Card-Based Zero-Knowledge Proof Protocol for Pancake Sorting*

Yuichi Komano**[1] and Takaaki Mizuki[2]

[1] Toshiba Corporation, 1 Komukai-Toshiba-Cho, Saiwai-ku, Kawasaki, Japan
[2] Tohoku University, 6–3 Aramaki-Aza-Aoba, Aoba-ku, Sendai, Japan
mizuki+lncsweb[atmark]tohoku.ac.jp

Abstract. Assume that, given a sequence of n integers from 1 to n arranged in random order, we want to sort them, provided that the only acceptable operation is a prefix reversal, which means to take any number of integers (sub-sequence) from the left of the sequence, reverse the order of the sub-sequence, and return them to the original sequence. This problem is called "pancake sorting," and sorting an arbitrary sequence with the minimum number of operations restricted in this way is known to be NP-hard. In this paper, we consider applying the concept of zero-knowledge proofs to the pancake sorting problem. That is, we design a physical zero-knowledge proof protocol in which a user (the prover) who knows how to sort a given sequence with ℓ operations can convince another user (the verifier) that the prover knows this information without divulging it.

Keywords: Zero-knowledge proof, Card-based cryptography, Pancake sorting

1 Introduction

"Pancake sorting" [27] is a problem of sorting a given sequence of n integers from 1 to n by using only "prefix reversals," which rearrange a sub-sequence of any length taken from the left in the reverse order. In this paper, we apply the concept of zero-knowledge proofs to the pancake sorting problem and propose a physical zero-knowledge proof protocol for the pancake sorting problem. This paper begins by explaining the pancake sorting problem in detail.

** Presently, the author is with Chiba Institute of Technology, Japan.

* This paper appears in Proceedings of SecITC 2022. This version of the contribution has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-031-32636-3_13. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use: https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms.

1.1 Pancake Sorting Problem

We take a sequence of five integers $(3, 5, 2, 1, 4)$ as an example. For this sequence, let us reverse its prefixes of lengths 2, 5, 4, and 3 in this order one by one, so that the sequence is rearranged as

$$(\underline{3, 5}, 2, 1, 4) \to (\underline{5, 3, 2, 1, 4}) \to (\underline{4, 1, 2, 3}, 5) \to (\underline{3, 2, 1}, 4, 5) \to (1, 2, 3, 4, 5).$$

Thus, the sorting (in ascending order) is completed in four *prefix reversals*. The four prefix reversals above can be represented as a sequence $(2, 5, 4, 3)$ that consists of the lengths in the prefix reversals; such a sequence of prefix reversal lengths completing the sorting is called a *solution* to a given sequence (to be sorted).

This kind of sorting problem is called *pancake sorting* [27]; the name comes from the problem of sorting a stack of pancakes of distinct diameters in order of diameter size by repeatedly flipping over a number of pancakes at the top with a spatula.

Let us formalize this problem. Let $n \geq 1$, and let $(x_1, x_2, \ldots, x_n)$ be an input sequence that consists of n integers randomly arranged from 1 to n. Such a sequence of n integers can be regarded as a permutation on $\{1, 2, \ldots, n\}$. That is, when S_n denotes the symmetric group of degree n, the sequence $(x_1, x_2, \ldots, x_n)$ can be represented by the following permutation $x \in S_n$:

$$x = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ x_1 & x_2 & x_3 & \cdots & x_n \end{pmatrix}.$$

Next, let us also express the prefix reversal operations in terms of permutation as follows. For each i such that $1 \leq i \leq n$, the operation of a prefix reversal of length i is represented by the following permutation $\mathsf{sw}_i \in S_n$:

$$\mathsf{sw}_i = \begin{pmatrix} 1 & 2 & 3 & \cdots & i-2 & i-1 & i \\ i & i-1 & i-2 & \cdots & 3 & 2 & 1 \end{pmatrix}.$$

Thus, $(y_1, y_2, \ldots, y_\ell) \in \{1, 2, \ldots, n\}^\ell$ is a solution to a sequence $x \in S_n$ if and only if

$$\mathsf{sw}_{y_\ell} \circ \mathsf{sw}_{y_{\ell-1}} \circ \cdots \circ \mathsf{sw}_{y_1} \circ x = \mathsf{id}$$

holds, where $\mathsf{id} \in S_n$ denotes the identity. Note that the above permutation sw_i is equal to its inverse sw_i^{-1}. That is, $\mathsf{sw}_i = \mathsf{sw}_i^{-1}$ holds for every i, $1 \leq i \leq n$. Note furthermore that $\mathsf{sw}_1 = \mathsf{id}$.

