Computational Indistinguishability between Quantum States and Its Cryptographic Application

Akinori Kawachi[1], Takeshi Koshiba[2], Harumichi Nishimura[3], and Tomoyuki Yamakami[4]

[1] Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Ookayama 2-12-1, Meguro-ku, Tokyo 152-8552, Japan
kawachi@is.titech.ac.jp

[2] Secure Computing Laboratory, Fujitsu Laboratories Ltd., 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki 211-8588, Japan
koshiba@acm.org

[3] ERATO Quantum Computation and Information Project, Japan Science and Technology Agency, Matsuo-bldg.2F, 406 Iseya-cho, Kamigyo-ku, Kyoto 602-0873, Japan
hnishimura@qci.jst.go.jp

[4] Computer Science Program, Trent University, Peterborough, Ontario, Canada K9J 7B8
TomoyukiYamakami@TrentU.CA

Abstract. We introduce a problem of distinguishing between two quantum states as a new underlying problem to build a computational cryptographic scheme that is "secure" against quantum adversary. Our problem is a natural generalization of the distinguishability problem between two probability distributions, which are commonly used in computational cryptography. More precisely, our problem QSCD_{ff} is the computational distinguishability problem between two types of random coset states with a hidden permutation over the symmetric group. We show that (i) QSCD_{ff} has the trapdoor property; (ii) the average-case hardness of QSCD_{ff} coincides with its worst-case hardness; and (iii) QSCD_{ff} is at least as hard in the worst case as the graph automorphism problem. Moreover, we show that QSCD_{ff} cannot be efficiently solved by any quantum algorithm that naturally extends Shor's factorization algorithm. These cryptographic properties of QSCD_{ff} enable us to construct a public-key cryptosystem, which is likely to withstand any attack of a quantum adversary.

1 Introduction

Since Diffie and Hellman [15] first used a computationally intractable problem to build a key exchange protocol, computational cryptography has been extensively investigated; especially, a number of practical cryptographic systems (e.g., public-key cryptosystems (PKCs), bit commitment schemes (BCSs), pseudorandom generators, and digital signature schemes) have been constructed under reasonable computational assumptions, such as the hardness of the integer factorization problem (IFP) and the discrete logarithm problem (DLP), where we have not found any efficient classical (deterministic or probabilistic) algorithm. Nevertheless, if an adversary runs a *quantum computer* (we call such an adversary a *quantum adversary*), he can efficiently solve various problems, including IFP (and quadratic residuosity problem) [40], DLP (and Diffie-Hellman problem) [10, 26, 40], and the principal ideal problem [22]. Therefore, the quantum adversary can easily break any cryptosystem whose security relies on the hardness of these problems.

A new area of cryptography, so-called *quantum cryptography*, has emerged to deal with quantum adversary and has been dramatically developed over the past two decades. In 1984, Bennett and Brassard [7] proposed a *quantum key distribution* scheme, which is a key distribution protocol using quantum communication. Later, Mayers [33] proved its unconditional security. Nevertheless, Mayers [32] and Lo and Chau [30] independently demonstrated that quantum mechanics cannot necessarily make all cryptographic schemes information-theoretically secure. In particular, they proved that no quantum BCS can be both concealing and binding unconditionally. Therefore, it is still important to take "computational" approaches to quantum cryptography. In the literature, there are a number of quantum cryptographic properties discussed from the complexity-theoretic point of view [1, 12–14, 16, 36].

Recall that a quantum computer is capable of breaking many computational assumptions on which the security of existing cryptographic protocols rely. To build a secure cryptosystem against any attack of a quantum adversary, it is important to discover computationally-hard problems that can be used as a building block of the cryptosystem. For example, the subset sum (knapsack) problem and the shortest vector problem are used as a basis of knapsack-based cryptosystems [24, 36] and lattice-based cryptosystems [4, 38]. Although quantum adversaries are currently ineffective in the attack on these cryptosystems, it is unknown whether they can essentially withstand quantum adversaries. We therefore continue searching for better underlying problems to build quantum cryptosystems which can withstand any attack of quantum adversaries. We discuss this issue in depth in Section 1.2.

This paper proposes a *new* problem, called $\mathrm{QSCD}_{\mathit{ff}}$ (quantum state computational distinguishability with fully flipped permutations), which satisfies useful cryptographic properties to build a quantum cryptosystem. Our problem $\mathrm{QSCD}_{\mathit{ff}}$ generalizes the distinguishability problems between two probability distributions used in [8, 18, 43].

Definition 1. The *advantage* of a polynomial-time quantum algorithm $\mathcal{A}$ that distinguishes between two l-qubit states ρ_0 and ρ_1 is the function $\delta(l)$ defined as:

$$\delta(l) = \left| \Pr_{\mathcal{A}}[\mathcal{A}(\rho_0) = 1] - \Pr_{\mathcal{A}}[\mathcal{A}(\rho_1) = 1] \right|,$$

where the subscript $\mathcal{A}$ means that outputs of $\mathcal{A}$ are determined randomly by measuring the final state of $\mathcal{A}$ on the computational basis. The distinguishability problem between ρ_0 and ρ_1 is said to be *solvable by $\mathcal{A}$ with absolute (infinitely-often, resp.) advantage $\delta(l)$* if the above equation holds for any sufficiently large (infinitely many, resp.) number l.

The problem QSCD_{ff} is defined as the distinguishability problem between two random coset states ρ_π^+ and ρ_π^- with a hidden permutation π. Let S_n be the symmetric group of degree n and let $\mathcal{K}_n = \{\pi \in S_n : \pi^2 = id \text{ and } \forall i \in \{1, ..., n\}[\pi(i) \neq i]\}$, where n is described as $2(2k+1)$ for some $k \in \mathbb{N}$.

Definition 2. QSCD_{ff} is the distinguishability problem between the following two quantum states:

$$\rho_\pi^+ = \frac{1}{2n!}\sum_{\sigma\in S_n}(|\sigma\rangle+|\sigma\pi\rangle)(\langle\sigma|+\langle\sigma\pi|) \text{ and } \rho_\pi^- = \frac{1}{2n!}\sum_{\sigma\in S_n}(|\sigma\rangle-|\sigma\pi\rangle)(\langle\sigma|-\langle\sigma\pi|),$$

where $\pi \in \mathcal{K}_n$.

The parameter n of the above definition is used to measure the computational complexity of our problem and is called the *security parameter* in the cryptographic context. From a technical reason, this security parameter must be of the form $2(2k+1)$ for a certain $k \in \mathbb{N}$ as stated above. Moreover, we assume that any permutation σ can be represented in binary using $O(n \log n)$ bits.

