryptosystems based on the elliptic-curve discrete logarithm problem [9], [13], 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 $|p|$ [24]. 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 nontrivial 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 $\mathbf{Z}_p$ [12]. However, no such reduction algorithm is known to exist for elliptic-curve groups with prime order [14].

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 3.1 also hold for the EDLP-based systems. Let $\texttt{DH}_E$ (resp. $\texttt{BM}_E$, $\texttt{EG}_E$, $\texttt{CONF}_E$, $\texttt{3PASS}_E$)

designate the EDLP-based DH (resp. BM, EG, CONF, 3PASS) problem. We have the following theorem.

Theorem 3.8. *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*

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

Proof. As Theorem 3.1, it is easily seen that $\mathsf{3PASS}_E \leq_m^{\mathrm{FP}} \mathsf{CONF}_E \leq_m^{\mathrm{FP}} \mathsf{EG}_E \equiv_m^{\mathrm{FP}} \mathsf{BM}_E \equiv_m^{\mathrm{FP}} \mathsf{DH}_E$. Thus, it suffices to show that $\mathsf{DH}_E \leq_m^{FP} \mathsf{3PASS}_E$. 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 $\mathsf{3PASS}_E$ always returns the correct answer to a query (A, B, P, E, p). Hence, $\mathsf{DH}_E \leq_m^{FP} \mathsf{3PASS}_E$. □

There is little known research on the distribution of the prime-order elliptic curves over all elliptic curves. A construction of the prime-order elliptic curves is also studied in [14], 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 [12]. Our theorem above is based on another interesting property of ordinary prime-order elliptic curves over $\mathbf{Z}_p$ that any nonzero 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

$$L_{\mathsf{Q}} = \{(x, y) |\ \mathsf{Q}(x) = y\},$$

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. This language is also known as the graph of Q when Q is regarded as a function. Clearly, these languages are in $\mathcal{NP} \cap$ 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 [2]. Thus, there may be a reduction sequence among these languages which is different from the reductions given in Theorem 3.1, 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 3.9. *If* L_{3PASS} *is in* $\mathcal{P}$, $\mathsf{DH} \leq_T^{\mathrm{FEP}} \mathsf{3PASS}$.

Proof. We exploit Algorithm 1 in the proof of Theorem 3.3 which reduces DH to $\mathsf{3PASS}^*$. Note that $\mathsf{3PASS}^*$ returns $\perp$ when there is no solution to the instance, whereas 3PASS does not. However, one can now check in deterministic polynomial time that the

value returned from 3PASS is correct because by the assumption, $L_{\mathtt{3PASS}}$ is in $\mathcal{P}$. As a slightly modified version of Algorithm 1, the Algorithm 4 shown below reduces DH to 3PASS in expected polynomial time.

```
% Algorithm 4
input A, B, g, p
d := false
while (d = false) do
        pick u, v ∈ Z_{p-1} at random
        A' := Ag^u; B' := Bg^v
        s := 3PASS(A', B', g, p)
        d := [((A', B', g, p), s) ∈ L_3PASS]
end while
C := s/(A^v B^u g^{uv})
output C
end
```

This completes the proof. □

Also we obtain

Corollary 3.10. *If $L_{\mathtt{3PASS}}$ is in $\mathcal{P}$, then* $\mathtt{3PASS} \equiv_T^{\mathrm{FEP}} \mathtt{CONF} \equiv_T^{\mathrm{FEP}} \mathtt{EG} \equiv_m^{\mathrm{FP}} \mathtt{BM} \equiv_m^{\mathrm{FP}} \mathtt{DH}$.

The corollary above gives a characterization of the complexity of $L_{\mathtt{3PASS}}$, i.e., $L_{\mathtt{3PASS}}$ in not in $\mathcal{P}$ if one of these equivalence relationships fails to hold. Note that, at present, assuming $L_{\mathtt{3PASS}} \in \mathcal{P}$ is not known to be related to the assumptions in Maurer's work [11] on the equivalence of DLP and DH.

The following theorem implies that $L_{\mathtt{DH}}$ has a perfect zero-knowledge interactive proof.

Theorem 3.11. *The language $L_{\mathtt{DH}}$ is random self-reducible in the sense of* [27].

