NP-Complete Sets for Computing Discrete Logarithms and Integer Factorization

Shingo HASEGAWA*, Shuji ISobe, and Hiroki SHIZUYA

Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan

Received July 14, 2011; final version accepted March 16, 2013

We explore an NP-complete set such that the problem of breaking a cryptographic scheme reduces to the complete set, where the reduction can be given in a straightforward form like the reduction from the graph isomorphism to the subgraph isomorphism. We construct such NP-complete sets \( \Pi_{DL} \) and \( \Pi_{IF} \) for the discrete logarithm problem modulo a prime and the integer factoring problem, respectively. We also show that the decision version of Diffie-Hellman problem reduces directly to \( \Pi_{DL} \) with respect to the polynomial-time many-one reducibility. These are the first complete sets that have direct reductions from significant cryptographic primitives.

KEYWORDS: computational complexity, cryptography

1. Introduction

Difficulty of solving a problem can be characterized by the terminology of computational complexity theory. To see this in the context of cryptography, define the following set:

\[
\pi_{DL} = \{(p, g, y, k) \mid \exists x [(y \equiv g^x \pmod{p}) \land (x \leq k)]\},
\]

where \( p \) is a prime. This set is associated with a typical cryptographic primitive called the discrete logarithm problem: given \((p, g, y)\), find \( x \) such that \( y \equiv g^x \pmod{p} \) if such an \( x \) exists. Quantum computer can easily solve it in theory [17]. However, in a practical sense, we have no efficient algorithm for this problem, and therefore it is often used to construct cryptographic schemes to ensure the difficulty of breaking the systems. Assume now that there is a polynomial-time algorithm \( M \) that on input \((p, g, y, k)\), outputs 1 if \((p, g, y, k) \in \pi_{DL}\), or 0 otherwise. Then the discrete logarithms modulo \( p \) can also be computed in polynomial time by calling \( M \) as a subroutine polynomially many times. Therefore the set \( \pi_{DL} \) is thought to express the difficulty of the discrete logarithm problem. This characterization originated in [2], and has been applied to other discrete logarithm problems [13, 14, 19]. Similarly, the following set corresponds to the integer factoring problem.

\[
\pi_{IF} = \{(n, k) \mid \exists d [(1 < \gcd(n, d) < n) \land (d \leq k)]\},
\]

where \( n \) is an integer to factor. It is clearly seen that \( \pi_{DL} \) and \( \pi_{IF} \) are sets in \( \text{NP} \), and their complements are also in \( \text{NP} \). Hence both \( \pi_{DL} \) and \( \pi_{IF} \) are in \( \text{NP} \cap \text{co-NP} \). If either \( \pi_{DL} \) or \( \pi_{IF} \) is complete for \( \text{NP} \), then the polynomial-time hierarchy collapses to \( \text{NP} \), which is believed to be an unlikely event. In this way, the difficulty of computing discrete logarithms modulo a prime and factoring integers have been characterized by some sets and their complexity classes.

Now we recall that \( \pi_{DL} \) and \( \pi_{IF} \) are in \( \text{NP} \). Thus \( \pi_{DL} \) and \( \pi_{IF} \) reduce to some NP-complete set. We then consider the following question: Which NP-complete set is appropriate to relate with \( \pi_{DL} \) or \( \pi_{IF} \)?

The question above could sound nonsense because any set in \( \text{NP} \) reduces to any NP-complete set, say SAT (i.e., the set of satisfiable Boolean formulas in conjunctive normal form). That is, for a set \( A \) in \( \text{NP} \) and any string \( x \), one can obtain a Boolean formula \( \varphi_x \) in time polynomial in the length of \( x \), such that \( x \in A \) if and only if \( \varphi_x \in \text{SAT} \). However, this generic reduction is very abstract and complicated, and hence we cannot capture the relationship between cryptographic primitives and NP-complete sets. Below we describe a relationship between the graph isomorphism (GI) and the subgraph isomorphism (SubGI) as a comprehensive example illustrating a simple reduction.