1.1 Our Contributions

This paper shows three cryptographic properties of QSCD_{ff} and its application to quantum cryptography. These properties are summarized as follows: (1) QSCD_{ff} has the trapdoor property; namely, given a hidden permutation π, we can efficiently distinguish between ρ_π^+ and ρ_π^-; (2) the average-case hardness of QSCD_{ff} over randomly chosen permutations $\pi \in \mathcal{K}_n$ coincides with its worst-case hardness; and (3) the hardness of QSCD_{ff} is lower-bounded by the worst-case hardness of the graph automorphism problem, defined as

GRAPH AUTOMORPHISM PROBLEM: (GA)
input: an undirected graph $G = (V, E)$;
output: YES if G has a non-trivial automorphism, and NO otherwise.

Since GA is not known to be solved efficiently, QSCD_{ff} seems hard to solve. Moreover, we show that QSCD_{ff} cannot be efficiently solved by any quantum algorithm that naturally extends Shor's factorization algorithm.

Technically speaking, the cryptographic properties of QSCD_{ff} follows mainly from the definition of the set $\mathcal{K}_n$ of the hidden permutations. Although the definition seems somewhat artificial, the following properties of $\mathcal{K}_n$ lead to cryptographic and complexity-theoretic properties of QSCD_{ff}: (i) $\pi \in \mathcal{K}_n$ is of order 2, which provides the trapdoor property of QSCD_{ff}. (ii) For any $\pi \in \mathcal{K}_n$, the conjugacy class of π is equal to $\mathcal{K}_n$, which enables us to prove the equivalence between the worst-case/average-case hardness of QSCD_{ff}. (iii) GA is (polynomial-time Turing) equivalent to its subproblem with the promise that a given graph has a unique non-trivial automorphism in $\mathcal{K}_n$ or none at all. This equivalence is exploited to give a complexity-theoretic lower bound of QSCD_{ff}, that is, the worst-case hardness of GA. For these proofs, we introduce new techniques: a new version of the so-called *coset sampling method*, which is broadly used in extensions of Shor's algorithm (see, e.g., [37]) and a quantum version of the hybrid argument, which is a strong tool for security reduction in modern cryptography.

As for an application of QSCD_{ff}, we also construct a public-key cryptosystem. Several advantages of using QSCD_{ff} will be discussed in depth in Section 1.2.

1.2 Comparison between Our Work and Previous Work

In recent literature, computational-complexity aspects of quantum states have been spotlighted in connection to quantum cryptography. For instance, the notion of statistical distinguishability between quantum states was investigated by Watrous [42] and Kobayashi [27] in the context of quantum zero-knowledge proofs. They proved that certain problems of statistically distinguishing between two quantum states are promise-complete for quantum zero-knowledge proof systems. Aharonov and Ta-Shma [2] also studied the computational complexity of quantum-state generation and showed its connection to quantum adiabatic computing and statistical zero-knowledge proofs.

Our distinguishability problem QSCD_{ff} is also rooted in computational complexity theory. In this subsection, we briefly discuss various advantages of using QSCD_{ff} as a basis of quantum cryptosystems in comparison with other existing cryptosystems and their underlying problems.

Average-case Hardness versus Worst-case Hardness. In general, the efficient solvability of a problem on average does not guarantee that the problem can be solved efficiently by a worst-case algorithm. It is therefore desirable to show that the average-case hardness of cracking a cryptographic system is equivalent to its worst-case hardness. Unfortunately, there are few cryptographic problems known to be reduced from average-case hardness to worst-case hardness.

There are two types of worst-case/average-case reductions discussed in the literature. The first one is a strong reduction, which transforms an arbitrary instance of length n to a random instance of the same length or length polynomial in n. Ajtai [3] found a remarkable connection between the average-case and the worst-case hardness for certain versions of the shortest vector problem (SVP) in this strong sense. He showed an efficient reduction from the problem of approximating the shortest vector in a given n-dimensional lattice in the worst case to the approximation problem of the shortest vectors in a random lattice over a certain class of lattices with a larger polynomial approximation factor in n. A reduction between average-case and worst-case hardness has since then been extensively studied. Micciancio and Regev [34], for instance, gave the average-case/worst-case connection factor of approximately n for approximating SVP (see [9] by Bogdanov and Trevisan and references therein for general results with respect to worst-case/average-case reductions).

The second type of reduction is a weak reduction of Tompa and Woll [41], where the reduction is randomized only over part of its instances. A typical example is DLP, which can be randomly reduced to itself by a reduction that maps instances to not all instances of the same length but rather all instances of the same underlying group. It is, nonetheless, unknown that there exists a reduction from DLP with the worst-case prime to DLP with a random prime.

In this paper, we show that $\mathrm{QSCD}_{f\!f}$ has a worst-case/average-case reduction of the first kind. Our reduction depends only on the length of the instance unlike a reduction for DLP and the average-case hardness of $\mathrm{QSCD}_{f\!f}$ coincides with its worst-case hardness unlike reductions for lattice problems. Note that DLP and the inverting problem of the RSA function, whose worst-case/average-case reductions are of the second kind, can be efficiently solved in the worst case by Shor's algorithm [40]. The graph isomorphism problem (GI) and GA—well-known graph-theoretical problems—also have the connection of the second kind [41]. Although no efficient quantum algorithm is discovered yet for them, there is no known cryptographic system whose security are reduced from them. Our distinguishability problem $\mathrm{QSCD}_{f\!f}$ is the first *cryptographic* problem with the worst-case/average-case reduction of the first kind, which has not been solved efficiently on a quantum computer.

Most problems seem to lack any strong connection between their average-case harness and worst-case hardness. In particular, there is no known cryptographic system that is based on the worst-case hardness of the subset sum problem or its subproblems.

Exponential time versus Subexponential time. The *hidden subgroup problem* (HSP) has been a central issue discussed for both positive and negative aspects of the power of quantum computation. Both IFP and DLP can be viewed as special cases of HSP on Abelian groups (AHSP). Kitaev [26] showed that AHSP can be efficiently solved. He introduced a polynomial-time algorithm for the quantum Fourier transform on Abelian groups, which is a generalization of the original quantum Fourier transform used in Shor's algorithm [40]. Although AHSP can be efficiently solved, the more general non-Abelian group case is unlikely to be solved by simply applying currently known techniques. (Some special non-Abelian group cases were studied in [17, 20, 23, 29, 35, 37].) Another important variant is the HSP on the dihedral groups (DHSP). Recently, Regev [37] demonstrated a quantum reduction from the unique shortest vector problem (uSVP) to a slightly different variant of DHSP. Note that uSVP is used in lattice-based PKCs [4, 38]. Moreover, Kuperburg [29] gave a subexponential-time quantum algorithm for DHSP. Although these results do not directly imply a subexponential-time quantum algorithm for uSVP, they may be an important clue to find the desired algorithm.