Proof. For an instance $((A, B, g, p), C)$, let $A' = Ag^r$, $B' = Bg^s$, and $C' = CA^sB^rg^{rs}$ 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 if $((A', B'g, p), C')$ is in $L_{\mathtt{DH}}$, so is $((A, B, g, p), C)$. This implies $L_{\mathtt{DH}}$ is random self-reducible. □

3.5. *Single-Use Versus Multiple-Use in Cryptosystems*

Consider the situation that we use the Shamir 3-pass scheme for transferring the same message s many times. In such a case, an adversary can get more information than single transfer. 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 $A_1, B_1, C_1, \ldots, A_k, B_k, C_k, \in \mathbf{Z}_p^*$, outputs s such that $A_j = s^{a_j}$, $B_j = s^{b_j}$, $C_j = s^{a_j b_j}$, and $a_j, b_j \in \mathbf{Z}_{p-1}^*$ $(j = 1, \ldots, k)$ if such an s exists.

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

Theorem 3.12. *For any fixed $k \geq 1$,* 1-3PASS (= 3PASS) $\leq_m^{\mathrm{FP}}$ k-3PASS.

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

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

Then, $((A_1, B_1, C_1, p), \ldots, (A_k, B_k, C_k, p))$ is an instance of k-3PASS. 1-3PASS (A, B, C, p) is computed as

$$\text{1-3PASS}(A, B, C, p) = k\text{-3PASS}((A_1, B_1, C_1, p), \ldots, (A_k, B_k, C_k, p)). \qquad \square$$

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 $C_{11}, C_{21}, \ldots, C_{1k}, C_{2k}, y, g \in \mathbf{Z}_p^*$, outputs $m \in \mathbf{Z}_p^*$ such that $C_{2j} = mg^{xr_j}$, $y = g^x$, $C_{1j} = g^{r_j}$ $(j = 1, \ldots, k)$ if such an m exists.

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

Theorem 3.13. *For any fixed $k \geq 1$,* 1-EG(= EG) $\leq_m^{\mathrm{FP}}$ k-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 $u_1, \ldots, u_k \in \mathbf{Z}_{p-1}$ at random. Then, put

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

Since $C_{1i} = g^{r+u_i}$ and $C_{2i} = mg^{x(r+u_i)}$, we now have an instance of k-EG as

$$((C_{11}, C_{21}, y, g, p), \ldots, (C_{1k}, C_{2k}, y, g, p)).$$

Then

$$\text{1-EG}(C_1, C_2, y, g, p) = k\text{-EG}((C_{11}, C_{21}, y, g, p), \ldots, (C_{1k}, C_{2k}, y, g, p)). \qquad \square$$

Okamoto [17] observed such a property in his scheme.

Theorem 3.14 [17]. *For any fixed $k \geq 1$,* 1-CONF(= CONF) $\leq_m^{\mathrm{FP}}$ k-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 3.1). Specifically, we have shown that these problems are equivalent under the stronger function model (Theorem 3.3), 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 3.5), or if it is the elliptic-curve discrete logarithm problem associated with an ordinary elliptic curve defined over $\mathbf{Z}_p$ (Theorem 3.8). 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 $L_{\mathtt{3PASS}}$ reduce to $L_{\mathtt{DH}}$ with respect to $\leq_T^p$-reducibility?
- Does $L_{\mathtt{DH}}$ reduce to $L_{\mathtt{3PASS}}$ with respect to $\leq_T^p$-reducibility?
- Does $L_{\mathtt{3PASS}}$ 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. Motoji Ohmori helped us improve the proof for the equivalence of BM and DH. Tatsuaki Okamoto informed us of his conference-key sharing scheme discussed in his Ph.D. thesis. The anonymous referees gave us many helpful comments on this paper.

References

[1] Bellare, M. and Micali, S., Non-interactive oblivious transfer and applications, In: *Advances in Cryptology—Crypto* '89, Lecture Notes in Computer Science, vol. 435, Springer-Verlag, Berlin, pp. 547–557, 1990.

[2] Brands, S., An efficient off-line electronic cash system based on the representation problem, CWI Technical Report CS-R9323, April 1993.