Let \( G_1 \) and \( G_2 \) be two graphs. For simplicity, we restrict ourselves to the graphs that are vertex-labeled, connected and simple. Further, without loss of generality, we assume that the set of vertices and the number of edges of \( G_1 \) are the same as those of \( G_2 \), namely \( G_1 = (V, E_1) \) and \( G_2 = (V, E_2) \) with \( \#E_1 = \#E_2 \). Then we say that \( G_1 \) and \( G_2 \) are isomorphic, if there exists a permutation \( \sigma \) on \( V \) such that for every pair of vertices \( i, j \in V, (i, j) \in E_1 \Leftrightarrow (\sigma(i), \sigma(j)) \in E_2 \). We write \( G_1 \cong G_2 \) if \( G_1 \) is isomorphic to \( G_2 \). We then define the set of all pairs of isomorphic graphs as \( \text{GI} = \{(G_1, G_2) \mid G_1 \cong G_2 \} \). The subgraph isomorphism can be regarded as a natural extension of the notion of GI.

*Corresponding author. E-mail: hasegawa@cite.tohoku.ac.jp
That is, for two graphs $G_1$ and $G_2$, where the set of vertices and the number of edges are not necessarily common, we say that $(G_1, G_2)$ is in SubGI if and only if there exists a subgraph $H$ in $G_2$ such that $G_1 \cong H$. SubGI is known to be NP-complete set, while GI is in $\text{NP} \cap \text{co-AM}$ [9]. By the definition above, we can see that SubGI contains GI as a subset. Thus GI directly and naturally reduces to SubGI, namely for any instance $(G_1, G_2)$ for GI, $(G_1, G_2)$ itself is directly queried to the SubGI oracle.

We are interested in an NP-complete set such that it contains a subset closely related to the discrete logarithm problem (resp. the integer factoring problem), and $\pi_{\text{DL}}$ (resp. $\pi_{\text{IF}}$) reduces to the NP-complete set in a direct manner, like the reduction from GI to SubGI. In other words, we intend to construct an NP-complete set which contains $\pi_{\text{DL}}$ (resp. $\pi_{\text{IF}}$) as a subset, or alternatively an NP-complete set which can be viewed as a superset of $\pi_{\text{DL}}$ (resp. $\pi_{\text{IF}}$) in a sense that there exists an immediate witness preserving reduction just like an insertion map: any instance $(p, g, y, k)$ for $\pi_{\text{DL}}$ (resp. $(n, k)$ for $\pi_{\text{IF}}$) is embedded in the query to the oracle without any change. Again, the question is, what is such an NP-complete set? This question was posed in [7], but the answer is not known so far.

In this paper, we define two sets $\Pi_{\text{DL}}$ and $\Pi_{\text{IF}}$, and prove that (a) $\Pi_{\text{DL}}$ and $\Pi_{\text{IF}}$ are complete for NP, and (b) $\pi_{\text{DL}} \leq_{m}^{P} \Pi_{\text{DL}}$ and $\pi_{\text{IF}} \leq_{m}^{P} \Pi_{\text{IF}}$ via an immediate reduction which is almost an insertion map, where we denote by $\leq_{m}^{P}$ the polynomial-time many-one reducibility. This implies that the discrete logarithm problem (resp. the integer factoring problem) reduces to an NP-complete set $\Pi_{\text{DL}}$ (resp. $\Pi_{\text{IF}}$) with respect to the polynomial-time Turing reducibility. Further, we also show that the decision Diffie-Hellman problem reduces to $\Pi_{\text{DL}}$ with respect to the $\leq_{m}^{P}$-reducibility. $\Pi_{\text{DL}}$ and $\Pi_{\text{IF}}$ defined in this paper are the first NP-complete sets that have direct reductions from significant cryptographic primitives.

Note that in [7] they found NPMV-complete functions that directly compute discrete logarithms and integer factorization, where NPMV denotes the class of all partial, multivalued functions computed nondeterministically in polynomial time [15]. Therefore our results should be regarded as a set (or a language) counterpart of the results presented in [7].

After the section for preliminaries, our main results are presented in §3. To further understand the properties of $\Pi_{\text{DL}}$ and $\Pi_{\text{IF}}$, we restrict the parameters of these sets and evaluate their complexity in §4.

2. Preliminaries

We start with the definitions of notions and notations used in this paper.