Our problem $\mathrm{QSCD}_{f\!f}$ is closely related to a much harder problem: HSP on the symmetric groups (SHSP). No subexponential-time quantum algorithm is known for SHSP. A distinguishability problem, similar to $\mathrm{QSCD}_{f\!f}$, defined in terms of SHSP was introduced by Hallgren, Russell and Ta-Shma [23], who showed that any standard algorithm[5] takes exponential time to solve their problem. Here, we show that their problem is polynomial-time reducible to $\mathrm{QSCD}_{\mathit{ff}}$. This immediately implies that that any standard algorithm that solves $\mathrm{QSCD}_{\mathit{ff}}$ also requires exponential time. The hardness result of Hallgren et al. was recently strengthened by Grigni et al. [20] and Kempe and Shalev [25]. Finding even a subexponential algorithm for $\mathrm{QSCD}_{\mathit{ff}}$ seems a daunting task. On the contrary, this suggests that our problem $\mathrm{QSCD}_{\mathit{ff}}$ is more reliable than, e.g., uSVP. This situation is similar to the case of DLP over different groups on classical computation. DLP over $\mathbb{Z}_p^*$ (p is a prime) is classically solved in subexponential time whereas there is no known classical subexponential-time algorithm for DLP over certain groups used in elliptic curve cryptography. It is believed that DLP over such groups is more reliable than DLP over $\mathbb{Z}_p^*$.

We prove that the computational complexity of $\mathrm{QSCD}_{\mathit{ff}}$ is lower-bounded by that of GA, which is not known to be in $\mathrm{NP} \cap \text{co-NP}$. Well-known upper bounds of GA are $\mathrm{NP} \cap \text{co-AM}$ [19, 39], SPP [5], and UAP [11]. To our best knowledge, most cryptographic problems fall in $\mathrm{NP} \cap \text{co-NP}$ and few cryptographic systems are lower-bounded by the worst-case hardness of the problems not known to be in $\mathrm{NP} \cap \text{co-NP}$.

Quantum Computational Cryptography. Quantum key distribution gives a foundation to symmetric-key cryptosystems (SKCs). For instance, the quantum key distribution scheme in [7] achieves unconditionally secure sharing of secret keys for SKCs using an authenticated classical communication channel. Both SKCs and PKCs have their own advantages and disadvantages. For instance, PKCs save a number of secret keys compared with SKCs in a large network; however, they need computational assumptions for their security and is vulnerable to, for instance, the man-in-the-middle attack. As an application of $\mathrm{QSCD}_{\mathit{ff}}$, we propose a new computational quantum PKC whose security relies on the computational hardness of $\mathrm{QSCD}_{\mathit{ff}}$.

Of many existing PKCs, few make their security solely based on the worst-case hardness of their underlying problems. Quantum adversaries can break many PKCs whose underlying problems are number-theoretic problems because these problems are solvable by efficient quantum algorithms. Recently, Okamoto, Tanaka, and Uchiyama [36] proposed a quantum analogue of PKCs based on a certain subset of the knapsack problem and showed that their cryptosystem withstands certain known attacks of a quantum adversary. Our quantum PKC also seems to resist a quantum adversary since we can prove the existence of a security reduction from the problem GA, which is not known to be solved efficiently even on a quantum computer.

2 Cryptographic Properties of $\mathrm{QSCD}_{\mathit{ff}}$

We show three cryptographic properties of $\mathrm{QSCD}_{\mathit{ff}}$ introduced in the previous section. These properties will help us construct a cryptographic system in Sec-

[5] The algorithms that run an essential part of Shor's algorithm [40] are simply called *standard methods.*

tion 3. Hereafter, we assume the reader's familiarity with basics of quantum computation. Recall the two quantum states $\rho_\pi^+ = \frac{1}{2n!}\sum_{\sigma\in S_n}(|\sigma\rangle + |\sigma\pi\rangle)(\langle\sigma| + \langle\sigma\pi|)$ and $\rho_\pi^- = \frac{1}{2n!}\sum_{\sigma\in S_n}(|\sigma\rangle - |\sigma\pi\rangle)(\langle\sigma| - \langle\sigma\pi|)$ for a hidden permutation $\pi \in \mathcal{K}_n$. For simplicity, let ι denote the maximally mixed state, i.e., $\iota = \frac{1}{n!}\sum_{\sigma\in S_n}|\sigma\rangle\langle\sigma|$, which appears later as a technical tool.

2.1 Trapdoor Property

We prove that QSCD_{ff} has the *trapdoor property*, which plays a key role in various cryptosystems. We present an efficient distinction algorithm between ρ_π^+ and ρ_π^- with a hidden permutation π in $\mathcal{K}_n$.

Theorem 1. There exists a polynomial-time quantum algorithm that, given $\pi \in \mathcal{K}_n$, distinguishes between ρ_π^+ and ρ_π^- with certainty.

Proof. Let χ be any given unknown state, which is either ρ_π^+ or ρ_π^-. The desired distinction algorithm for χ is given as follows.

(D1) Prepare two quantum registers: the first register holds a control bit and the second one holds χ. Apply the Hadamard transformation H to the first register. The state of the system now becomes $H|0\rangle\langle 0|H \otimes \chi$.

(D2) Apply the Controlled-π operator C_π to the two registers, where $C_\pi|0\rangle|\sigma\rangle = |0\rangle|\sigma\rangle$ and $C_\pi|1\rangle|\sigma\rangle = |1\rangle|\sigma\pi\rangle$ for any $\sigma \in S_n$. Since $\pi^2 = id$ for any $\pi \in \mathcal{K}_n$, the state of the entire system is expressed as $\frac{1}{n!}\sum_{\sigma\in S_n}|\psi_{\pi,\sigma}^+\rangle\langle\psi_{\pi,\sigma}^+|$ if $\chi = \rho_\pi^+$ and $\frac{1}{n!}\sum_{\sigma\in S_n}|\psi_{\pi,\sigma}^-\rangle\langle\psi_{\pi,\sigma}^-|$ if $\chi = \rho_\pi^-$, where

$$\begin{aligned}|\psi_{\pi,\sigma}^\pm\rangle &= C_\pi\left(\frac{1}{2}|0\rangle\left(|\sigma\rangle \pm |\sigma\pi\rangle\right) + |1\rangle\left(|\sigma\rangle \pm |\sigma\pi\rangle\right)\right)\\ &= \frac{1}{2}|0\rangle(|\sigma\rangle \pm |\sigma\pi\rangle) + \frac{1}{2}|1\rangle(|\sigma\pi\rangle \pm |\sigma\rangle).\end{aligned}$$

(D3) Apply the Hadamard transformation to the first register. If χ is ρ_π^+ and ρ_π^-, then the state of the system becomes $(H \otimes I)|\psi_{\pi,\sigma}^+\rangle = \frac{1}{\sqrt{2}}|0\rangle\left(|\sigma\rangle + |\sigma\pi\rangle\right)$ and $(H \otimes I)|\psi_{\pi,\sigma}^-\rangle = \frac{1}{\sqrt{2}}|1\rangle\left(|\sigma\rangle - |\sigma\pi\rangle\right)$, respectively. Measure the first register on the computational basis. If the result is 0, output YES; otherwise, output NO. Clearly, we obtain the correct answer with probability 1. □

2.2 Reduction from the Worst Case to the Average Case

We reduce the worst-case hardness of QSCD_{ff} to its average-case hardness. Such a reduction implies that QSCD_{ff} with a random π is at least as hard as QSCD_{ff} with the most difficult π.