Following this formulation, we can check, for example, that the above sequence $(3, 5, 2, 1, 4)$ and its solution $(2, 5, 4, 3)$ satisfy

$$\mathsf{sw}_3 \circ \mathsf{sw}_4 \circ \mathsf{sw}_5 \circ \mathsf{sw}_2 \circ \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 2 & 1 & 4 \end{pmatrix} = \mathsf{id}.$$

.

Table 1. Values of $h(n)$

n		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$h(n)$	(OEIS A058986)	0	1	3	4	5	7	8	9	10	11	13	14	15	16	17	18	19	20	22

1.2 Computational Complexity of Pancake Sorting

Since the pancake sorting problem was introduced in the 1970s, many researchers have reported algorithms for minimizing the number of prefix reversals and their lower bounds.

Among all the solutions to a sequence $x \in S_n$, any solution with the minimum number of prefix reversals is said to be *optimal*; we denote the minimum number of prefix reversals as $\alpha(x)$, that is, the length of any optimal solution to x. For example, because the length of the solution $(2, 5, 4, 3)$ to the sequence $(3, 5, 2, 1, 4)$ is four and there is no solution whose length is smaller than four, it is an optimal solution and we have $\alpha((3, 5, 2, 1, 4)) = 4$[3]. We also write $h(n)$ as the longest length of optimal solutions to sequences of n integers. That is, we define

$$h(n) := \max\{\alpha(x) \mid x \in S_n\}$$

for every $n \geq 1$.

As shown in Table 1, the values of $h(n)$ have been obtained up to $n = 19$ by numerical calculations or observations (e.g., $[2, 9, 10, 19, 30]$), and they are registered in the On-Line Encyclopedia of Integer Sequences (OEIS) as OEIS A058986[4]. Finding the values of $h(n)$ for $n \geq 20$ is an open problem.

As general upper and lower bounds on $h(n)$, Gates and Papadimitriou [12] showed that $\frac{17}{16}n \leq h(n) \leq \frac{5n+5}{3}$ in 1979. Since then, the bounds on $h(n)$ have been analyzed, and the best lower and upper bounds currently known are $\frac{15}{14}n \leq h(n)$ [19] and $h(n) \leq \frac{18}{11}n$ [8], respectively.

In contrast, the complexity of finding its optimal solution given an arbitrary sequence $x \in S_n$ (sorting by prefix reversals, or MIN-SBPR) had been unsolved for many years, until in 2012 Bulteau, Fertin, and Rusu [3, 4] proved that this problem, MIN-SBPR, is NP-hard.

1.3 Contribution

As explained in Sect. 1.2, MIN-SBPR is an NP-hard problem, and hence, there are possible situations where it is valuable to be the only one who knows a solution to a particular sequence of a pancake sorting problem. Therefore, we will attempt to apply the concept of the *zero-knowledge proof* [13] to the pancake sorting problem.

[3] In this way, we sometimes use the terms "sequence" and "permutation" interchangeably.

[4] `https://oeis.org/A058986`

Suppose that there are two users, a prover P and a verifier V, and only the prover P knows a solution $(y_1, y_2, \ldots, y_\ell)$ of length ℓ to a sequence $x \in S_n$. Assume furthermore that P wants to convince V that P knows the solution without leaking any information about the solution. Our contribution is to propose a zero-knowledge proof protocol for the pancake sorting problem that achieves this goal. The proposed protocol is a so-called *physical* zero-knowledge proof protocol that can be executed using a physical deck of cards.

The following is an example of a game where the proposed protocol will be useful. When a sequence $(12, 19, 20, 4, 13, 17, 5, 10, 16, 15, 11, 1, 7, 14, 3, 6, 18, 8, 2, 9)$ of size $n = 20$ of the pancake sorting problem is given, multiple players try to find a solution to it, and a player wins the game if he/she finds the solution with the shortest length. In this case, if a player discloses the solution to other players, another player who sees the solution may take it as his/her own achievement or may use it as a hint for finding a solution to the next game, which makes the game less fun. Therefore, a player who has found a solution of certain length first convinces other players that he/she knows the solution without leaking any information using our zero-knowledge proof protocol, and then the player discloses the solution after gaining sufficient recognition so that he/she can correctly claim the achievements or be judged as a winner of the game.

