Card-based Zero-knowledge Proof Protocols for Graph Problems and Their Computational Model*

Daiki Miyahara[1,2], Hiromichi Haneda[1], and Takaaki Mizuki[1,2]

[1] Tohoku University, Sendai, Japan
daiki.miyahara.q4[atmark]alumni.tohoku.ac.jp,
mizuki+lncs[atmark]tohoku.ac.jp
[2] National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Abstract. Zero-Knowledge Proof (ZKP) is a cryptographic technique that enables a prover to convince a verifier that a given statement is true without revealing any information other than its truth. It is known that ZKP can be realized by physical objects such as a deck of cards; recently, many such "card-based" ZKP protocols for pencil puzzles (such as Sudoku and Numberlink) have been devised. In this paper, we shift our attention to graph theory problems from pencil puzzles: We propose card-based ZKP protocols for the graph 3-coloring problem and the graph isomorphism problem. Similar to most of the existing card-based ZKP protocols, our two protocols have no soundness error. The proposed protocols can be implemented without any technical knowledge, and the principle of zero-knowledge proof is easy to understand. In particular, our protocol for the graph isomorphism problem requires only three shuffles regardless of the sizes of a pair of given graphs. In addition, to deal with our proposed protocols more rigorously, we present a formal framework for card-based ZKP protocols which are non-interactive and have no soundness error.

Keywords: Physical zero-knowledge proof · Card-based cryptography · Graph 3-coloring problem · Graph isomorphism problem

1 Introduction

Suppose that there are two parties, the prover, Peggy, and the verifier, Victor. The prover Peggy has a witness w guaranteeing that a statement x is true, while the verifier Victor does not have it. In this case, a *Zero-Knowledge Proof* (ZKP) protocol, whose concept was devised by Goldwasser et al. in 1989 [9], enables Peggy to convince Victor that the statement x is true without leaking any information about the witness w. Such a ZKP protocol must satisfy the following three conditions.

* This paper appears in Proceedings of ProvSec 2021. The final authenticated version is available online at https://doi.org/10.1007/978-3-030-90402-9_8.

Completeness. If x is true, then Victor accepts.

Soundness. If x is false, then no matter how Peggy behaves, Victor rejects with an overwhelming probability.

Zero-knowledge. No information about w other than the fact that x is true is leaked to Victor.

The probability that Victor accepts even though x is false is called a *soundness error* probability, denoted by δ. If a ZKP protocol having such a probability δ is executed ℓ times, Victor rejects with a probability of $1 - \delta^\ell$. Thus, we can satisfy the soundness condition by repeatedly running such a protocol sufficient times.

Normally, ZKP protocols are implemented on computers and network systems, based on cryptographic primitives, such as public-key cryptography. By contrast, there are *physical* ZKP protocols that do not rely on computers; for example, Gradwolh et al. [10] in 2009 invented the first physical ZKP protocol for Sudoku using a deck of physical cards. This protocol directly verifies a solution of a Sudoku puzzle without reducing it to other NP-complete problems, such as 3SAT. Therefore, a physical ZKP protocol is suitable for visual understanding of the concept of ZKP, as it can be performed with human hands.

It should be noted that because any Boolean circuit can be securely evaluated by card-based cryptography (e.g., [4,14,15,19,21,30]), we can construct a physical ZKP protocol for any 3SAT instance [20].

1.1 Existing Physical ZKP Protocols

Many physical ZKP protocols using a deck of cards have been constructed for Nikoli's pencil puzzles, such as Sudoku [10, 24, 29], Makaro [3], Slitherlink [16], and Numberlink [26]. These protocols are fun, and their proofs can be easily understood because they were presented using pictures of a deck of cards (as will be seen in Section 2).

Going back to history, Goldreich et al. [8] in 1991 proved that, for all languages in NP, there exist ZKP protocols based on cryptographic primitives. In their paper, they also presented a physical ZKP protocol for the 3-coloring problem using boxes having locks to clarify the presentation of their concept. This physical ZKP protocol has a soundness error as will be seen in Section 2.3, meaning that the protocol needs to be repeated many times.

1.2 Contribution

In this paper, we shift our attention to *graph theory* problems from pencil puzzles. We propose card-based ZKP protocols for two famous graph problems: the *3-coloring* problem and the *graph isomorphism* problem. Similar to most of the existing card-based ZKP protocols, our two protocols have no soundness error. The proposed protocols can be implemented without any technical knowledge, and the principle of ZKP is easy to understand. In particular, our protocol for

the graph isomorphism problem requires only three shuffles regardless of how large a pair of given graphs is.

In addition to constructing the two protocols, we present a formal framework for card-based ZKP protocols which are non-interactive and have no soundness error. Using this proposed framework, we can describe such card-based ZKP protocols in a rigorous way.

We emphasize that this paper is an attempt to connect physical ZKP protocols (in cryptology) and graph theory. Hence, we believe that our work explores new directions of physical ZKP protocols toward graph problems. Constructing efficient ZKP protocols for other famous graph problems is an interesting problem, including the ones in Karp's 21 NP-complete problems [13], as a physical ZKP protocol for the Hamiltonian cycle problem has recently been designed [28].

2 Preliminaries

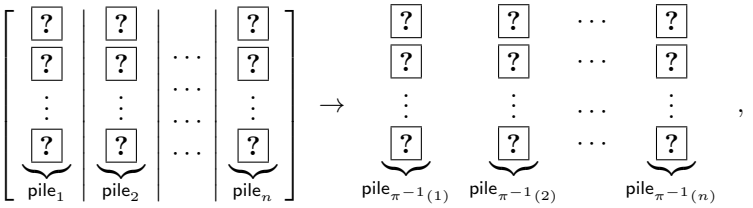
In this section, we introduce notations of a deck of cards and a shuffling action used in our proposed protocols later. We then introduce the 3-coloring problem and the graph isomorphism problem. We also describe an existing protocol for the 3-coloring problem [8].