Theorem 2. Assume that there exists a polynomial-time quantum algorithm $\mathcal{A}$ that solves QSCD_{ff} with absolute (infinitely-often, resp.) non-negligible advantage for a uniformly random $\pi \in \mathcal{K}_n$; namely, there exists a polynomial p

such that, for any sufficiently large (infinitely many, resp.) number n,

$$\left| \Pr_{\pi,\mathcal{A}}[\mathcal{A}(\rho_\pi^+) = 1] - \Pr_{\pi,\mathcal{A}}[\mathcal{A}(\rho_\pi^-) = 1] \right| > 1/p(n),$$

where π is chosen uniformly at random from $\mathcal{K}_n$. Then, there exists a polynomial-time quantum algorithm $\mathcal{B}$ that solves QSCD$_{\mathit{ff}}$ with absolute (infinitely-often, resp.) non-negligible advantage in the worst case.

Proof. Let χ be any given state, which is either ρ_π^+ or ρ_π^-. The desired worst-case algorithm $\mathcal{B}$ is built from the average-case algorithm $\mathcal{A}$ in the following way.

(R1) Choose a permutation $\tau \in S_n$ uniformly at random and then multiply χ by τ from the right. If $\chi = \rho_\pi^+$, then we obtain the quantum state

$$\begin{aligned}\chi' &= \frac{1}{2n!} \sum_{\sigma \in S_n} (|\sigma\tau\rangle + |\sigma\tau\tau^{-1}\pi\tau\rangle)(\langle\sigma\tau| + \langle\sigma\tau\tau^{-1}\pi\tau|) \\ &= \frac{1}{2n!} \sum_{\sigma' \in S_n} (|\sigma'\rangle + |\sigma'\tau^{-1}\pi\tau\rangle)(\langle\sigma'| + \langle\sigma'\tau^{-1}\pi\tau|).\end{aligned}$$

If $\chi = \rho_\pi^-$, then we obtain $\chi' = \frac{1}{2n!} \sum_{\sigma \in S_n} (|\sigma\rangle - |\sigma\tau^{-1}\pi\tau\rangle)(\langle\sigma| - \langle\sigma\tau^{-1}\pi\tau|)$.

(R2) Invoke the average-case algorithm $\mathcal{A}$ on the input χ'.

(R3) Output the outcome of $\mathcal{A}$.

Note that $\tau^{-1}\pi\tau \in \mathcal{K}_n$ for any τ and there exists a $\tau \in S_n$ satisfying that $\tau^{-1}\pi\tau = \pi'$ for any $\pi' \in \mathcal{K}_n$. Hence, the conjugacy class of π is equal to $\mathcal{K}_n$. Moreover, the number of all $\tau \in S_n$ for which $\tau^{-1}\pi\tau = \pi'$ is independent of the choice of $\pi' \in \mathcal{K}_n$. From these properties, $\tau^{-1}\pi\tau$ is uniformly distributed over $\mathcal{K}_n$. Therefore, feeding the input χ' to algorithm $\mathcal{A}$ guarantees the non-negligible advantage. □

2.3 Hardness of QSCD$_{\mathit{ff}}$

We show that the computational complexity of QSCD$_{\mathit{ff}}$ is lower-bounded by that of GA by constructing an efficient reduction from GA to QSCD$_{\mathit{ff}}$. Our reduction constitutes two parts: a reduction from GA to a variant of GA, called UniqueGA$_{\mathit{ff}}$, and a reduction from UniqueGA$_{\mathit{ff}}$ to QSCD$_{\mathit{ff}}$. We also discuss a relationship between QSCD$_{\mathit{ff}}$ and SHSP, which suggests that QSCD$_{\mathit{ff}}$ may be hard for polynomial-time quantum algorithms to solve.

To describe the desired reduction, we begin with introducing two variants of GA. Earlier, Köbler, Schöning and Torán [28] introduced the following *unique graph automorphism problem* (UniqueGA).

Unique Graph Automorphism Problem: (UniqueGA)
input: an undirected graph $G = (V, E)$;
promise: G has a unique non-trivial automorphism or no non-trivial automorphisms;
output: YES if G has the non-trivial automorphism, and NO otherwise.

Notice that UniqueGA is called (1GA, GA) as a promise problem in [28]. Connection to our distinguishability problem, we introduce the *unique graph automorphism with fully-flipped permutation* (UniqueGA$_{\mathit{ff}}$), which plays an important role in the reduction.

> Unique Graph Automorphism with Fully-Flipped Permutation: (UniqueGA$_{\mathit{ff}}$)
> input: an undirected graph $G = (V, E)$, where $|V| = n = 2(2k+1)$ for some $k \in \mathbb{N}$;
> promise: G has a unique non-trivial automorphism $\pi \in \mathcal{K}_n$, or no non-trivial automorphisms;
> output: YES if G has the non-trivial automorphism, and NO otherwise.

Next, we discuss the so-called *coset sampling method*, which has been largely used in many extensions of Shor's algorithm.

Lemma 1. There exists a polynomial-time quantum algorithm that, given an instance G of UniqueGA$_{\mathit{ff}}$, generates a quantum state ρ_π^+ if G is an "YES" instance with its unique non-trivial automorphism π, or $\iota = \frac{1}{n!}\sum_{\sigma\in S_n}|\sigma\rangle\langle\sigma|$ if G is a "NO" instance.

Proof. Given an instance G of UniqueGA$_{\mathit{ff}}$, we first prepare the quantum state $\frac{1}{\sqrt{n!}}\sum_{\sigma\in S_n}|\sigma\rangle|\sigma(G)\rangle$, where $\sigma(G)$ is the graph resulting from by relabeling its nodes according to a permutation σ. By discarding the second register, we obtain the unique quantum state χ in the first register. Then, $\chi = \rho_\pi^+$ if G is an "YES" instance with the unique non-trivial automorphism π, and $\chi = \iota$ otherwise, as requested. □

Now, we introduce a new version of the coset sampling method as a technical tool for our reduction. Note that this algorithm essentially requires the fact that the hidden π is an odd permutation, which is one of the special properties of $\mathcal{K}_n$.

Lemma 2. There exists a polynomial-time quantum algorithm that, given an instance G of UniqueGA$_{\mathit{ff}}$, generates a quantum state ρ_π^- if G is an "YES" instance with the unique non-trivial automorphism π, or ι if G is a "NO" instance.