In addition, we expect that our proposed protocol can be used as a good educational tool for teaching lay-people the concept of zero-knowledge proof as well as the sorting problem. Furthermore, as will be explained in Sect. 6, our proposed technique can be applied to more general problems (beyond the pancake sorting problem).

1.4 Related Work

One problem similar to the pancake sorting problem is *Topswops* [25, 26, 45]. In Topswops, for a sequence of integers, a prefix reversal of the first k integers, where k is the leading integer of the sequence, is repeated until the leading integer becomes 1. Recently, the authors constructed a physical zero-knowledge proof protocol that can verify that a Topswops game terminates with a predefined number of prefix reversals while the input sequence of integers is kept confidential [29]. Although pancake sorting is similar to Topswops, what is being kept secret in the zero-knowledge proofs is different. In the former, the solution should be kept secret, whereas in the latter, the input sequence of integers should be kept secret.

In addition, numerous physical zero-knowledge proof protocols for pencil puzzles have been constructed to date using a physical deck of cards. Examples are Akari [5], Cryptarithmetic [22], Hashiwokakero (Bridges) [63], Heyawake [49], Hitori [49,52], Juosan [39], Kakuro [5,40], KenKen [5], Makaro [6,66], Masyu [32], Nonogram [7,54], Norinori [11], Numberlink [58,60], Nurikabe [49,52], Nurimisaki [50], Ripple Effect [61,62], Shikaku [65], Slitherlink [32,33], Sudoku [14,55, 57,67,68], Suguru [48,51], Takuzu [5,39], and Usowan [53].

Card-based cryptography that performs cryptographic tasks using a deck of physical cards has been growing rapidly in recent years [42,43]. Hot topics include secure and efficient protocols in the private model [1,35,46], multi-valued-output symmetric function evaluation [64, 71], information leakage due to operative errors [44], graph automorphism shuffles [41], secure sorting [16], multi-valued protocols with a direction encoding [76], the half-open action [38], card-minimal protocols [15,28], and single-shuffle protocols [31,73]. Furthermore, very recently, Shinagawa and Nuida [74] showed that a certain single-shuffle protocol implies the existence of a private simultaneous messages protocol; this is the first successful and amazing result that directly connects 'physical' protocols with 'digital' protocols. It should be noted that several studies [34, 56, 75] on card-based cryptography were reported in the previous SecITC conferences.

2 Preliminaries

In this section, we first explain the physical properties of cards used in this paper, then describe how to encode permutations and integers using cards, and finally introduce the "pile-scramble shuffle" [21], which will be used in our protocol. Hereinafter, n denotes the size of a pancake sorting problem (i.e., the length of an input sequence of integers).

2.1 Physical Cards

In this paper, two types of physical cards are used:

Integer cards Each card has an integer from 1 to n written on its face, such as $\boxed{1}\boxed{2}\boxed{3}\cdots\boxed{n}$, and the reverse side of every card has the same pattern $\boxed{?}$.

Black and red cards Each card has a ♣ or ♡ symbol on its face, and the back of every card has the same pattern $\boxed{?}$.

We use the notation

$$\underset{i}{\boxed{?}}$$

to denote a face-down integer card whose face is $\boxed{i}$ for an integer $1 \leq i \leq n$.

2.2 Permutation Commitment

As explained in Sect. 1.1, a sequence of integers and an operation of prefix reversal in the pancake sorting problem are represented by permutations. Therefore, we introduce a method to represent permutations with integer cards, as often used in card-based cryptography [20, 67].