2.1 A Deck of Cards

Both of our proposed protocols use a two-colored deck of cards, such as black $\boxed{\clubsuit}$ and red $\boxed{\heartsuit}$ cards. In addition, our protocol for the graph isomorphism problem (presented in Section 4) uses numbered cards, such as $\boxed{1}\boxed{2}\boxed{3}\cdots$. The backs of all these cards, denoted by $\boxed{?}$, are indistinguishable.

2.2 Pile-scramble Shuffle

In our construction, we will use a shuffling action called the *pile-scramble shuffle*. This action uniformly shuffles multiple piles of face-down cards at random. More precisely, for some natural number $n\,(\geq 2)$, let $(\mathsf{pile}_1, \mathsf{pile}_2, \ldots, \mathsf{pile}_n)$ denote a sequence of n piles of cards where each pile consists of the same number of cards. Applying a pile-scramble shuffle to such a sequence of piles (denoted by $[\cdot|\ldots|\cdot]$) results in:



where $\pi \in S_n$ is a random permutation uniformly chosen from the symmetric group of degree n, denoted by S_n. In this case, we regard cards in the same

"column" as a pile; thus, the resulting order of cards in each pile does not change. We also consider applying a pile-scramble shuffle "vertically," i.e., cards in the same row are regarded as a pile and all the piles are shuffled.

A pile-scramble shuffle was first used by Ishikawa et al. [12] in 2015. It can be easily implemented by using rubber bands or envelopes to fix each pile of cards and scrambles the piles to randomize the order of them. We assume that, as in the case of usual card games, even if only one player performs a pile-scramble shuffle, nobody (including the executor) can know the resulting order of piles. If some players are skeptical, they may repeat the shuffling action in turn until they are satisfied.

2.3 Known Physical Protocol for 3-coloring Problem [8]

The *3-coloring* problem is a decision problem to determine whether vertices of a given undirected graph $G = (V, E)$ can be colored with three colors such that every two adjacent vertices are assigned different colors. More precisely, the problem is to determine whether there exists a mapping $\phi : V \to \{1, 2, 3\}$ such that any edge $(u, v) \in E$ satisfies $\phi(u) \neq \phi(v)$. This problem is known to be NP-complete [6].

Goldreich et al. [8] in 1991 presented a physical ZKP protocol for the 3-coloring problem. It uses boxes each having a lock with a corresponding key, such as safety boxes. Assuming that Peggy knows a correct coloring ϕ but Victor does not, the protocol proceeds as follows.

1. Let n be the number of vertices in a given graph G. Peggy prepares n boxes and assigns a box to each vertex.
2. Peggy assigns three random colors to $\{1, 2, 3\}$ and puts the corresponding color for each vertex into the box without Victor's seeing it. More precisely, Peggy chooses a random permutation $\pi \in S_3$, and for every $u \in V$, Peggy puts $\pi(\phi(u))$ into the box corresponding to u.
3. Victor randomly chooses one edge $(u, v) \in E$ and tells it to Peggy.
4. Peggy sends Victor the keys to the two boxes corresponding to u and v.
5. Victor opens the two boxes using the keys received. If they contain different colors, then Victor continues to the next iteration; otherwise, Victor rejects.

This protocol satisfies the three conditions required for a ZKP protocol. If Peggy has a correct mapping ϕ, then Peggy can always convince Victor because the two boxes corresponding to the two adjacent vertices chosen by Victor never contain the same color, i.e., $\pi(\phi(u)) \neq \pi(\phi(v))$. If Peggy does not have ϕ, then Victor rejects with a probability of at least $1/m$, where m is the number of edges in the given graph G, i.e., this protocol has a soundness error. By repeating this protocol ℓ times, Victor can detect such a malicious prover Peggy with a probability of $1-(1-\frac{1}{m})^{\ell}$. Since Peggy discloses to Victor only the randomly assigned colors of the two adjacent vertices, Victor cannot obtain more information than the fact that the two vertices are colored with different colors.

2.4 Graph Isomorphism Problem

The *graph isomorphism* problem is a decision problem to determine whether two given undirected graphs are isomorphic. More specifically, given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the problem is to determine whether there exists a permutation $\pi : V_1 \to V_2$ such that $(u, v) \in E_1$ if and only if $(\pi(u), \pi(v)) \in E_2$.

It has been believed that the graph isomorphism problem is neither in P nor NP-complete. A quasi-polynomial time algorithm has been reported by Babai [1, 11].

3 Card-based ZKP for 3-coloring Problem

In this section, we construct a physical ZKP protocol for the 3-coloring problem with no soundness error. As in the existing protocol [8] introduced in Section 2.3, our proposed protocol enables the prover, Peggy, to convince the verifier, Victor, that, for a given undirected graph $G = (V, E)$, Peggy has a mapping $\phi : V \to \{1, 2, 3\}$ such that any edge $(u, v) \in E$ satisfies $\phi(u) \neq \phi(v)$ without revealing any information about ϕ.

In our protocol, Peggy first places sequences of cards representing ϕ that she has, and then Peggy and Victor publicly manipulate the sequences for the verification. Thus, after Peggy places the sequences as input, either Peggy or Victor (or even a third party) may manipulate the cards.

The idea behind our proposed protocol is to verify that *every* pair of adjacent vertices is colored with different colors one by one. Our protocol proceeds as follows.

1. Let $V = \{1, 2, \ldots, n\}$. For every vertex $i \in \{1, 2, \ldots, n\}$, Peggy prepares a sequence of face-down cards representing $\phi(i)$ according to the following encoding rule (with one red card and two black cards):[3]

$$\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit} = 1, \quad \boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit} = 2, \quad \boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit} = 3. \tag{1}$$

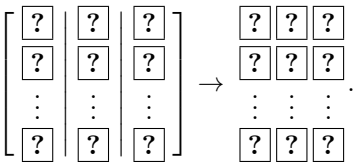
   Place such n sequences vertically one by one as follows:

$$\begin{array}{l} \boxed{?}\boxed{?}\boxed{?} = \phi(1) \\ \boxed{?}\boxed{?}\boxed{?} = \phi(2) \\ \quad\vdots\quad\vdots\quad\vdots \\ \boxed{?}\boxed{?}\boxed{?} = \phi(n) \end{array}.$$

2. For every edge $(i, j) \in E$, perform the following steps. If Victor does not reject for any edge, then Victor accepts.