We denote by $R[X_1, \ldots, X_k]$ the ring of polynomials in $k$ indeterminates $(X_1, \ldots, X_k)$ over a ring $R$. By $\mathbb{Z}/n\mathbb{Z}$, we denote the residue class ring modulo $n$, and by $(\mathbb{Z}/n\mathbb{Z})^k$ its group of units. If $p$ is in $\text{Primes}$, the set of prime numbers, we write $\mathbb{F}_p$ instead of $\mathbb{Z}/p\mathbb{Z}$, which is the finite field of $p$ elements.

Let $S$ be a partially ordered set. For $k$-tuples $a = (a_1, \ldots, a_k) \in S^k$ and $b = (b_1, \ldots, b_k) \in S^k$, we write $a \leq b$ if and only if $\forall i(1 \leq i \leq k) a_i \leq b_i$.

We use $\text{NP}$ and $\text{P}$ as the classes of sets in the usual manner. By $\leq_{m}^{P}$ and $\leq_{f}^{P}$, we denote the polynomial-time many-one reducibility and the polynomial-time Turing reducibility, respectively. For sets $A, B$, we have that $A \leq_{m}^{P} B \Rightarrow A \leq_{f}^{P} B$, but the converse does not necessarily hold. A set $A$ is complete for NP with respect to the $\leq_{m}^{P}$-reducibility if and only if $A \in \text{NP}$ and $\forall B \in \text{NP} B \leq_{m}^{P} A$. SAT is a typical NP-complete set that consists of all the satisfiable Boolean formulas in conjunctive normal form. A set $A \in \text{NP}$ is NP-complete if SAT $\leq_{m}^{P} A$ because the $\leq_{m}^{P}$-reducibility is transitive.

To define function classes, we introduce the notion of a transducer [5, 15]. A transducer $T$ is a nondeterministic Turing machine. $T$ computes a value $y$ on input $x$ if there is an accepting computation of $T$ on $x$ for which $y$ is the final content of $T$’s output tape. Such transducers compute partial, multivalued functions $f$, and we write $f(x) \mapsto y$. NPMV is the set of all partial, multivalued functions computed by nondeterministic polynomial time-bounded transducers, and $\text{PF}$ is the set of all partial functions computed by deterministic polynomial time-bounded transducers. For a function $f(X)$ with a variable $X$, we write $f(X \leftarrow x)$ to express a value of $f$ at $X = x$.

Let $\text{dom}(\text{DL}) = \{(p, g, y)| (p \in \text{Primes}) \land (g \in \mathbb{F}_p^*) \land (y \in \langle g \rangle)\}$, where $\langle g \rangle$ denotes the cyclic subgroup of $\mathbb{F}_p^*$ generated by $g$. Then the function $\text{DL}$ is defined as follows: For each $(p, g, y) \in \text{dom}(\text{DL})$, $\text{DL}(p, g, y) \mapsto x \in \mathbb{Z}/(p - 1)\mathbb{Z}$ such that $y \equiv g^x \pmod{p}$.

Let $\text{dom}(\text{IF})$ is the set of all composites. Then function $\text{IF}$ is defined as follows: For each $n \in \text{dom}(\text{IF})$, $\text{IF}(n) \mapsto d \in \mathbb{N}$ such that $1 < \gcd(n, d) < n$.

Let $A$ be a set in $\text{NP}$ with the representation $A = \{x | \exists u[(x, u) \in B]\}$, where $B \in \text{P}$. We say $A$ has self-computable solutions [9] if there exists a function $f \in \text{PF}^A$ such that for all $x \in A, (x, f(x)) \in B$, where $f \in \text{PF}^A$ means that $f$ reduces to the set $A$ with respect to the $\leq_{m}^{P}$-reducibility. Intuitively, if a set $A \in \text{NP}$ has self-computable solutions, one can compute a witness $w$ for $x \in A$, using $A$ itself as the oracle set.

The sets $\pi_{\text{DL}}$ and $\pi_{\text{IF}}$ defined in §1 have self-computable solutions, where the corresponding functions are $\text{DL}$ and $\text{IF}$, respectively. In other words, we have that $\text{DL} \in \text{PF}^{|\text{IF}\text{II}}$ and $\text{IF} \in \text{PF}^{|\text{IF}\text{II}}$.
3. Main Results

We are exploring NP-complete sets $L_1$ and $L_2$ such that $\pi_{DL} \leq^p_m L_1$ and $\pi_{IF} \leq^p_m L_2$, where the reductions are as straightforward as the one from GI to SubGI. For the discrete logarithm problem, we introduce $\Pi_{DL}$, prove that it is complete for NP, and show that $\pi_{DL}$ directly reduces to $\Pi_{DL}$. We proceed in the same way for the integer factoring problem.