To represent a permutation $\pi \in S_n$, we simply use n integer cards $\boxed{1}\boxed{2}\boxed{3}\cdots\boxed{n}$ and arrange them according to the values of $\pi(1), \pi(2), \ldots, \pi(n)$:

$$\boxed{\pi(1)}\boxed{\pi(2)}\boxed{\pi(3)}\cdots\boxed{\pi(n)}.$$

Consider turning over these n cards: we call n face-down cards

$$\underset{\pi(1)}{?}\ \underset{\pi(2)}{?}\ \underset{\pi(3)}{?}\ \cdots\ \underset{\pi(n)}{?}$$

a *permutation commitment* to $\pi \in S_n$. For example, a permutation commitment to $\mathsf{sw}_3 \in S_n$ (corresponding to a prefix reversal of length 3) is

$$\underset{3}{?}\underset{2}{?}\underset{1}{?}\underset{4}{?}\underset{5}{?}\cdots\ \underset{n-1}{?}\underset{n}{?}.$$

In the following, we write a permutation commitment to $\pi \in S_n$ as

$$\pi : ?\,?\,\cdots\,? \qquad \text{or} \qquad \underbrace{?\,\|}_{\pi}.$$

2.3 Integer Commitment

As explained in Sect. 2.2, a sequence $x \in S_n$ and a prefix reversal sw_i are represented with permutation commitments. Another important element of the pancake problem is a 'solution,' and hence, we introduce "integer commitments" here to express the solution with cards.

With $n-1$ black cards and one red card, let us encode integers from 1 to n as

$$\begin{aligned}
&\heartsuit\,\clubsuit\,\clubsuit\,\clubsuit\,\cdots\,\clubsuit\,\clubsuit = 1\\
&\clubsuit\,\heartsuit\,\clubsuit\,\clubsuit\,\cdots\,\clubsuit\,\clubsuit = 2\\
&\clubsuit\,\clubsuit\,\heartsuit\,\clubsuit\,\cdots\,\clubsuit\,\clubsuit = 3\\
&\qquad\qquad\qquad\vdots\\
&\clubsuit\,\clubsuit\,\clubsuit\,\clubsuit\,\cdots\,\clubsuit\,\heartsuit = n.
\end{aligned}$$

That is, the position of the red card $\heartsuit$ determines the integer. Following this encoding rule, we will call a sequence of face-down cards an *integer commitment*. Such an encoding rule is often used in card-based cryptography [37, 59, 77].

According to convention, we write an integer commitment to i, $1 \le i \le n$, as the symbol $E_n(i)$:

$$E_n(i) : ?\,?\,?\,\cdots\,?,$$

where only the i-th card is $\heartsuit$ and the remaining $n-1$ cards are $\clubsuit$ as mentioned above.

2.4 Pile-scramble Shuffle and Composition of Permutations

A *pile-scramble shuffle* [21] is a shuffling operation by which several piles of cards of the same size are shuffled.

As an example, suppose that we have three permutation commitments to a sequence $x \in S_n$, the identity $\mathsf{id} \in S_n$, and a prefix reversal $\mathsf{sw}_i \in S_n$:

$$\begin{aligned} x &: \boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?} \\ \mathsf{id} &: \boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?} \\ \mathsf{sw}_i &: \boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}. \end{aligned}$$

Considering each (vertical) column consisting of three cards as a single pile, we apply a pile-scramble shuffle to the n piles; then, the transition is as follows:

$$\left[\begin{array}{c|c|c|c|c} \boxed{?} & \boxed{?} & \boxed{?} & \cdots & \boxed{?} \\ \boxed{?} & \boxed{?} & \boxed{?} & \cdots & \boxed{?} \\ \boxed{?} & \boxed{?} & \boxed{?} & \cdots & \boxed{?} \end{array}\right] \quad \rightarrow \quad \begin{aligned} r \circ x &: \boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?} \\ r \circ \mathsf{id} &: \boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?} \\ r \circ \mathsf{sw}_i &: \boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}, \end{aligned}$$

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

We then turn over the bottom row, namely the permutation commitment to $r \circ \mathsf{sw}_i$, and sort the vertical columns without collapsing them based on the n integers appearing in the bottom row. With this sort, $(r \circ \mathsf{sw}_i)^{-1}$ acts upon the top two rows, and the cards are rearranged as follows (note that $\mathsf{sw}_i = \mathsf{sw}_i^{-1}$ holds):

$$\begin{aligned} \mathsf{sw}_i \circ x &: \boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?} \\ \mathsf{sw}_i &: \boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?} \\ &\quad \boxed{1}\boxed{2}\boxed{3}\cdots\boxed{n}. \end{aligned}$$

Thus, the above series of operations allows us to compose permutations of the prefix reversal sw_i and of the sequence x, while the permutation commitment to sw_i remains intact. The proposed protocol in this paper uses this technique, which originally comes from the "permutation division protocol" developed by Hashimoto et al. [17, 18].