[3] An encoding rule representing a positive integer in this manner was first considered by Shinagawa et al. in 2015 [31].

(a) Regarding cards in the same column as a pile, apply a pile-scramble shuffle (horizontally) to the n sequences as follows:



Let us emphasize that anyone cannot know the resulting order after performing a pile-scramble shuffle.[4]

(b) Reveal all the cards of the i-th and j-th sequences. If the two sequences represent different colors, i.e., the two red cards are revealed to be at different positions, it means that $\phi(i) \neq \phi(j)$, and hence, continue the protocol after turning over the revealed cards; otherwise, Victor rejects. Note that information about the values of $\phi(i)$ and $\phi(j)$ does not leak because a pile-scramble shuffle has been applied to the sequences in the previous step.

Let m be the number of edges in the given graph G. The number of required cards and shuffles for this protocol is $3n$ and m, respectively.

We present a security proof of this protocol in Section 6.1. This proof is based on our computational model formalized in Section 5.

4 Card-based ZKP for Graph Isomorphism Problem

In this section, we construct a card-based ZKP protocol for the graph isomorphism problem with no soundness error. Our proposed protocol enables Peggy to convince Victor that for two given undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, Peggy has a permutation $\pi : V_1 \to V_2$ (as a witness) such that $(u, v) \in E_1$ if and only if $(\pi(u), \pi(v)) \in E_2$.

4.1 Idea

Assume that Peggy has a correct permutation $\pi \in S_n$ where n denotes the number of vertices in the two given graphs G_1 and G_2. Let $A(G_1)$ and $A(G_2)$ denote their adjacency matrices, respectively. Then, the following equation holds for the permutation matrix P_π corresponding to π [7]:

$$A(G_2) = P_\pi^{\mathrm{T}} A(G_1) P_\pi,$$

[4] One might think that the resulting order could be easily known because there are only six possibilities. One possible implementation is to put piles of cards into a box or ball whose inside is invisible from outside and then throw it up to randomize the order of them (cf. [32]).

where for a row vector $\boldsymbol{e}_i$ in which the i-th element is 1 and the remaining ones are 0, $1 \leq i \leq n$, the permutation matrix P_π is

$$P_\pi = \begin{bmatrix} \boldsymbol{e}_{\pi(1)} \\ \vdots \\ \boldsymbol{e}_{\pi(n)} \end{bmatrix},$$

and P_π^{T} is the transpose. From this equation, it suffices that Peggy and Victor place sequences of face-down cards representing $A(G_1)$, and Peggy having π rearranges the sequences according to the permutation matrix P_π so that the resulting sequences represent $A(G_2)$ (without revealing any information about π).

4.2 Description

In our proposed protocol, Peggy first prepares a sequence of face-down cards representing π, and then Peggy and Victor publicly manipulate the sequences for the verification. Our protocol proceeds as follows.

1. Peggy prepares a sequence of face-down numbered cards from $\boxed{1}$ to $\boxed{n}$ representing the inverse permutation $\pi^{-1} \in S_n$ (that she knows as a witness) according to the following encoding:

   $$\underbrace{\boxed{?}}_{\pi(1)}\underbrace{\boxed{?}}_{\pi(2)}\cdots\underbrace{\boxed{?}}_{\pi(n)} \quad [\pi^{-1}].$$

   This sequence is called the sequence $[\pi^{-1}]$ where the parentheses indicate that all cards in the sequence are face-down.
2. Let $\boxed{\clubsuit}$ represent 0 and $\boxed{\heartsuit}$ represent 1. According to this encoding, place sequences of face-down cards representing the $n \times n$ adjacency matrix of G_1, namely $A(G_1)$. For example, the following 4×4 adjacency matrix is represented using sequences of cards as follows:

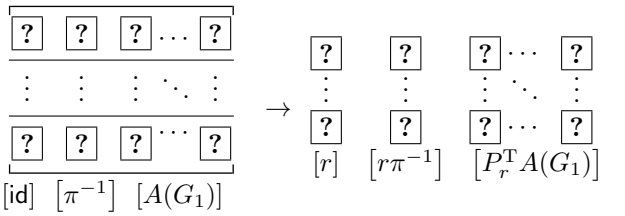
   $$\begin{pmatrix} 0&1&1&0 \\ 1&0&1&1 \\ 1&1&0&0 \\ 0&1&0&0 \end{pmatrix} \rightarrow \begin{array}{cccc} \boxed{\clubsuit}&\boxed{\heartsuit}&\boxed{\heartsuit}&\boxed{\clubsuit} \\ \boxed{\heartsuit}&\boxed{\clubsuit}&\boxed{\heartsuit}&\boxed{\heartsuit} \\ \boxed{\heartsuit}&\boxed{\heartsuit}&\boxed{\clubsuit}&\boxed{\clubsuit} \\ \boxed{\clubsuit}&\boxed{\heartsuit}&\boxed{\clubsuit}&\boxed{\clubsuit} \end{array}.$$

   Then, place the sequence of $[\pi^{-1}]$ that Peggy prepared and a sequence of the identity permutation $[\mathsf{id}]$ consisting of $\boxed{1}\boxed{2}\cdots\boxed{n}$ (in this order) on the left side of the matrix $[A(G_1)]$ vertically, as follows:

   $$\begin{array}{ccc} \boxed{?} & \boxed{?} & \boxed{?}\cdots\boxed{?} \\ \boxed{?} & \boxed{?} & \boxed{?}\cdots\boxed{?} \\ \vdots & \vdots & \vdots\ \ddots\ \vdots \\ \boxed{?} & \boxed{?} & \boxed{?}\cdots\boxed{?} \\ [\mathsf{id}] & [\pi^{-1}] & [A(G_1)] \end{array}.$$

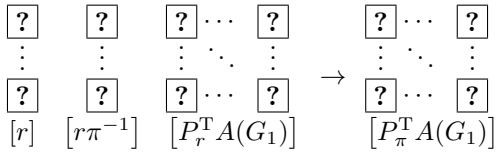
Note that the sequence [id] represents the position of each card in the sequence $[\pi^{-1}]$.