3.1 Discrete logarithm problem

Definition 3.1. Let $p \in \text{Primes} \setminus \{2\}$, $g = (g_1, \ldots, g_6) \in (\mathbb{F}_p^*)^6$, $F = (F_1, \ldots, F_t) \in (\mathbb{Z}/(p-1)\mathbb{Z}[X_1, \ldots, X_t])^t$, $y = (y_1, \ldots, y_t) \in (\mathbb{F}_p^*)^t$, and $d = (d_1, \ldots, d_t) \in (\mathbb{Z}/(p-1)\mathbb{Z})^t$. We define the set $\Pi_{DL}$ as

$$\Pi_{DL} = \{ (p, g, F, y, d) \mid \exists x = (x_1, \ldots, x_t) \in (\mathbb{Z}/(p-1)\mathbb{Z})^t \text{ such that } \left( \bigwedge_{i=1}^t y_i \equiv \frac{F_i}{g} (X_1^{x_1} \cdots X_t^{x_t}) \pmod p \right) \land (x \leq d) \}.$$

From the definition above, it is easily seen that $\Pi_{DL}$ is in NP. We show that the set is NP-complete.

Theorem 3.2. $\Pi_{DL}$ is complete for NP with respect to the $\leq^p_m$-reducibility.

Proof. We show that $\text{SAT} \leq^p_m \Pi_{DL}$. Let $\varphi(x_1, \ldots, x_t) = \bigwedge_{i=1}^t C_i$ be a Boolean formula in conjunctive normal form, where $C_i$’s are clauses. We transform each $C_i$ into $c_i(x_1, \ldots, x_t) \in \mathbb{F}_2[X_1, \ldots, X_t]$ by the arithmetization [10, 16]. Assume without loss of generality that $C_i = \bigvee_{j=1}^{m_i} L_{ij}$, where $L_{ij}$ and $m_i$ denote the literals and the number of literals, respectively within the clause $C_i$. The arithmetization proceeds as follows. For each clause, we map each $c_i(x_1, \ldots, x_t) \in \mathbb{F}_2[X_1, \ldots, X_t]$ by the arithmetization [10, 16]. Assume without loss of generality that $C_i = \bigvee_{j=1}^{m_i} L_{ij}$, where $L_{ij}$ and $m_i$ denote the literals and the number of literals, respectively within the clause $C_i$. The arithmetization proceeds as follows. For each clause, we map each $c_i(x_1, \ldots, x_t) \in \mathbb{F}_2[X_1, \ldots, X_t]$ by the following form:

$$c_i(x_1, \ldots, x_t) = \left( x_{i_1} \cdots x_{i_L} \right) \left( c_{i_1} \cdots c_{i_L} \right) \left( x_{i_{L+1}} \cdots x_{i_{m_i}} \right) \left( c_{i_{L+1}} \cdots c_{i_{m_i}} \right).$$

Since 2 is a primitive root mod 3, the order of 2 in $\mathbb{F}_3^*$ is $\#\mathbb{F}_3^* = 2$. Thus $2^e \equiv 2 \pmod 3$ if and only if $e \equiv 1 \pmod 2$. If there exists $x \in \{0, 1\}^e$ such that for all $i (1 \leq i \leq \ell)$, $c_i[x_1 \leftarrow x_1, \ldots, x_k \leftarrow x_k] = 1$, then we have that for all $i (1 \leq i \leq \ell)$, $2^e \equiv 2 \pmod 3$. Therefore $\varphi$ is in SAT if and only if $(p, g, F, y, d)$ defined above is in $\Pi_{DL}$.

The statement of the next corollary implies that $\text{DL} \in \mathsf{P}^{\Pi_{DL}}$. The reduction given in the proof shows that $\Pi_{DL}$ is the NP-complete set that we are looking for.

Corollary 3.3. $\pi_{DL} \leq^p_m \Pi_{DL}$.