3 Proposed Protocol

In this section, we propose a physical zero-knowledge proof protocol for the pancake sorting problem using permutation and integer commitments.

Let $x \in S_n$ be an input sequence and let $y = (y_1, y_2, \ldots, y_\ell)$ be a solution to x with length ℓ. That is, $\mathsf{sw}_{y_\ell} \circ \mathsf{sw}_{y_{\ell-1}} \circ \cdots \circ \mathsf{sw}_{y_1} \circ x = \mathsf{id}$ holds. Assume that the sequence x and the length of the solution ℓ are public information and that only the prover P knows the solution y (i.e., the verifier V does not know y).

3.1 Concept

As seen in Sect. 2.4, from permutation commitments to x and sw_i, it is easy to construct a composition of permutations $\mathsf{sw}_i \circ x$. A permutation commitment

to the input sequence x can be created publicly. Thus, if the prover P prepares permutation commitments to $\mathsf{sw}_{y_1}, \mathsf{sw}_{y_2}, \ldots, \mathsf{sw}_{y_\ell}$ corresponding to the solution $y = (y_1, y_2, \ldots, y_\ell)$, then by composing them, we have

$$\mathsf{sw}_{y_\ell} \circ \mathsf{sw}_{y_{\ell-1}} \circ \cdots \circ \mathsf{sw}_{y_1} \circ x : \boxed{?}\boxed{?}\boxed{?} \cdots \boxed{?}.$$

By turning over this permutation commitment and checking that it is the identity id, we can guarantee that the prover P knows y. Based on these ideas, we propose our protocol as described below.

Note that if the prover P directly creates and places permutation commitments to $\mathsf{sw}_{y_1}, \ldots, \mathsf{sw}_{y_\ell}$ by himself/herself, then we cannot guarantee that they surely correspond to some prefix reversals; therefore, we need a more elaborate way to arrange permutation commitments to $\mathsf{sw}_{y_1}, \ldots, \mathsf{sw}_{y_\ell}$.

3.2 Protocol Description

First of all, as an input to our protocol, a prover P, who knows a solution $y = (y_1, y_2, \ldots, y_\ell)$, creates integer commitments $E_n(y_1), E_n(y_2), \ldots, E_n(y_\ell)$ corresponding to $(y_1, y_2, \ldots, y_\ell)$ in secret and places them on the table as follows:

$$\begin{aligned} E_n(y_1) &: \boxed{?}\boxed{?} \cdots \boxed{?} \\ E_n(y_2) &: \boxed{?}\boxed{?} \cdots \boxed{?} \\ &\vdots \\ E_n(y_\ell) &: \boxed{?}\boxed{?} \cdots \boxed{?}. \end{aligned} \tag{1}$$

In addition, $n+2$ sets of integer cards $\boxed{1}\boxed{2}\boxed{3}\cdots\boxed{n}$ as additional cards are prepared.

Our protocol is executed with the above cards as input. Because our protocol is non-interactive (cf. [36]), it may be executed individually by either the prover P or by the verifier V (or even by any third party).

Protocol 1 (Proposed protocol)

1. Using additional $n+2$ sets of integer cards, arrange $n+2$ permutation commitments to $\mathsf{sw}_1, \mathsf{sw}_2, \ldots, \mathsf{sw}_n$, x, and id, as follows:

$$\begin{aligned} &\underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_1}\underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_2} \cdots \underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_n} \\ x &: \boxed{?}\boxed{?}\boxed{?} \cdots \boxed{?} \\ \mathsf{id} &: \boxed{?}\boxed{?}\boxed{?} \cdots \boxed{?}. \end{aligned}$$

2. Take the permutation commitments to $\mathsf{sw}_1, \mathsf{sw}_2, \ldots, \mathsf{sw}_n$, and place them along with the integer commitments $E_n(y_1), E_n(y_2), \cdots, E_n(y_\ell)$ of Eq. (1)

as follows:

$$\begin{array}{rc} & \underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_1}\ \underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_2}\ \cdots\ \underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_n} \\ E_n(y_1): & \boxed{?}\ \boxed{?}\ \cdots\ \boxed{?} \\ E_n(y_2): & \boxed{?}\ \boxed{?}\ \cdots\ \boxed{?} \\ & \vdots \\ E_n(y_\ell): & \boxed{?}\ \boxed{?}\ \cdots\ \boxed{?}. \end{array}$$

Note that in the first row, the permutation commitment to sw_{y_i} for each i, $1 \le i \le \ell$, appears in the column where the card $\boxed{\heartsuit}$ appears in the $(i+1)$-th row[5].