Proof. Similar to the algorithm of Lemma 1, we prepare the quantum state $\frac{1}{\sqrt{n!}}\sum_{\sigma\in S_n}|\sigma\rangle|\sigma(G)\rangle$. Next, we compute the sign of each permutation in the first register and then invert its phase if the permutation is odd. We obtain the quantum state $\frac{1}{\sqrt{n!}}\sum_{\sigma\in S_n}(-1)^{\mathrm{sgn}(\sigma)}|\sigma\rangle|\sigma(G)\rangle$, where $\mathrm{sgn}(\sigma) = 0$ if σ is even, and $\mathrm{sgn}(\sigma) = 1$ otherwise. By discarding the second register, we can obtain a quantum state χ in the first register. Note that, since π is odd, if σ is odd (even, resp.) then $\sigma\pi$ is even (odd, resp.). Therefore, $\chi = \rho_\pi^-$ if G is an "YES" instance with the unique non-trivial automorphism π, and $\chi = \iota$ otherwise. □

We are now ready to present a reduction from GA to QSCD$_{\mathit{ff}}$, which implies that QSCD$_{\mathit{ff}}$ is computationally at least as hard as GA.

Theorem 3. If there exists a polynomial-time quantum algorithm that solves QSCD_{ff} with absolute non-negligible advantage, there exists a polynomial-time quantum algorithm that solves any instance of GA in the worst case with non-negligible probability.

Proof. We first show that GA is polynomial-time Turing equivalent to $\mathrm{UniqueGA}_{ff}$ and then give a reduction from $\mathrm{UniqueGA}_{ff}$ to QSCD_{ff}. The reduction from GA to $\mathrm{UniqueGA}_{ff}$ is similar to the one given by Köbler, Schöning and Torán [28], who presented a polynomial-time Turing reduction from GA to UniqueGA. Their polynomial-time algorithm for GA invokes UniqueGA as an oracle with a promised input, that is, a graph with even number of nodes which has either the unique non-trivial automorphism without fixed points or no non-trivial automorphisms. Carefully reading the construction of their reduction, we can easily modify it to fit our reduction from GA to $\mathrm{UniqueGA}_{ff}$. Moreover, slightly modifying the gadgets for their reduction, we can satisfy the condition that $n = 2(2k+1)$ for some $k \in \mathbb{N}$. Thus, we obtain the following lemma.

Lemma 3. $\mathrm{UniqueGA}_{ff}$ is polynomial-time Turing equivalent[6] to GA.

The complete proof of this lemma is placed in Appendix. It therefore suffices to show a reduction from $\mathrm{UniqueGA}_{ff}$ to QSCD_{ff}. Assume that there exists a polynomial-time quantum algorithm $\mathcal{A}$ that solves QSCD_{ff} with absolute non-negligible advantage. For a given instance G of $\mathrm{UniqueGA}_{ff}$, we perform the following procedure:

(S1) Generate two sequences $S^+ = (\chi^+, \ldots, \chi^+)$ and $S^- = (\chi^-, \ldots, \chi^-)$ of $8p^2(n)n$ quantum states from G using the algorithms of Lemmas 1 and 2, respectively.
(S2) Invoke $\mathcal{A}$ on each component in S^+ and S^- as an input. Let $R^+ = (\mathcal{A}(\chi^+), \ldots, \mathcal{A}(\chi^+))$ and $R^- = (\mathcal{A}(\chi^-), \ldots, \mathcal{A}(\chi^-))$ be the resulting sequences.
(S3) Output YES if the gap between the numbers of 1's in R^+ and R^- is at least $4p(n)n$, output NO otherwise.

Note that if G is an "YES" instance, $S^+ = (\overbrace{\rho_\pi^+, \ldots, \rho_\pi^+}^{8p^2(n)n})$ and $S^- = (\overbrace{\rho_\pi^-, \ldots, \rho_\pi^-}^{8p^2(n)n})$, otherwise $S^+ = S^- = (\overbrace{\iota, \ldots, \iota}^{8p^2(n)n})$. Therefore, if G is an "YES" instance, then there is a gap between the numbers of 1's in R^+ and in R^- because of the property of $\mathcal{A}$; otherwise, there is no gaps between them.

We now estimate this gap by the Hoeffding bound. Let X^+ and X^- be two random variables expressing the numbers of 1's in R^+ and in R^-, respectively. If G is an "YES" instance, $\Pr[|X^+ - X^-| > 4p(n)n] > 1 - 2e^{-n}$ by the Hoeffding bound since $|\Pr[\mathcal{A}(\rho_\pi^+) = 1] - \Pr[\mathcal{A}(\rho_\pi^-) = 1]| > 1/p(n)$. Similarly, if G is a "NO" instance, $\Pr[|X^+ - X^-| < 4p(n)n] > 1 - 2e^{-n}$. This guarantees the above procedure to solve $\mathrm{UniqueGA}_{ff}$ efficiently, as requested. □

[6] If a Turing reduction to a promise problem makes only queries that satisfy the promise, the reduction is called *smart* [21]. Smart reductions are desirable for security reductions. The reduction from GA to UniqueGA in [28] is indeed smart and thus, so is this reduction.

As stated in Section 1, the distinguishability problem QSCD_{ff} is rooted in SHSP. It is shown that a natural extension of Shor's algorithm cannot solve the distinguishability problem between ρ_π^+ and ι in [23, 20, 25]. Here, we give a theorem on a relationship between QSCD_{ff} and the distinguishability problem between ρ_π^+ and ι.

Before stating the theorem, we give a conversion algorithm for ρ_π^+ and ρ_π^-. This algorithm will be used in the proof of the theorem as well as the construction of a PKC in the subsequent section.

Lemma 4. There exists a polynomial-time quantum algorithm that converts ρ_π^+ into ρ_π^- and keeps ι as it is with certainty.

Proof. Given ρ_π^+, the desired algorithm inverts its phase according to the sign of the permutation by performing the following transformation:

$$|\sigma\rangle + |\sigma\pi\rangle \longmapsto (-1)^{\mathrm{sgn}(\sigma)}|\sigma\rangle + (-1)^{\mathrm{sgn}(\sigma\pi)}|\sigma\pi\rangle.$$

Recall that $\mathrm{sgn}(\sigma) = 0$ if σ is even and $\mathrm{sgn}(\sigma) = 1$ otherwise. Note that deciding the sign of a given permutation takes only polynomial time. Since π is odd, the above algorithm converts ρ_π^+ into ρ_π^-. Clearly, the algorithm does not alter the quantum state ι. □

The following theorem implies that QSCD_{ff} cannot be efficiently solved by any algorithm that naturally extends Shor's factoring algorithm. To prove the theorem, we need a quantum version of the so-called *hybrid argument*.

Theorem 4. If there exists a polynomial-time quantum algorithm that solves QSCD_{ff} with absolute (infinitely-often, resp.) non-negligible advantage, then there exists a polynomial-time quantum algorithm that solves the distinguishability problem between ρ_π^+ and ι with absolute (infinitely-often, resp.) non-negligible advantage.