Proof. $(p, g, y, k) \in \pi_{DL} \iff (p, g, X, y, k) \in \Pi_{DL}$. \hfill \Box

A typical cryptographic schemes based on the difficulty of computing discrete logarithms modulo a prime is the Diffie-Hellman key exchange scheme. Intuitively, the scheme is designed for two parties, Alice and Bob, to securely establish a shared private key by exchanging data over a public channel, where the third party could eavesdrop the communication. Let $p$ be a prime, $g \in \mathbb{F}_p^*$, and $p$ and $g$ be public. The protocol proceeds as follows: Alice picks $a$ randomly and uniformly from $\mathbb{Z}/(p-1)\mathbb{Z}$, computes $A = g^a \pmod p$, and sends $A$ to Bob. Bob picks $b$ randomly and uniformly from $\mathbb{Z}/(p-1)\mathbb{Z}$, computes $B = g^b \pmod p$, and sends $B$ to Alice. Alice computes the private key $C = B^a \pmod p$ and Bob computes $C = A^b \pmod p$. An eavesdropper can obtain $p, g, A, B$. However, we have no efficient algorithm that on input $(p, g, A, B)$, outputs $C \in \mathbb{F}_p^*$ such that $C = g^{ab}$, $A = g^a$, and $B = g^b$.

Breaking the Diffie-Hellman scheme trivially reduces to solving the discrete logarithm problem, or to deciding membership in $\pi_{DL}$, and hence to $\Pi_{DL}$ with respect to the $\leq^p_m$-reducibility. Another problem related to the Diffie-Hellman scheme is on the decision Diffie-Hellman problem (DDH for short) [1, 8]. Define $\text{DDH} = \{(p, g, A, B, C) \exists (a, b) \in \mathbb{Z}/(p-1)\mathbb{Z}[1 \equiv g^{a+b} \pmod p) \land (C \equiv g^{ab} \pmod p))\}$. Since $\text{DDH}$ is in NP, it reduces to some NP-complete set. We give in the following corollary a direct reduction from $\text{DDH}$ to $\Pi_{DL}$ with respect to the $\leq^p_m$-reducibility.

Corollary 3.4. $\text{DDH} \leq^p_m \Pi_{DL}$.

Proof. $(p, g, A, B, C) \in \text{DDH} \iff (p, (g, g, g), (X_1, X_2, X_1X_2), (A, B, C), (p-2, p-2)) \in \Pi_{DL}$. \hfill \Box
Recall that \( \pi_{DL} \) has self-computable solutions. Since \( \Pi_{DL} \) is an extension of \( \pi_{DL} \), the next proposition immediately follows.

**Proposition 3.5.** \( \Pi_{DL} \) has self-computable solutions.

**Proof.** Let \((p, g, F, y, d)\) be in \( \Pi_{DL} \). Then it is possible to find \( \delta = (\delta_1, \ldots, \delta_k) \leq d \) such that \((p, g, F, y, \delta) \in \Pi_{DL} \) by making queries to the oracle \( \Pi_{DL} \). The strategy is to simply execute the binary search algorithm for \( k \) times, and this leads to the smallest \( \delta \). Here we say that \( \delta = (\delta_1, \ldots, \delta_k) \) is the smallest if for each \( i \), \((p, g, F, y, (\delta_1, \ldots, \delta_i, \delta_{i+1}) \) is in \( \Pi_{DL} \) but \((p, g, F, y, (\delta_1, \ldots, \delta_i - 1, \delta_{i+1}) \) is not in \( \Pi_{DL} \). Therefore, we have \( \bigwedge_{i=1}^{k} x_i \equiv g_i^{(x_i \equiv h_1, \ldots, x_k \equiv h_k)} \mod p \). Since the binary search algorithm runs in polynomial time, the overall running time is also bounded by some polynomial. Thus \( \Pi_{DL} \) has self-computable solutions.