3. Apply a pile-scramble shuffle[6]:

$$\left[\begin{array}{c|c|c|c} \boxed{?}\!\!\boxed{?} & \boxed{?}\!\!\boxed{?} & \cdots & \boxed{?}\!\!\boxed{?} \\ \boxed{?} & \boxed{?} & \cdots & \boxed{?} \\ \vdots & \vdots & & \vdots \\ \boxed{?} & \boxed{?} & \cdots & \boxed{?} \end{array}\right].$$

Because the pile-scramble shuffle does not change within each vertical column, the above statement "in the first row, the permutation commitment to sw_{y_i} appears in the column where the card $\boxed{\heartsuit}$ appears in the $(i+1)$-th row" remains valid. Therefore, we turn over the integer commitment in the second row, and identify the permutation commitment $\underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_{y_1}}$ above $\boxed{\heartsuit}$:

$$\begin{array}{c} \boxed{?}\!\!\boxed{?}\ \cdots\ \underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_{y_1}}\ \cdots\ \boxed{?}\!\!\boxed{?} \\ \boxed{\clubsuit}\!\!\boxed{\clubsuit}\ \cdots\ \boxed{\heartsuit}\ \cdots\ \boxed{\clubsuit}\!\!\boxed{\clubsuit} \\ \boxed{?}\!\!\boxed{?}\ \cdots\ \boxed{?}\ \cdots\ \boxed{?}\!\!\boxed{?} \\ \vdots\ \ \vdots\ \quad \vdots\ \quad \vdots \\ \boxed{?}\!\!\boxed{?}\ \cdots\ \boxed{?}\ \cdots\ \boxed{?}\!\!\boxed{?} \end{array}.$$

4. Use the permutation commitment to sw_{y_1} just identified and the permutation commitments to x and id to construct a composition of permutations $\mathsf{sw}_{y_1} \circ x$ (still holding a permutation commitment to sw_{y_1}), as described in Sect. 2.4:

$$\begin{array}{rc} x: & \boxed{?}\!\!\boxed{?}\ \cdots\ \boxed{?}\!\!\boxed{?} \\ \mathsf{id}: & \boxed{?}\!\!\boxed{?}\ \cdots\ \boxed{?}\!\!\boxed{?} \\ \mathsf{sw}_{y_1}: & \boxed{?}\!\!\boxed{?}\ \cdots\ \boxed{?}\!\!\boxed{?} \end{array} \quad \rightarrow \quad \left[\begin{array}{c|c|c|c} \boxed{?}\!\!\boxed{?} & \boxed{?}\!\!\boxed{?} & \cdots & \boxed{?}\!\!\boxed{?} \\ \boxed{?}\!\!\boxed{?} & \boxed{?}\!\!\boxed{?} & \cdots & \boxed{?}\!\!\boxed{?} \\ \boxed{?}\!\!\boxed{?} & \boxed{?}\!\!\boxed{?} & \cdots & \boxed{?}\!\!\boxed{?} \end{array}\right] \quad \rightarrow \quad \begin{array}{rc} \mathsf{sw}_{y_1} \circ x: & \boxed{?}\!\!\boxed{?}\ \cdots\ \boxed{?}\!\!\boxed{?} \\ \mathsf{sw}_{y_1}: & \boxed{?}\!\!\boxed{?}\ \cdots\ \boxed{?}\!\!\boxed{?} \\ & \boxed{1}\!\!\boxed{2}\ \cdots\ \boxed{n}. \end{array}$$

[5] Here, $E_n(y_1)$ is the second row, $E_n(y_2)$ is the third row, and so on.

[6] Instead of a pile-scramble shuffle, one may use a "pile-shifting shuffle" [47, 72].