Proof. We prove only the absolute advantage case. Assume that a polynomial-time quantum algorithm $\mathcal{A}$ solves QSCD_{ff} with absolute non-negligible advantage; namely, there exist a number $n_0 \geq 1$ and a polynomial $q(n)$ such that

$$\left|\Pr_{\mathcal{A}}[\mathcal{A}(\rho_\pi^+) = 1] - \Pr_{\mathcal{A}}[\mathcal{A}(\rho_\pi^-) = 1]\right| > 1/q(n)$$

for all numbers $n \geq n_0$. Let $\mathcal{A}'$ be the algorithm that applies the conversion algorithm of Lemma 4 to a given state χ $(= \rho_\pi^+$ or $\iota)$ and then feeds the resulting state χ' $(= \rho_\pi^-$ or $\iota)$ to $\mathcal{A}$. Note that $\mathcal{A}'(\rho_\pi^+) = \mathcal{A}(\rho_\pi^-)$ and $\mathcal{A}'(\iota) = \mathcal{A}(\iota)$. It immediately follows by the triangle inequality that, for any number $n \geq n_0$,

$$\left|\Pr_{\mathcal{A}}[\mathcal{A}(\rho_\pi^+) = 1] - \Pr_{\mathcal{A}}[\mathcal{A}(\iota) = 1]\right| + \left|\Pr_{\mathcal{A}'}[\mathcal{A}'(\rho_\pi^+) = 1] - \Pr_{\mathcal{A}'}[\mathcal{A}'(\iota) = 1]\right| > 1/q(n).$$

This inequality implies that, for each number $n \geq n_0$, we obtain either

$$\left|\Pr_{\mathcal{A}}[\mathcal{A}(\rho_\pi^+) = 1] - \Pr_{\mathcal{A}}[\mathcal{A}(\iota) = 1]\right| > 1/2q(n)$$

or

$$\left|\Pr_{\mathcal{A}'}[\mathcal{A}'(\rho_\pi^+) = 1] - \Pr_{\mathcal{A}'}[\mathcal{A}'(\iota) = 1]\right| > 1/2q(n).$$

The desired algorithm $\mathcal{B}$ first chooses either $\mathcal{A}$ or $\mathcal{A}'$ at random and then simulates the chosen algorithm. Obviously, this algorithm solves the distinguishability problem between ρ_π^+ and ι with absolute non-negligible advantage, completing the proof. □

3 Application

We have shown useful cryptographic properties of QSCD$_{ff}$. As an application of QSCD$_{ff}$, we build a quantum public-key cryptosystem (PKC) whose security relies on the hardness of QSCD$_{ff}$. First, we give an efficient quantum algorithm that generates ρ_π^+ from π.

Lemma 5. There exists a polynomial-time quantum algorithm that, given $\pi \in \mathcal{K}_n$, generates the quantum state ρ_π^+ with certainty.

Proof. The desired generation algorithm uses two registers and is given as follows. The correctness of the algorithm is obvious.

(G1) Choose a permutation σ from S_n uniformly at random and store it in the second register. Then, the entire system is in the state $|0\rangle|\sigma\rangle$.
(G2) Apply the Hadamard transformation to the first register: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\sigma\rangle$.
(G3) Apply the Controlled-π to the both registers: $\frac{1}{\sqrt{2}}(|0\rangle|\sigma\rangle + |1\rangle|\sigma\pi\rangle)$.
(G4) Apply the Hadamard transformation to the first register again: $\frac{1}{2}((|0\rangle + |1\rangle)|\sigma\rangle + (|0\rangle - |1\rangle)|\sigma\pi\rangle)$.
(G5) Measure the first register on the computational basis. If 0 is observed, then the quantum state in the second register is ρ_π^+. Otherwise, the state of the second register is ρ_π^-. Now, apply the conversion algorithm given in Lemma 4 to ρ_π^-. □

Next, we describe our quantum PKC and give its security proof. For the security proof, we need to specify the model of attacks. Of all attack models in [6], we pay our attention to a quantum analogue of *the indistinguishability against the chosen plaintext attack (IND-CPA)*. In particular, we adopt the weakest scenario in quantum counterparts of IND-CPA as follows.

Alice (sender) wants to send securely a classical message to Bob (receiver) via a quantum channel. Assume that Alice and Bob are polynomial-time quantum Turing machines. Bob first generates certain quantum states for encryption keys. Alice then requests Bob for his encryption keys. Note that anyone can request him for the encryption keys. Now, we assume that Eve (adversary) can pick up the encrypted messages from the quantum channel, and tries to extract the original message using her quantum computer, i.e., a polynomial-time quantum Turing machine. Since Eve can also obtain Bob's encryption keys as well as Alice does, she can exploit polynomially many encryption keys to distinguish the encrypted message. Thus, we assume that Eve attacks the protocol during the message transmission phase to reveal the content of the encrypted message.

The protocol to transmit a message using our PKC consists of two phases: the key transmission phase and the message transmission phase. We will give a reduction from the worst-case hardness of GA to such Eve's attack.

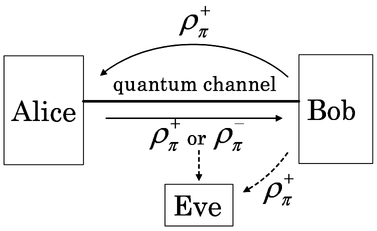

Fig. 1: our public-key cryptosystem

We first describe the protocol of our quantum PKC as follows.
[Key transmission phase]
(A1) Bob chooses a decryption key π uniformly at random from $\mathcal{K}_n$.
(A2) Bob generates sufficiently many copies of the encryption key ρ_π^+.
(A3) Alice obtains encryption keys from Bob.
[Message transmission phase]
(A4) Alice encrypts 0 or 1 into ρ_π^+ or ρ_π^-, respectively, and sends it to Bob.
(A5) Bob decrypts Alice's message using the decryption key π.

Step (A1) can be easily implemented by uniformly choosing transpositions one by one in such a way that all transpositions are different and by forming the product of these transpositions. Step (A2) is done by the generation algorithm of Lemma 5. For Step (A4), we exploit the conversion algorithm of Lemma 4. Note that Alice sends Bob either the received state ρ_π^+ or its converted state ρ_π^- depending on Alice's bit. Finally, the distinction algorithm in Theorem 1 achieves Step (A5).

The security of our PKC is shown by reducing GA to Eve's attack during the message transmission phase. Our reduction is a modification of the reduction given in Theorem 3.

Proposition 1. Assume that there exists a polynomial-time quantum adversary $\mathcal{A}$ in the message transmission phase that, for any sufficiently large n, satisfies the following inequality

$$\left|\Pr_{\pi,\mathcal{A}}[\mathcal{A}(\rho_\pi^+, \rho_\pi^{+\otimes l(n)}) = 1] - \Pr_{\pi,\mathcal{A}}[\mathcal{A}(\rho_\pi^-, \rho_\pi^{+\otimes l(n)}) = 1]\right| > 1/p(n)$$

for a certain polynomial $l(n)$ indicating the number of the encryption keys in use by $\mathcal{A}$ and another polynomial $p(n)$. Then, there exists a polynomial-time quantum algorithm that solves any instance of GA in the worst case with non-negligible probability.