3. Regarding the cards in the same row as a pile, apply a pile-scramble shuffle to the piles as follows:

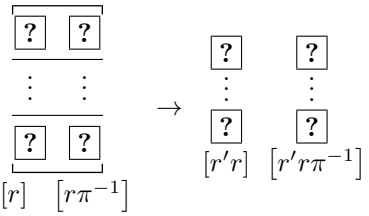


where $r \in S_n$ is a random permutation generated by applying a pile-scramble shuffle.

4. Reveal the sequence $[r\pi^{-1}]$ to obtain the information about $r\pi^{-1}$. Transform the matrix $[P_r^{\mathrm{T}}A(G_1)]$ to $[P_\pi^{\mathrm{T}}A(G_1)]$ by sorting the rows of the matrix in ascending order according to $r\pi^{-1}$:

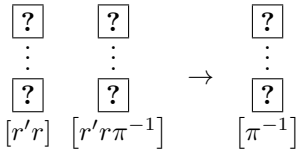


5. Turn over the sequence revealed in the previous step. Then, regarding cards in the same row as a pile, apply a pile-scramble shuffle to the sequences $[r]$ and $[r\pi^{-1}]$ as follows:



where $r' \in S_n$ is a random permutation generated by applying a pile-scramble shuffle.

6. Reveal the sequence $[r'r]$. Sort the sequence $[r'r\pi^{-1}]$ in ascending order according to $r'r$. This sorting applies the inverse permutation $(r'r)^{-1}$ to the sequence $[r'r\pi^{-1}]$, and hence, the sequence $[r'r\pi^{-1}]$ becomes a sequence $[\pi^{-1}]$:



7. Horizontally place the sequence $[\pi^{-1}]$ above the matrix $[P_\pi^{\mathrm{T}} A(G_1)]$ as follows:

$$\begin{array}{ccc} \boxed{?} \cdots \boxed{?} & & [\pi^{-1}] \\ \boxed{?} \cdots \boxed{?} & & \\ \vdots \quad \ddots \quad \vdots & & [P_\pi^{\mathrm{T}} A(G_1)] \\ \boxed{?} \cdots \boxed{?} & & \end{array} .$$

8. Regarding the cards in the same column as a pile, apply a pile-scramble shuffle to the piles:

$$\left[\begin{array}{c|c|c} \boxed{?} & \cdots & \boxed{?} \\ \boxed{?} & \cdots & \boxed{?} \\ \vdots & \ddots & \vdots \\ \boxed{?} & \cdots & \boxed{?} \end{array}\right] \begin{array}{l} [\pi^{-1}] \\ \\ [P_\pi^{\mathrm{T}} A(G_1)] \end{array} \rightarrow \begin{array}{ccc} \boxed{?} \cdots \boxed{?} & & [r''\pi^{-1}] \\ \boxed{?} \cdots \boxed{?} & & \\ \vdots \quad \ddots \quad \vdots & & [P_\pi^{\mathrm{T}} A(G_1) P_{r''}] \\ \boxed{?} \cdots \boxed{?} & & \end{array} ,$$

where $r'' \in S_n$ is a random permutation generated by applying a pile-scramble shuffle.

9. Reveal the sequence $[r''\pi^{-1}]$. Sort the columns of the matrix $[P_\pi^{\mathrm{T}} A(G_1) P_{r''}]$ in ascending order according to $r''\pi^{-1}$ to transform the matrix to $[P_\pi^{\mathrm{T}} A(G_1) P_\pi]$:

$$\begin{array}{ccc} \boxed{?} \cdots \boxed{?} & & [r''\pi^{-1}] \\ \boxed{?} \cdots \boxed{?} & & \\ \vdots \quad \ddots \quad \vdots & & [P_\pi^{\mathrm{T}} A(G_1) P_{r''}] \\ \boxed{?} \cdots \boxed{?} & & \end{array} \rightarrow \begin{array}{ccc} \boxed{?} \cdots \boxed{?} & & \\ \vdots \quad \ddots \quad \vdots & & [P_\pi^{\mathrm{T}} A(G_1) P_\pi] \\ \boxed{?} \cdots \boxed{?} & & \end{array} .$$

10. Reveal all the cards of the matrix $[P_\pi^{\mathrm{T}} A(G_1) P_\pi]$. If they represent the adjacency matrix of G_2, then Victor accepts; otherwise, Victor rejects.

Let m be the number of edges in the given graphs. The total number of required cards is $n^2 + 2n$, because $2m$ $\boxed{\heartsuit}$s and $(n^2 - 2m)$ $\boxed{\clubsuit}$s are used for representing the adjacency matrix of G_1, and $2n$ numbered cards are used for the sequences of $[\pi^{-1}]$ and $[\mathsf{id}]$. The number of required shuffles is three, which is constant regardless of the size of a pair of given graphs.

We present a security proof of this protocol in Section 6.2.

5 Basic Formalization of Card-based ZKP Protocols

In this section, we give a formalization of card-based ZKP protocols to deal with our proposed protocols more rigorously.

Remember that our two protocols presented in Sections 3 and 4 are *non-interactive*: After the prover, Peggy, places a hidden sequence of face-down cards at the beginning of each of the protocols according to a witness (that only Peggy knows), the protocol can be executed by *anyone* publicly; for example, it suffices that Peggy does every action while the verifier, Victor, watches all behaviors of Peggy. Note that most of the existing ZKP protocols for pencil puzzles (e.g. [3, 17, 22, 26, 27, 29]) are also non-interactive.

Thus, this section begins with clarifying the relationship between a witness and a hidden sequence of cards.