5. Return $\underbrace{\boxed{?}\!\!\boxed{}}_{\mathsf{sw}_{y_1}}$ obtained in the previous step to its original position in Step (3) and remove the second row:

$$\begin{array}{ccccccc} \boxed{?}\!\!\boxed{} & \boxed{?}\!\!\boxed{} & \cdots & \underbrace{\boxed{?}\!\!\boxed{}}_{\mathsf{sw}_{y_1}} & \cdots & \boxed{?}\!\!\boxed{} \\ \boxed{?} & \boxed{?} & \cdots & \boxed{?} & \cdots & \boxed{?} \\ \vdots & \vdots & & \vdots & & \vdots \\ \boxed{?} & \boxed{?} & \cdots & \boxed{?} & \cdots & \boxed{?} \end{array}.$$

6. Repeat Steps (3) through (5) $\ell - 1$ times, but do not execute the process corresponding to Step (5) in the last iteration. That is, we identify the permutation commitments from sw_{y_2} to sw_{y_ℓ} and compose them sequentially to the composition of permutations $\mathsf{sw}_{y_1} \circ x$. Finally, we obtain the composition of permutations as follows:

$$\mathsf{sw}_{y_\ell} \circ \mathsf{sw}_{y_{\ell-1}} \circ \cdots \circ \mathsf{sw}_{y_1} \circ x : \boxed{?}\!\!\boxed{?}\!\!\boxed{?} \cdots \boxed{?}.$$

7. Turn over the permutation commitment to $\mathsf{sw}_{y_\ell} \circ \mathsf{sw}_{y_{\ell-1}} \circ \cdots \circ \mathsf{sw}_{y_1} \circ x$ obtained in the previous step and return "accept" if it is id; otherwise return "reject."

4 Security and Performance

In this section, we discuss the security and performance of Protocol 1 described in Sect. 3.

4.1 Security

First, let us check that, for a given sequence $x \in S_n$ and a length ℓ, our protocol performs a zero-knowledge proof for a solution y (of length ℓ); in other words, we need to show that the protocol satisfies the completeness, soundness, and zero-knowledge properties.

Completeness Suppose that the prover P correctly places integer commitments according to the solution y. In this case, as can be seen from the construction of our protocol, it is not rejected at any step and is also accepted at the final step.

Soundness Assume that the prover P places illegal commitments as input. There are two cases to be considered; we will show that our protocol eventually rejects the invalid solution in both cases.

(i) If there is an illegal integer commitment (placed by P) which does not consist of one red card and $n-1$ black cards, then it is detected and rejected when the integer commitment is turned over in Step (3) of our protocol.

(ii) If the input integer commitments correspond to an incorrect solution $y' = (y'_1, y'_2, \ldots, y'_\ell)$, then the permutation commitment to x is rearranged according to y', as can be seen from the construction of our protocol. However, the rearranged permutation is not id in Step (7) and our protocol rejects it.

Zero-knowledge Suppose that the integer commitments corresponding to the solution $y = (y_1, y_2, \ldots, y_\ell)$ are correctly placed. No information about y_i is leaked during the execution of our protocol, because a pile-scramble shuffle is applied immediately before the cards are turned over to be face-up (except in the final step). The identity id that is opened in the final step is public information. Therefore, our protocol is information-theoretically secure.

4.2 Performance

This subsection discusses the number of cards and the number of shuffles required in our protocol.

First, for the number of cards, as described at the beginning of Sect. 3.2, we use ℓ $\boxed{\heartsuit}$ cards and $(n-1)\ell$ $\boxed{\clubsuit}$ cards for integer commitments. In addition, we use $n+2$ sets of cards from $\boxed{1}$ to $\boxed{n}$ as additional cards. Thus, $n\ell$ black and red cards and $n(n+2)$ integer cards are used. Therefore, our protocol requires $n^2 + (2+\ell)n$ cards in total.

The only shuffling operation used in the proposed protocol is the pile-scramble shuffle. The pile-scramble shuffle is performed once in Step (3) and once in Step (4), and each of these steps is executed ℓ times. Therefore, in total, the number of shuffles required in our protocol is 2ℓ.

5 Variants

This section presents variants of our protocol.

5.1 Non-interactive Protocol with Fewer Additional Cards

In our protocol, the first integer commitment is

$$E_n(y_1) : \boxed{?}\boxed{?}\cdots\boxed{?},$$

where the y_1-th card is $\boxed{\heartsuit}$ and the other $n-1$ cards are $\boxed{\clubsuit}$. Let us consider a variant that uses $\boxed{1}$ instead of $\boxed{\heartsuit}$ and $\boxed{2}, \boxed{3}, \cdots, \boxed{n}$ instead of $\boxed{\clubsuit}$. In this case, after $\underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_{y_1}}$ is identified in Step (3) of our protocol, the opened $\boxed{1}, \boxed{2}, \boxed{3}, \cdots, \boxed{n}$ can be used as additional cards in Step (4), and hence, we can reduce the number of additional cards by n.