[3] Coppersmith, D., Cheating at mental poker, *Advances in Cryptology—Crypto* '85, Lecture Notes in Computer Science, vol. 218, Springer-Verlag, Berlin, pp. 104–107, 1986.

[4] Coppersmith, D., Odlyzko, A. M., and Schroeppel, R., Discrete logarithms in $GF(p)$, *Algorithmica*, vol. 1, pp. 1–15, 1986.

[5] den Boer, B., Diffie–Hellman is as strong as discrete log for certain primes, *Advances in Cryptology—*

Eurocrypt '88, Lecture Notes in Computer Science, vol. 403, Springer-Verlag, Berlin, pp. 530–539, 1990.

[6] Diffie, W. and Hellman, M. E., New directions in cryptography, *IEEE Trans. Inform. Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.

[7] ElGamal, T., A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Trans. Inform. Theory*, vol. IT-31, no. 4, pp. 469–472, 1985.

[8] Impagliazzo, R. and Rudich, S., Limits on the provable consequences of one-way permutations, *Proc.* 21*st STOC*, pp. 44–61, 1989.

[9] Koblitz, N., Elliptic curve cryptosystems, *Math. Comput.*, vol. 48, pp. 203–209, 1987.

[10] Koblitz, N., *A Course in Number Theory and Cryptography*, Graduate Texts in Mathematics, vol. 114, Springer-Verlag, New York, 1987.

[11] 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, vol. 839, Springer-Verlag, Berlin, pp. 271–281, 1994.

[12] Menezes, A., Okamoto, T., and Vanstone, S. A., Reducing elliptic logarithms to logarithms in a finite field, *Proc.* 23*rd STOC*, pp. 80–89, 1991.

[13] Miller, V., Uses of elliptic curves in cryptography, *Advances in Cryptology—Crypto* '85, Lecture Notes in Computer Science, vol. 218, Springer-Verlag, Berlin, pp. 417–426, 1986.

[14] Miyaji, A., On ordinary elliptic curve cryptosystems, in *Advances in Cryptology—Asiacrypt* '91, Lecture Notes in Computer Science, vol. 739, Springer-Verlag, Berlin, pp. 460–469, 1993.

[15] Odlyzko, A. M., Discrete logarithms in finite fields and their cryptographic significance, *Advances in Cryptology—Eurocrypt* '84, Lecture Notes in Computer Science, vol. 209, Springer-Verlag, Berlin, pp. 224–314, 1985.

[16] Ohmori, M., Personal communication via email, 1995.

[17] Okamoto, T., Encryption and authentication schemes based on public-key systems Ph.D. Thesis, The University of Tokyo, 1988.

[18] Okamoto, T., Personal communication via email, 1994.

[19] Rabin, M., How to exchange secrets by oblivious transfer, Technical Memo TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[20] Ribenboim, P., *The Book of Prime Number Records*, Springer-Verlag, New York, 1988.

[21] Rivest, R. L., Cryptography, Chapter 13 of *Handbook of Theoretical Computer Science*, Vol. A, *Algorithms and Complexity* (Jan van Leeuwen, ed.) MIT Press, Cambridge, MA, pp. 717–755, 1990.

[22] Rudich, S., The use of interaction in public cryptosystems, *Advances in Cryptology—Crypto* '91, Lecture Notes in Computer Science, vol. 576, Springer-Verlag, Berlin, pp. 242–251, 1992.

[23] Sakurai, S., and Shizuya, H., Relationships among the computational powers of breaking discrete log cryptosystems, *Advances in Cryptology—Eurocrypt* '95, Lecture Notes in Computer Science, vol. 921, Springer-Verlag, Berlin, pp. 341–355, 1995.

[24] Schoof, R., Elliptic curves over finite field and the computation of square roots mod p, *Math. Comput.*, vol. 44, pp. 483–494, 1985.

[25] Shamir, A., Rivest, R. L., and Adleman, L., Mental Poker, MIT/LCS, TM-125, Feb. 1979.

[26] Stinson, R. D., *Cryptography: Theory and Practice*, CRC Press, Boca Raton, FL, pp. 267–268, 1995.

[27] Tompa, M. and Woll, H., Random self-reducibility and zero-knowledge interactive proofs of possession of information, *Proc.* 28*th FOCS*, pp. 472–482, 1987.