5.1 Witness Subsequence

Let $L \subseteq \Sigma^*$ be a language that captures a decision problem (such as the 3-coloring problem and graph isomorphism problem), where Σ is an alphabet. In our setting, Peggy and Victor are given a problem *instance* $x \in L$ such that only Peggy knows a *witness* w of the instance x; w being a witness of x means that, given a pair (x, w), everyone (including Peggy and Victor) can easily confirm that $x \in L$ (say, it can be computed in polynomial time).

For example, given an instance of the graph isomorphism problem, a permutation π which transforms one graph into the other graph serves a witness. As seen in Section 4, Peggy who knows the permutation π is supposed to privately arrange a sequence of face-down cards encoding π:

$$\underbrace{\boxed{?}}_{\pi(1)}\underbrace{\boxed{?}}_{\pi(2)}\cdots\underbrace{\boxed{?}}_{\pi(n)} \quad [\pi^{-1}].$$

Therefore, in general, Peggy and Victor must agree upon a correspondence between a witness and a sequence of cards; we call such a sequence of cards a *witness subsequence*.

Fixing a language L, for an instance $x \in L$, we denote by W_x the set of all witnesses of x. If Peggy knows a witness $w \in W_x$ and she is honest, she should place a witness subsequence correctly (with all cards' faces down); we call it a *correct witness subsequence*. If Peggy does not know any witness, she may place a 'wrong' sequence of cards that follows the 'format' at least; we say a sequence of cards is a *legal witness subsequence* if there exist an instance $y \in L$ and a witness $w' \in W_y$ such that the sequence corresponds to w'. If Peggy is malicious, she may place a random sequence of cards; we call any witness sequence which is not legal an *illegal witness subsequence*.

Let us consider the case where $x \notin L$. If an instance x is clearly outside of L (say, the numbers of edges of G_1 and G_2 are different), Victor would not agree with executing any protocol; therefore, for such an instance, we do not have to construct a protocol. On the other hand, there are instances $x \notin L$ for which we have to construct protocols; define $\breve{L} \subseteq \Sigma^* - L$ as

$$\breve{L} = \{x \notin L \mid \text{Victor cannot determine if } x \in L\}.$$

If Peggy is malicious, she may present an instance $x \in \breve{L}$ to Victor, and place some subsequence to run a protocol (although there is no witness). We call such a subsequence an illegal witness subsequence as well.

Consequently, we are supposed to construct a protocol for every instance in $L \cup \breve{L}$.

5.2 Input to Protocol

As seen above, at the beginning of a protocol, Peggy is supposed to prepare a witness subsequence. In addition to the witness subsequence, we need some *helping cards*; for example, our protocol for the graph isomorphism (presented in

Section 4) uses n^2 black $\boxed{\clubsuit}$ or red $\boxed{\heartsuit}$ cards as well as n numbered cards. These helping cards are placed with their faces up at the beginning of the protocol.

As mentioned, if Peggy is malicious, she may place an illegal witness subsequence. To this end, she may prepare some number of cards stealthily and use them to arrange such an illegal subsequence. Therefore, we have to take into account such stealthy cards (owned by Peggy). Therefore, in addition to the witness subsequence and helping cards, we consider *stealthy cards*: That is, we assume that every input to a protocol consists of these three parts, and that a *deck* D to consider accommodates all these cards.

5.3 Abstract Protocol for ZKP

A card-based protocol itself has been well formalized already [18]. We will slightly adjust the model of protocols by mainly adding a couple of two states, as follows.

First, we review some terms. Let D be a deck containing all cards as mentioned above. We call any element $c \in D$ an atomic card, $\frac{?}{c}$ a face-down card, and $\frac{c}{?}$ a face-up card. Define $\mathrm{top}(\frac{u}{v}) = u$, call

$$\mathrm{top}(\Gamma) = (\mathrm{top}(\alpha_1), \mathrm{top}(\alpha_2), \cdots, \mathrm{top}(\alpha_{|D|}))$$

a visible sequence of a sequence $\Gamma = (\alpha_1, \alpha_2, \ldots, \alpha_{|D|})$, and let Vis^D be the set of all visible sequences from the deck D.

Next, we consider the input to a protocol. As mentioned before, the input consists of three parts: the witness subsequence, helping cards, and stealthy cards. Considering all possible input sequences, we use U to denote the set of all such sequences.

We now consider a state of a protocol. Contrary to the conventional card-based model, we introduce two additional states q_{accept} and q_{reject}. That is, Q is the set of states including the initial state q_0, the accepting state q_{accept}, and the rejecting state q_{reject}. When a protocol terminates with q_{accept}, it means that Victor accepts an input sequence Γ; when it terminates with q_{reject}, it means that Victor rejects.

Based on these definitions and terms, a protocol is defined as follows.

Definition 1. *A card-based protocol P is a 4-tuple $P = (D, U, Q, A)$ that satisfies:*

- *D is a deck.*
- *U is an input set.*
- *Q is a set of states containing q_0, q_{accept}, and q_{reject}.*
- *$A : (Q \backslash \{q_{\mathrm{accept}}, q_{\mathrm{reject}}\}) \times \mathsf{Vis}^D \to Q \times \mathsf{Action}$ is an action function. Here, Action is a set of all actions consisting of the followings.*
  - *Turning over (turn, T): This action is to turn over cards in the positions specified by $T \subseteq \{1, 2, \cdots, |D|\}$. Thus, this transforms a sequence $\Gamma = (\alpha_1, \alpha_2, \cdots, \alpha_{|D|})$ as follows:*

$$\mathsf{turn}_T(\Gamma) \coloneqq (\beta_1, \beta_2, \cdots, \beta_{|D|}),$$

such that for $\mathsf{swap}(\frac{u}{v}) := \frac{v}{u}$,

$$\beta_i = \begin{cases} \mathsf{swap}(\alpha_i) & \text{if } i \in T, \\ \alpha_i & \text{otherwise.} \end{cases}$$