### 3.2 Integer factoring problem

**Definition 3.6.** Let \( n \in \mathbb{N}, F \in \mathbb{Z}/n\mathbb{Z}[X_1, \ldots, X_k] \), and \( d = (d_1, \ldots, d_k) \in (\mathbb{Z}/n\mathbb{Z})^k \). We define the set \( \Pi_{IF} \) as
\[
\Pi_{IF} = \{(n, F, d) \mid \exists x = (x_1, \ldots, x_k) \in (\mathbb{Z}/n\mathbb{Z})^k \\mid \langle 1 < F(X_1 - x_1, \ldots, X_k - x_k) < n \rangle \wedge (F(X_1 - x_1, \ldots, X_k - x_k) | n) \wedge (x \leq d) \}\}.
\]

**Theorem 3.7.** \( \Pi_{IF} \) is complete for \( \mathbb{NP} \) with respect to the \( \leq_{m}^{P} \)-reducibility.

**Proof.** We reduce SAT to \( \Pi_{IF} \) with respect to the \( \leq_{m}^{P} \)-reducibility. The proof proceeds in the same way as Theorem 3.2. For a Boolean formula \( \varphi(X_1, \ldots, X_k) \), let \( F_{\varphi}(X_1, \ldots, X_k) \in \mathbb{F}_2[X_1, \ldots, X_k] \) be its arithmetization.

We transform \( \varphi \) into the following form:
\[(n, F, d) = (4, F_{\varphi} + 1, (3, \ldots, 3)).\]

This \((4, F_{\varphi} + 1, (3, \ldots, 3))\) is in \( \Pi_{IF} \) if and only if there exists \((x_1, \ldots, x_k) \in (\mathbb{Z}/4\mathbb{Z})^k \) such that \( F(X_1 - x_1, \ldots, X_k - x_k) \equiv 2 \mod 4 \) because \( 2 \) is the only nontrivial factor of \( 4 \). \( F(X_1 - x_1, \ldots, X_k - x_k) \equiv 2 \mod 4 \) implies \( F_{\varphi}(X_1 - x_1, \ldots, X_k - x_k) \equiv 1 \mod 2 \). This means that \((x_1, \ldots, x_k) \mod 2\) is an assigning assignment for \( \varphi[X_1, \ldots, X_k] \). This completes the proof.

The next corollary implies that \( \Pi_{IF} \leq_{m}^{P} \Pi_{IF} \).

**Corollary 3.8.** \( \pi_{IF} \leq_{m}^{P} \Pi_{IF} \).

**Proof.** \((n, k) \in \pi_{IF} \iff (n, X, k) \in \Pi_{IF} \)

We can prove that \( \Pi_{IF} \) has self-computable solutions similarly to Proposition 3.5.

**Proposition 3.9.** \( \Pi_{IF} \) has self-computable solutions.

### 4. Discussion on the Complexity of \( \Pi_{DL} \) and \( \Pi_{IF} \)

We have seen that \( \Pi_{DL} \) and \( \Pi_{IF} \) are complete for \( \mathbb{NP} \) with respect to the \( \leq_{m}^{P} \)-reducibility. Those two sets are closely related to the security of cryptographic schemes based on the discrete logarithm problem and the integer factoring problem, as described in §3. Although these results do not immediately influence the security of existing cryptographic schemes, exploring the properties specific to \( \Pi_{DL} \) and \( \Pi_{IF} \) could lead to a breakthrough on an efficient algorithm for some restricted variants of \( \Pi_{DL} \) or \( \Pi_{IF} \), and could affect the security. It is therefore worth knowing any complexity-theoretic result concerning these sets.

In this section, we look into one of such properties. Namely, we prove that \( \Pi_{DL} \) or \( \Pi_{IF} \) can remain \( \mathbb{NP} \)-complete when parameters of the set are restricted. We will show that under some weak assumptions, \( \Pi_{DL} \) remains \( \mathbb{NP} \)-complete even if \( F = (F_1, \ldots, F_\ell) \in (\mathbb{Z}/(p - 1)\mathbb{Z}[X_1, \ldots, X_k])^\ell \) is restricted to \( \ell = 1 \) and \( k = 1 \), and that \( \Pi_{IF} \) remains \( \mathbb{NP} \)-complete even if the degree of \( F \in \mathbb{Z}/n\mathbb{Z}[X_1, \ldots, X_k] \) is restricted to \( 4k \).

#### 4.1 \( \Pi_{DL} \)

We show that a known number-theoretic \( \mathbb{NP} \)-complete set reduces to a restricted version of \( \Pi_{DL} \) under two assumptions: the Extended Riemann Hypothesis (ERH) and the Heath-Brown Conjecture (HBC). We start with introducing the two assumptions and the \( \mathbb{NP} \)-complete set.