Proof. The proof immediately follows by replacing ρ_π^+, ρ_π^-, and ι in the proof of Theorem 3 with $(\rho_\pi^+, \rho_\pi^{+\otimes l(n)})$, $(\rho_\pi^-, \rho_\pi^{+\otimes l(n)})$, and $(\iota, \iota^{\otimes l(n)})$, respectively. $\square$

4 Concluding Remarks

The computational distinguishability problem $\mathrm{QSCD}_{\mathit{ff}}$ has shown useful properties to build a computational PKC whose security is based on the computational hardness of GA. Although GA is reducible to $\mathrm{QSCD}_{\mathit{ff}}$, the gap between the hardness of GA and that of $\mathrm{QSCD}_{\mathit{ff}}$ seems large because a combinatorial structure of its underlying graphs which GA enjoys is completely lost in $\mathrm{QSCD}_{\mathit{ff}}$. It is therefore important to discover a classical problem, such as the problems of finding a centralizer or finding a normalizer [31], which captures the true hardness of $\mathrm{QSCD}_{\mathit{ff}}$. Discovering an efficient quantum algorithm for $\mathrm{QSCD}_{\mathit{ff}}$ is likely to require a new tool and a new technique, which also bring a breakthrough in quantum computation. It is important to discover useful quantum states whose computational distinguishability is used for constructing a more secure cryptosystem.

Acknowledgments. The authors are grateful to Hirotada Kobayashi and Claude Crépeau for fruitful discussions, to John Watrous for useful comments on key ideas, to Donald Beaver, Louis Salvail, and the anonymous reviewers for their valuable suggestions.

References

1. M. Adcock and R. Cleve. A quantum Goldreich-Levin theorem with cryptographic applications. In *Proc. 19th Symp. Theoretical Aspects of Computer Science*, LNCS 2285, pp.323–334 (2002).
2. D. Aharonov and A. Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proc. 35th ACM Symp. Theory of Computing*, pp.20–29 (2003).
3. M. Ajtai. Generating hard instances of lattice problems. In *Proc. 28th ACM Symp. Theory of Computing*, pp.99–108 (1996).
4. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. 29th ACM Symp. Theory of Computing*, pp.284–293 (1997).
5. V. Arvind and P. P. Kurur. Graph isomorphism is in SPP. In *Proc. 43rd IEEE Symp. Foundations of Computer Science*, pp.743–750 (2002).
6. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology – CRYPTO'98*, pp.26–45 (1998).
7. C. H. Bennett and G. Brassard. Quantum cryptography: public key distribution and coin tossing. In *Proc. IEEE International Conf. Computers, Systems, and Signal Processing*, pp.175–179 (1984).
8. M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13(4):850–864, 1984.
9. A. Bogdanov and L. Trevisan. On worst-case to average-case reductions for NP problems. In *Proc. 44th IEEE Symp. Foundations of Computer Science*, pp.308–317 (2004).
10. D. Boneh and R. J. Lipton. Quantum cryptanalysis of hidden linear functions. In *Advances in Cryptology – CRYPTO'95*, LNCS 963, pp.424–437 (1995).
11. M. Crâsmaru, C. Glaßer, K. W. Regan, and S. Sengupta. A protocol for serializing unique strategies. In *Proc. 29th Symp. Mathematical Foundations of Computer Science*, LNCS 3153, pp.660–672 (2004).

12. C. Crépeau, P. Dumais, D. Mayers, and L. Salvail. Computational collapse of quantum state with application to oblivious transfer. In *Proc. 1st Theory of Cryptography Conf.*, LNCS 2951, pp.374–393 (2004).
13. C. Crépeau, F. Légaré, and L. Salvail. How to convert the flavor of a quantum bit commitment. In *Advances in Cryptology – EUROCRYPT'01*, LNCS 2045, pp.60–77 (2001).
14. I. Damgård, S. Fehr, and L. Salvail. Zero-knowledge proofs and string commitments withstanding quantum attacks. In *Advances in Cryptology – CRYPTO'04*, LNCS 3152, pp.254–272 (2004).
15. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.
16. P. Dumais, D. Mayers, and L. Salvail. Perfectly concealing quantum bit commitment from any quantum one-way permutation. In *Advances in Cryptology – EUROCRYPT 2000*, LNCS 1807, pp.300–315 (2000).
17. M. Ettinger and P. Høyer. On quantum algorithms for noncommutative hidden subgroups. *Advances in Applied Mathematics*, 25:239–251, 2000.
18. S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. System Sci.*, 28(2):270–299, 1984.
19. S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof system. In *Advances in Computing Research, Vol. 5: Randomness and Computation*, pp.73–90. JAI Press, 1989.
20. M. Grigni, L. J. Schulman, M. Vazirani, and U. Vazirani. Quantum mechanical algorithms for the nonabelian hidden subgroup problem. In *Proc. 33rd ACM Symp. Theory of Computing*, pp.68–74 (2001).
21. J. Grollmann and A. L. Selman. Complexity measures for public-key cryptosystems. *SIAM J. Comput.*, 17(2):309–335, 1988.
22. S. Hallgren. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. In *Proc. 34th ACM Symp. Theory of Computing*, pp.653–658 (2002).
23. S. Hallgren, A. Russell, and A. Ta-Shma. The hidden subgroup problem and quantum computation using group representations. *SIAM J. Comput.*, 32(4):916–934, 2003.
24. R. Impagliazzo and M. Naor. Efficient cryptographic schemes provably as secure as subset sum. *J. Cryptology*, 9(4):199–216, 1996.
25. J. Kempe and A. Shalev. The hidden subgroup problem and permutation group theory. In *Proc. 16th ACM-SIAM Symp. Discrete Algorithms*, 2005.
26. A. Kitaev. Quantum measurements and the Abelian stabilizer problem. quant-ph/9511026, 1995.
27. H. Kobayashi. Non-interactive quantum perfect and statistical zero-knowledge. In *Proc. 14th International Conf. Algorithms and Computation*, LNCS 2906, pp.178–188 (2003).
28. J. Köbler, U. Schöning, and J. Torán. *The Graph Isomorphism Problem: Its Structural Complexity*. Birkhäuser Boston Inc., 1993.
29. G. Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. quant-ph/0302112, 2003.
30. H.-K. Lo and H. F. Chau. Is quantum bit commitment really possible? *Physical Review Letters*, 78(17):3410–3413, 1997.
31. E. M. Luks. Permutation groups and polynomial-time computation. *Groups and Computation*, 11:139–175, 1993.
32. D. Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical Review Letters*, 78(17):3414–3417, 1997.