- *Permuting* (perm, π)*: This action is applying a permutation* $\pi \in S_{|D|}$ *to a sequence of cards and transforms* $\Gamma = (\alpha_1, \alpha_2, \ldots, \alpha_{|D|})$ *as follows:*

$$\mathsf{perm}_\pi(\Gamma) \coloneqq \left(\alpha_{\pi^{-1}(1)}, \alpha_{\pi^{-1}(2)}, \cdots, \alpha_{\pi^{-1}(|D|)}\right).$$

- *Shuffling* $(\mathsf{shuf}, \Pi, \mathcal{F})$*: This action is applying a permutation chosen from a permutation set* $\Pi \subseteq S_{|D|}$ *according to a probability distribution* $\mathcal{F}$ *on* Π*. This action transforms* $\Gamma = (\alpha_1, \alpha_2, \cdots, \alpha_{|D|})$ *as follows:*

$$\mathsf{shuf}_{\Pi,\mathcal{F}}(\Gamma) \coloneqq \mathsf{perm}_\pi(\Gamma),$$

where $\pi \in \Pi$ *is drawn according to* $\mathcal{F}$*.*

A protocol $P = (D, U, Q, A)$ runs as an abstract machine: Staring from the initial state q_0 with some input $\Gamma_0 \in U$, the state and the current sequence change according to the output of the action function. When its state becomes q_{accept} or q_{reject}, the protocol terminates. Considering an execution of the protocol, the tuple of all sequences $(\Gamma_0, \Gamma_1, \cdots, \Gamma_t)$ appeared from the initial state q_0 to the final is called a *sequence trace*. Similarly, $(\mathrm{top}(\Gamma_0), \mathrm{top}(\Gamma_1), \cdots, \mathrm{top}(\Gamma_t))$ is called a *visible sequence trace*.

5.4 Properties of ZKP

Based on the formalization thus far, we formally define a card-based ZKP protocol (collection) that is non-interactive and has no soundness error, as follows.

Definition 2. *Let* L *be a language, and let* $x \in L \cup \breve{L}$*. We say that a protocol* $P_x = (D, U, Q, A)$ *is* compatible *with the instance* x *if its input set* U *contains every possible sequence whose prefix is a witness subsequence (corresponding to a witness* $w \in W_x$*).*[5]

Definition 3. *Let* L *be a language. Assume that, for every instance* $x \in L \cup \breve{L}$*, we have a protocol* P_x *compatible with* x*. We call the set of all these protocols a* ZKP protocol collection *for* L *if the following three conditions are met:*

Completeness *If* $x \in L$ *and an initial sequence* $\Gamma_0 \in U$ *for the protocol* P_x *contains a correct witness subsequence (corresponding to a witness* $w \in W_x$*), the protocol starting with* Γ_0 *always terminates with the accepting state* q_{accept}*.*

Soundness *If* $x \in L$ *and an initial sequence* Γ_0 *for* P_x *does not contain any correct witness subsequence, the protocol starting with* Γ_0 *always terminates with the rejecting state* q_{reject}*. If* $x \in \breve{L}$*,* P_x *always terminates with* q_{reject}*.*

Zero-knowledge *Let* $x \in L$*, and consider any distribution on input set* U *of the protocol* P_x*. For any run of the protocol, the distribution of input and that of the visible sequence trace are stochastically independent.*

[5] All other sequences in U start with illegal witness subsequences.

6 Proof of ZKP Properties for Our Protocols

In this section, we prove the completeness, soundness, and zero-knowledge of our proposed protocols based on the formalization presented in Section 5.

6.1 3-coloring Problem

We prove that our ZKP protocol for the 3-coloring problem presented in Section 3 satisfies the three conditions.

Theorem 1 (Completeness) *If the input sequence Γ_0 corresponding to ϕ contains a correct witness subsequence, the protocol always terminates with the accepting state q_{accept}.*

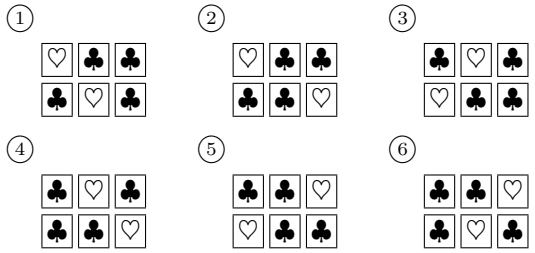
Proof. In Step 2(a), we apply a pile-scramble shuffle horizontally to the three piles. Let $\pi \in S_3$ be a random permutation generated by this pile-scramble shuffle. Two sequences that represented $\phi(i)$ and $\phi(j)$ before applying the pile-scramble shuffle become to represent $\pi\phi(i)$ and $\pi\phi(j)$, i.e., the positions of the red cards in the two sequences are $\pi\phi(i)$-th and $\pi\phi(j)$-th, respectively. By revealing the two sequences, we can know whether $\pi\phi(i) = \pi\phi(j)$ (i.e., $\phi(i) = \phi(j)$) or not. Therefore, the protocol always terminates with the accept state q_{accept}. □

Theorem 2 (Soundness) *If the input sequence Γ_0 corresponding to ϕ does not contain any correct witness subsequence, the protocol always terminates with the rejecting state q_{reject}.*

Proof. We consider the case where a sequence placed in Step 1 does not contain a correct witness subsequence. In Step 2(a), we make vertical piles by pile-scramble shuffle. That is, the sequence is placed in such a way that $\psi(u) = \psi(v)$ satisfied in Step 1. When the turn operation (turn, T) is performed in Step 2(b), it is found that $\boxed{\heartsuit}$ is in the same column, resulting in the rejecting state. □

Theorem 3 (Zero-knowledge) *For any run of the protocol, the distribution of input and that of the visible sequence trace are stochastically independent.*

Proof. In Step 2(b), the sequence to be turned over by the operation (turn, T) is randomly selected from the following six patterns:



Let $r \in S_3$ be a uniformly randomly generated permutation. This sequence is transformed by (perm, r). Thus, the visible sequence trace of the protocol is uniformly distributed. Therefore, the distribution of input and the visible sequence trace are stochastically independent. □

6.2 Graph Isomorphism Problem

We prove that our ZKP protocol for the graph isomorphism problem presented in Section 4 satisfies the three conditions.