ERH is the hypothesis concerning the property of zeros of an extended version of the Riemann zeta function. Therefore the ordinary Riemann hypothesis follows from ERH. A succinct explanation of ERH is found in the appendix of [12]. The point is that the hypothesis implicates the distribution of primitive roots modulo a prime. It is known that, for any prime \( p \), the least primitive root mod \( p \) is bounded above by some polynomial in \( |p| \) if ERH holds [18], where \(|p|\) denotes the length of \( p \) in the binary form. This implies that under ERH, the least primitive root mod \( p \) is so small that we can find it by an exhaustive search in deterministic polynomial time in \(|p| \) if the prime factorization of \( p - 1 \) is given [3].
The second assumption is on the distribution of primes in arithmetic progressions. Although HBC has not yet been proved, it is believed to be true [21]. HBC asserts that for \( n \geq 2, a \geq 1 \) with \( \gcd(n, a) = 1 \), the least prime of the form \( kn + a \) (for \( k \geq 0 \)) is bounded above by \( c|n|^2 \), where \( c \) is a constant and \( |n| \) is the length of \( n \) in the binary form [6]. In other words, given \( n \) and \( a \) satisfying the conditions above, we can find \( k \) such that \( kn + a \) is prime by an exhaustive search in deterministic polynomial time in \( |n| \) if HBC holds. It is not known to hold that ERH implies HBC or its converse.

The last component of our reduction is the following number-theoretic NP-complete set:

\[
Q = \{(n, a, d) | \exists x \in \mathbb{Z}/n\mathbb{Z}[(x^2 \equiv a \pmod{n}) \land (x \leq d)]\}.
\]

This set is complete for \( \mathbf{NP} \) with respect to the \( \leq \text{P}_m \)-reducibility [11]. Further it remains NP-complete even if the prime factorization of \( n \) is given [11]. Namely,

\[
Q_j = \{(n, a, d, ((p_1, e_1), \ldots, (p_r, e_r))) | \exists x \in \mathbb{Z}/n\mathbb{Z}[(x^2 \equiv a \pmod{n}) \land (x \leq d) \land (n = p_1^{e_1} \cdots p_r^{e_r})]\}
\]

is NP-complete, where the description for verifying \( p_i \in \text{Primes} \) and \( e_i > 0 \) (\( 1 \leq i \leq r \)) is omitted for simplicity.

Now we are ready to show the following statement.

**Proposition 4.1.** Let \( \Pi_{DL} \) be as defined in Definition 3.1. Under ERH and HBC, \( \Pi_{DL} \) remains NP-complete even if \( F = (F_1, \ldots, F_t) \in (\mathbb{Z}/(p - 1)\mathbb{Z}[X_1, \ldots, X_t])^t \) is restricted to \( \ell = 1 \) and \( k = 1 \).

**Proof.** Let \( \Pi_{DL}[c_{-1}] \) denote the restricted version of \( \Pi_{DL} \) defined as

\[
\Pi_{DL}[c_{-1}] = \{(p, g, F(X), y, d) | \exists x \in \mathbb{Z}/(p - 1)\mathbb{Z}[(y \equiv g^{f(X-x)} (p \pmod{p}) \land (x \leq d))]\}.
\]

We show that \( Q_j \) reduces to \( \Pi_{DL}[c_{-1}] \) with respect to the \( \leq \text{P}_m \)-reducibility.

Let \( (n, a, d, ((p_1, e_1), \ldots, (p_r, e_r))) \) be an instance of \( Q_j \). We transform this into an instance of \( \Pi_{DL}[c_{-1}] \) by the following steps.

i. On input \( (n, a, d, ((p_1, e_1), \ldots, (p_r, e_r))) \), search the least \( k \) such that \( p = kn + 1 \) is prime, and compute the prime factorization of \( k \).

Under HBC, this can be done in deterministic polynomial time in \( |n| \). Since \( k \) is so small as \( O(|n|^2) \), it can be factored also in deterministic polynomial time in \( |n| \).

ii. Search the least primitive root \( g \pmod{p} \), using the prime factorization of \( n \) and \( k \).