33. D. Mayers. Unconditional security in quantum cryptography. *J. ACM*, 48(3):351–406, 2001.
34. D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measure. In *Proc. 45th IEEE Symp. Foundations of Computer Science*, pp.372–381 (2004).
35. C. Moore, D. Rockmore, A. Russell, and L. J. Schulman. The hidden subgroup problem in affine groups: basis selection in Fourier sampling. In *Proc. 15th ACM-SIAM Symp. Discrete Algorithms*, pp.1106–1115 (2004).
36. T. Okamoto, K. Tanaka, and S. Uchiyama. Quantum public-key cryptosystems. In *Advances in Cryptology – CRYPTO 2000*, LNCS 1880, pp.147–165 (2000).
37. O. Regev. Quantum computation and lattice problems. In *Proc. 43rd IEEE Symp. Foundations of Computer Science*, pp.520–529 (2002).
38. O. Regev. New lattice-based cryptographic constructions. In *Proc. 35th ACM Symp. Theory of Computing*, pp.407–416 (2003).
39. U. Schöning. Graph isomorphism is in the low hierarchy. *J. Comput. System Sci.*, 37:312–323, 1988.
40. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26:1484–1509, 1997.
41. M. Tompa and H. Woll. Random self-reducibility and zero knowledge interactive proofs of possession of information. In *Proc. 28th IEEE Symp. Foundations of Computer Science*, pp.472–482 (1987).
42. J. Watrous. Limits on the power of quantum statistical zero-knowledge. In *Proc. 43rd IEEE Symp. Foundations of Computer Science*, pp.459–468 (2002).
43. A. C.-C. Yao. Theory and applications of trapdoor functions. In *Proc. 23rd IEEE Symp. Foundations of Computer Science*, pp.80–91 (1982).

Appendix: Reduction from GA to UniqueGA$_{\mathit{ff}}$

In this Appendix, we prove Lemma 3. Köbler, Schöning and Torán [28] proved the polynomial-time Turing equivalence between GA and UniqueGA. We first review their reduction and then show how to modify it to obtain the reduction from GA to UniqueGA$_{\mathit{ff}}$. Note that the reduction from UniqueGA$_{\mathit{ff}}$ to GA is obvious.

We begin with a technical tool and notations. The reduction of Köbler et al. uses a technical tool called a *label* to distinguish each node of a given graph G from the others. The label j attached to node i consists of two chains, one of which is of length $2n+3$ connected to node i and the other is of length j connected to the $n+2$-nd node of the first chain (Fig. 2).

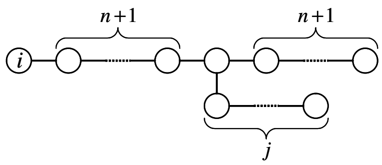

Fig. 2: label

Note that the total size of the label j is $2n+j+3$. Let $G_{[i]}$ denote the graph obtained from G by attaching label 1 to node i. Similarly, $G_{[i_1,\ldots,i_j]}$ is defined as the graph with labels $1, ..., j$ respectively attached to nodes $i_1, ..., i_j$. Note that any automorphism of $G_{[i]}$ maps the node i into itself and any label adds no new automorphism into the modified graph. Let $Aut(G)$ be the automorphism group of the graph G and let $Aut(G)_{[1,...,i]}$ be the point-wise stabilizer of $\{1, ..., i\}$ in $Aut(G)$, i.e., $Aut(G)_{[1,...,i]} = \{\sigma \in Aut(G) : \forall j \in \{1, ..., i\}[\sigma(j) = j]\}$.

Köbler et al. [28] proved the following theorem. The reduction from GA to UniqueGA in [28] is described in its proof.

Theorem 5. [28, Theorem 1.31] GA is polynomial-time Turing reducible to UniqueGA.

Proof. Given an oracle $\mathcal{O}$ for UniqueGA, the following algorithm solves GA in polynomial time. Let G be any given instance of GA.

(U1) Repeat (U2)-(U3) for each i starting with n down to 1.
(U2) Repeat (U3) for each j ranging from $i + 1$ to n.
(U3) Invoke $\mathcal{O}$ with input graph $G_{[1,...,i-1,i]} \cup G_{[1,...,i-1,j]}$. If the outcome of $\mathcal{O}$ is YES, output YES and halt.
(U4) Output NO.

If G is an "YES" instance, there is at least one non-trivial automorphism. Take the largest number $i \in \{1, ..., n\}$ such that there exists a number $j \in \{1, ..., n\}$ and a non-trivial automorphism $\pi \in Aut(G)_{[1,...,i]}$ for which $\pi(i) = j$ and $i \neq j$. We claim that there is exactly one such non-trivial automorphism. This is seen as follows. First, note that $Aut(G)_{[1,...,i-1]}$ is expressed as $Aut(G)_{[1,...,i-1]} = \pi_1 Aut(G)_{[1,...,i]} + \cdots + \pi_d Aut(G)_{[1,...,i]}$. For any two distinct cosets $\pi_s Aut(G)_{[1,...,i]}$ and $\pi_t Aut(G)_{[1,...,i]}$ and for any two automorphisms $\sigma \in \pi_s Aut(G)_{[1,...,i]}$ and $\sigma' \in \pi_t Aut(G)_{[1,...,i]}$, it holds that $\sigma(i) \neq \sigma'(i)$. Since $|Aut(G)_{[1,...,i]}| = 1$ and there exists the unique coset $\pi_k Aut(G)$ such that $\sigma(i) = j$ for any $\sigma \in \pi_k Aut(G)$ by the definition of i, we obtain $|\pi_k Aut(G)_{[1,...,i]}| = 1$. This implies that the non-trivial automorphism π is unique. Note that the unique non-trivial automorphism interchanges two subgraphs $G_{[1,...,i-1,i]}$ and $G_{[1,...,i-1,j]}$. Therefore, the above algorithm successfully outputs YES at Step (U3).

On the contrary, if G is a "NO" instance, then for every distinct i and j, the modified graph has no non-trivial automorphism. Thus, the above algorithm correctly rejects such a graph G. □

Finally, we describe the reduction from GA to UniqueGA$_{ff}$ by slightly modifying the reduction given in the above proof.

Lemma 6. GA is polynomial-time Turing reducible to UniqueGA$_{ff}$.

Proof. We only need to change the number of nodes to invoke oracle UniqueGA$_{ff}$ in (U3). To do so, we first modify the size of each label. Since the number m of all nodes $G_{[1,...,i-1,i]} \cup G_{[1,...,i-1,j]}$ is even, if there is no k such that $m = 2(2k + 1)$ then we add one more node appropriately to the original labels. We then attach our modified labels of length $2n+i+4$ and $2n+j+4$ to nodes i and j, respectively. Note that this modified graph satisfies the promise of UniqueGA$_{ff}$. Our algorithm therefore works correctly for any instance of GA. □