Theorem 4 (Completeness) *If the input sequence Γ_0 corresponding to π contains a correct witness subsequence, the protocol always terminates with the accepting state q_{accept}.*

Proof. As shown in Section 4.1, there exists a permutation matrix P_π for the adjacency matrices $A(G_1)$ and $A(G_2)$ of two isomorphic graphs G_1 and G_2 as follows:

$$A(G_2) = P_\pi^T A(G_1) P_\pi.$$

In Step 4, $(\mathsf{perm}, (r\pi^{-1})^{-1})$ is equal to computing $P_\pi^{\mathrm{T}} A(G_1)$, and in Step 9, $(\mathsf{perm}, (r''\pi^{-1})^{-1})$ is equal to computing $(P_\pi^{\mathrm{T}} A(G_1))P_\pi$.

Therefore, the protocol always terminates with q_{accept} in Step 10. □

Theorem 5 (Soundness) *If the input sequence Γ_0 corresponding to ϕ does not contains any correct witness subsequence, the protocol always terminates with the rejecting state q_{reject}.*

Proof. Consider the case where a witness subsequence placed in Step 1 is not correct but legal. Let π' be a permutation corresponding to that witness subsequence. The card sequences in Steps 3, 4, 8, and 9 is transformed by (perm, π'). As shown in Section 4.1, if the permutation matrix $P_{\pi'}$ corresponding to π is used, it is transformed into a sequence corresponding to the graph G_2' instead of the graph G_2 such that

$$A(G_2') = P_{\pi'}^{\mathrm{T}} A(G_1) P_{\pi'}.$$

Thus, when (turn, T) is performed in Step 10, the sequence is different from that of the adjacency matrix in G_2, resulting in the rejecting state. Next, consider the case where a sequence corresponding to an illegal witness subsequence is placed. In this case, when (turn, T) is performed in Step 10, it is found that the sequence does not follow the format of the sequence, resulting in the rejecting state as well. □

Theorem 6 (Zero-knowledge) *For any run of the protocol, the distribution of input and that of the visible sequence trace are stochastically independent.*

Proof. As seen in Section 2.2, pile-scramble shuffles are applied so that random permutations r, r', r'' are generated. The sequence in Steps 3, 5, and 8 is transformed by (perm,r), (perm,r'), and (perm,r''), respectively. Thus, the visible sequence trace of the protocol is uniformly distributed. Therefore, the distribution of input and the visible sequence trace are stochastically independent. □

7 Conclusion

In this paper, we proposed physical ZKP protocols using a deck of cards for the two major graph problems. Our protocols have no soundness error and they are easy to implement. In particular, it is interesting to note that our ZKP protocol for the graph isomorphism problem requires only three shuffles. Similar to the proposed protocol, we believe that we can propose a card-based ZKP with no soundness error for other graph problems. In addition, we constructed a rigorous definition of a card-based ZKP protocol that is non-interactive and with no soundness error.

As future work, we are interested in the subgraph isomorphism problem[6] and in analyzing computation classes in more details. Furthermore, formalizing interactive card-based ZKP protocols (e.g., [2,5,16,23]) is an important future task. In addition, investigating the relationship between our model and the standard definitions of ZKP in details will be expected.

Acknowledgements

We thank the anonymous referees, whose comments have helped us improve the presentation of the paper. We would like to thank Hideaki Sone for his cooperation in preparing a Japanese draft version at an earlier stage of this work. We would also like to thank Kazumasa Shinagawa for his idea improving a protocol for the 3-coloring problem. The first author is grateful to Haruka Mizuta for helpful discussions at the beginning of this work. This work was supported in part by JSPS KAKENHI Grant Numbers JP19J21153 and JP21K11881.

References

1. Babai, L.: Graph isomorphism in quasipolynomial time. In: ACM Symposium on Theory of Computing. pp. 684–697. STOC '16, ACM, New York (2016), https://doi.org/10.1145/2897518.2897542
2. Bultel, X., Dreier, J., Dumas, J.G., Lafourcade, P.: Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen. In: Demaine, E.D., Grandoni, F. (eds.) Fun with Algorithms. LIPIcs, vol. 49, pp. 8:1–8:20. Schloss Dagstuhl, Dagstuhl (2016), https://doi.org/10.4230/LIPIcs.FUN.2016.8
3. Bultel, X., Dreier, J., Dumas, J., Lafourcade, P., Miyahara, D., Mizuki, T., Nagao, A., Sasaki, T., Shinagawa, K., Sone, H.: Physical zero-knowledge proof for Makaro. In: Izumi, T., Kuznetsov, P. (eds.) Stabilization, Safety, and Security of Distributed Systems. LNCS, vol. 11201, pp. 111–125. Springer, Cham (2018), https://doi.org/10.1007/978-3-030-03232-6_8
4. Crépeau, C., Kilian, J.: Discreet solitary games. In: Stinson, D.R. (ed.) Advances in Cryptology—CRYPTO' 93. LNCS, vol. 773, pp. 319–330. Springer, Berlin, Heidelberg (1994), https://doi.org/10.1007/3-540-48329-2_27

[6] Very recently, Ruangwises and Itoh [25,28] implied that a card-based ZKP protocol for the Hamiltonian cycle problem can be constructed in a similar way to our protocol for the graph isomorphism problem.