Under ERH, such \( g \) can be found in deterministic polynomial time in \( |p| \). Note that in general, \( g \) is a primitive root \( \pmod{p} \) if and only if for any prime \( q \) dividing \( p - 1 \), \( g^{\frac{p - 1}{q}} \neq 1 \pmod{p} \) [3], and by the input and the step above, we have the prime factorization of \( kn = p - 1 \).

iii. Output \( (p, g^k, X^2, g^{ku}, d) \).

Since the order of the base \( g^k \) is \( n \), \( (p, g^k, X^2, g^{ku}, d) \) is in \( \Pi_{DL}[c_{-1}] \) if and only if there exists \( x \in \mathbb{Z}/n\mathbb{Z} \) such that \( x^2 \equiv a \pmod{n} \) and \( x \leq d \). Hence \( Q_j \leq \text{P}_m \Pi_{DL}[c_{-1}] \), and the statement follows.

4.2 \( \Pi_{IF} \)

In Theorem 3.7, we have proved the NP-completeness of \( \Pi_{IF} \) by reducing from SAT. Some restricted version of SAT, for example (3,4)-SAT which will be defined later, is known to remain NP-complete [20], so the restricted version of \( \Pi_{IF} \) remains NP-complete as long as we can reduce (3,4)-SAT to it. In fact we show that \( \Pi_{IF} \) remains NP-complete even if the degree of \( F \in \mathbb{Z}/n\mathbb{Z}[X_1, \ldots, X_t] \) is restricted to 4k, by reducing from (3,4)-SAT.

Let \( \varphi(X_1, \ldots, X_t) = \bigwedge_{i=1}^{t} C_i \) be a Boolean formula in conjunctive normal form, where \( C_i \) are clauses. We call \( \varphi \) an \( (r, s) \)-formula if every clause contains exactly \( r \) variables, and every variable occurs in at most \( s \) clauses. The \( (r, s) \)-SAT is the SAT restricted to \( (r, s) \)-formulas. It is shown in [20] that (3,4)-SAT, a variation of 3-SAT [4], is NP-complete, whereas every (3,3)-formula is satisfiable and 2-SAT is in \( \mathbf{P} \) [4]. Thus, (3,4)-SAT is the tightest restriction so that \( (r, s) \)-SAT can remain NP-complete.

**Proposition 4.2.** Let \( \Pi_{IF} \) be as defined in Definition 3.6. \( \Pi_{IF} \) remains NP-complete even if the degree of \( F \in \mathbb{Z}/n\mathbb{Z}[X_1, \ldots, X_t] \) is restricted to 4k.

**Proof.** Let \( \varphi(X_1, \ldots, X_t) \) be a (3,4)-formula, and let \( F(X_1, \ldots, X_t) \) be the polynomial as the arithmetization of \( \varphi \). Since every variable appears in at most four clauses, the degree of \( F \) is at most 4k. This means that (3,4)-formula reduces to \( \Pi_{IF} \) even if the degree of \( F \) is restricted to 4k. Hence the restricted version of \( \Pi_{IF} \) remains NP-complete.

5. Concluding Remarks

Related to the discrete logarithm problem modulo a prime (DL) and the integer factoring problem (IF), we have given two NP-complete sets \( \Pi_{DL} \) and \( \Pi_{IF} \), and shown that they have direct polynomial-time many-one reductions from \( \pi_{DL} \) and \( \pi_{IF} \), respectively. Since \( DL \leq^P \Pi_{DL} \) and \( IF \leq^P \Pi_{IF} \), we consequently have \( DL \leq^P \Pi_{DL} \) and \( IF \leq^P \Pi_{IF} \), respectively.
We do not think that this paper just made a long list of known NP-complete sets a bit longer. These two sets are the first NP-complete sets that have straightforward reductions from standard and significant cryptographic primitives. The proofs of Corollaries 3.3, 3.4 and 3.8 would be applied to breaking some cryptographic scheme if an efficient algorithm for some restricted variants of $\Pi_{DL}$ or $\Pi_{IF}$ were developed, where the properties presented in Propositions 4.1 and 4.2 would be significant for the development. Our technique used in this paper could be helpful for constructing an NP-complete set reducible directly from some cryptographic primitives other than DL and IF.

Acknowledgments

The authors would like to thank the two anonymous reviewers for their valuable comments and fruitful suggestions.

REFERENCES