5. Dumas, J.G., Lafourcade, P., Miyahara, D., Mizuki, T., Sasaki, T., Sone, H.: Interactive physical zero-knowledge proof for Norinori. In: Du, D.Z., Duan, Z., Tian, C. (eds.) Computing and Combinatorics. LNCS, vol. 11653, pp. 166–177. Springer, Cham (2019), https://doi.org/10.1007/978-3-030-26176-4_14
6. Garey, M., Johnson, D., Stockmeyer, L.: Some simplified NP-complete graph problems. Theor. Comput. Sci. **1**(3), 237–267 (1976), https://doi.org/10.1016/0304-3975(76)90059-1
7. Godsil, C., Royle, G.F.: Algebraic Graph Theory, Graduate Texts in Mathematics, vol. 207. Springer (2001), https://doi.org/10.1007/978-1-4613-0163-9
8. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. J. ACM **38**(3), 690–728 (1991), https://doi.acm.org/10.1145/116825.116852
9. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989), https://doi.org/10.1137/0218012
10. Gradwohl, R., Naor, M., Pinkas, B., Rothblum, G.N.: Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. Theory Comput. Syst. **44**(2), 245–268 (2009), https://doi.org/10.1007/s00224-008-9119-9
11. Grohe, M., Schweitzer, P.: The graph isomorphism problem. Commun. ACM **63**(11), 128–134 (2020), https://doi.org/10.1145/3372123
12. Ishikawa, R., Chida, E., Mizuki, T.: Efficient card-based protocols for generating a hidden random permutation without fixed points. In: Calude, C.S., Dinneen, M.J. (eds.) Unconventional Computation and Natural Computation. LNCS, vol. 9252, pp. 215–226. Springer, Cham (2015), https://doi.org/10.1007/978-3-319-21819-9_16
13. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (eds.) Complexity of Computer Computations. pp. 85–103. Springer, Boston, MA (1972), https://doi.org/10.1007/978-1-4684-2001-2_9
14. Koch, A.: Cryptographic Protocols from Physical Assumptions. Ph.D. thesis, Karlsruhe Institute of Technology (KIT) (2019), https://doi.org/10.5445/IR/1000097756
15. Koch, A., Schrempp, M., Kirsten, M.: Card-based cryptography meets formal verification. New Gener. Comput. **39**(1), 115–158 (2021), https://doi.org/10.1007/s00354-020-00120-0
16. Lafourcade, P., Miyahara, D., Mizuki, T., Robert, L., Sasaki, T., Sone, H.: How to construct physical zero-knowledge proofs for puzzles with a "single loop" condition. Theor. Comput. Sci. (2021), https://doi.org/10.1016/j.tcs.2021.07.019, in press
17. Miyahara, D., Robert, L., Lafourcade, P., Takeshige, S., Mizuki, T., Shinagawa, K., Nagao, A., Sone, H.: Card-based ZKP protocols for Takuzu and Juosan. In: Farach-Colton, M., Prencipe, G., Uehara, R. (eds.) Fun with Algorithms. LIPIcs, vol. 157, pp. 20:1–20:21. Schloss Dagstuhl, Dagstuhl (2020), https://doi.org/10.4230/LIPIcs.FUN.2021.20
18. Mizuki, T., Shizuya, H.: A formalization of card-based cryptographic protocols via abstract machine. Int. J. Inf. Secur. **13**(1), 15–23 (2014), https://doi.org/10.1007/s10207-013-0219-4
19. Mizuki, T., Sone, H.: Six-card secure AND and four-card secure XOR. In: Deng, X., Hopcroft, J.E., Xue, J. (eds.) Frontiers in Algorithmics. LNCS, vol. 5598, pp. 358–369. Springer, Berlin, Heidelberg (2009), https://doi.org/10.1007/978-3-642-02270-8_36

20. Niemi, V., Renvall, A.: Secure multiparty computations without computers. Theor. Comput. Sci. **191**(1–2), 173–183 (1998), https://doi.org/10.1016/S0304-3975(97)00107-2
21. Ono, H., Manabe, Y.: Card-based cryptographic logical computations using private operations. New Gener. Comput. **39**(1), 19–40 (2021), https://doi.org/10.1007/s00354-020-00113-z
22. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Physical zero-knowledge proof for Suguru puzzle. In: Devismes, S., Mittal, N. (eds.) Stabilization, Safety, and Security of Distributed Systems. LNCS, vol. 11891, pp. 235–247. Springer, Cham (2020), https://doi.org/10.1007/978-3-030-64348-5_19
23. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Interactive physical ZKP for connectivity: Applications to Nurikabe and Hitori. In: De Mol, L., Weiermann, A., Manea, F., Fernández-Duque, D. (eds.) Connecting with Computability. LNCS, vol. 12813, pp. 373–384. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-80049-9_37
24. Ruangwises, S.: Two standard decks of playing cards are sufficient for a ZKP for Sudoku. In: Computing and Combinatorics. LNCS, Springer, Cham (2021), to appear
25. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for connected spanning subgraph problem and Bridges puzzle (2020), https://arxiv.org/abs/2011.02313
26. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for Numberlink puzzle and k vertex-disjoint paths problem. New Gener. Comput. **39**(1), 3–17 (2021), https://doi.org/10.1007/s00354-020-00114-y
27. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for Ripple Effect. In: Hong, S.H., Nandy, S.C., Uehara, R. (eds.) WALCOM: Algorithms and Computation. LNCS, vol. 12635, pp. 296–307. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-68211-8_24
28. Ruangwises, S., Itoh, T.: Physical ZKP for connected spanning subgraph: Applications to Bridges puzzle and other problems. In: Unconventional Computation and Natural Computation. LNCS, Springer, Cham (2021), to appear
29. Sasaki, T., Miyahara, D., Mizuki, T., Sone, H.: Efficient card-based zero-knowledge proof for Sudoku. Theor. Comput. Sci. **839**, 135–142 (2020), https://doi.org/10.1016/j.tcs.2020.05.036
30. Shinagawa, K.: On the Construction of Easy to Perform Card-Based Protocols. Ph.D. thesis, Tokyo Institute of Technology (2020)
31. Shinagawa, K., Mizuki, T., Schuldt, J.C.N., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Multi-party computation with small shuffle complexity using regular polygon cards. In: Au, M.H., Miyaji, A. (eds.) Provable Security. LNCS, vol. 9451, pp. 127–146. Springer, Cham (2015), https://doi.org/10.1007/978-3-319-26059-4_7
32. Ueda, I., Miyahara, D., Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: Secure implementations of a random bisection cut. Int. J. Inf. Secur. **19**(4), 445–452 (2020), https://doi.org/10.1007/s10207-019-00463-w