5.2 Interactive Protocol with Fewer Additional Cards

Our protocol uses ℓ $\boxed{\heartsuit}$ cards and $(n-1)\ell$ $\boxed{\clubsuit}$ cards for integer commitments $E_n(y_1), E_n(y_2), \cdots, E_n(y_\ell)$. Instead of preparing all $E_n(y_1), E_n(y_2), \cdots, E_n(y_\ell)$ in advance, if the prover P places $E_n(y_i)$ every time Steps (3) and (6) are executed, the protocol could be executed with one $\boxed{\heartsuit}$ card and $n-1$ $\boxed{\clubsuit}$ cards.

Note that the order of permutation commitments $\underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_{y_1}}\underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_{y_2}}\cdots\underbrace{\boxed{?}\!\!\boxed{?}}_{\mathsf{sw}_{y_n}}$ is randomized by the pile-scramble shuffle in Step (3). Hence, the prover can not determine the correct position of the permutation commitment corresponding to the next prefix reversal, and hence, it is impossible for the prover to place a new integer commitment correctly. To address this issue, just after returning the permutation commitment to its original position in Step (5), the prover applies a pile-scramble shuffle, turns over the permutation commitments to be face-up, places a new integer commitment corresponding to the next prefix reversal, and turns over the permutation commitments again to be face-down. After that, the piles of cards are shuffled with a pile-scramble shuffle in Step (3), as invoked by Step (6). Hence, although this variant can reduce the number of required cards, the number of shuffles increases by $\ell - 1$.

Similar to Sect. 5.1, the number of additional cards can be reduced more by using $\boxed{1}\boxed{2}\boxed{3}\cdots\boxed{n}$ instead of one $\boxed{\heartsuit}$ card and $n-1$ $\boxed{\clubsuit}$ cards.

5.3 Protocol with Fewer Shuffles

Because sw_i satisfies $\mathsf{sw}_i = \mathsf{sw}_i^{-1}$ (namely, $\mathsf{sw}_i \circ \mathsf{sw}_i = \mathsf{sw}_i^{-1} \circ \mathsf{sw}_i = \mathsf{id}$), two consecutive applications of sw_i to a sequence $x \in S_n$ will not change it. If a player tries to find a solution with a shorter length for the pancake sorting problem, the same prefix reversal is never performed twice in a row. Instead of executing Step (3) of our protocol (and the step that is equivalent to Step (3) in Step (6)) for $E_n(y_i)$ one at a time, executing two steps for $E_n(y_i)$ and $E_n(y_{i+1})$ together can reduce the number of shuffles by $\lfloor \ell/2 \rfloor$. This variant applies to both the interactive and non-interactive protocols above.

5.4 Protocol with Fewer Cards

The permutation commitment $\mathsf{sw}_1 = \mathsf{id}$ does not change a sequence $x \in S_n$. Similar to the discussion in Sect. 5.3, if a player tries to find a solution with a shorter length for the pancake sorting problem, $\mathsf{sw}_1 = \mathsf{id}$ is never performed in our protocol as well as the above-mentioned variants. Hence, we can omit the leftmost column in Step (2) of our protocol so that Steps (2), (3), and (5) are performed with $n-1$ piles (columns) of cards. In this variant, we can reduce the number of cards by $n + \ell$.

6 Conclusion

In this paper, we proposed a physical zero-knowledge proof protocol for the pancake sorting problem. The main idea is to combine permutation and integer commitments so that a prover can efficiently place a solution and efficiently perform prefix reversals secretly.

Because the pancake sorting problem has many variations (e.g., the introduction of settings where pancakes have two distinct sides [12, 69, 70]), building new or generic protocols for them is one of our future works.

Beyond the pancake sorting problem and its variants, our zero-knowledge proof protocol can be modified for the following general problem[7]: Assuming that a sequence x of length n (which is not necessarily a permutation) and an integer ℓ along with m distinct permutations $\sigma_1, \sigma_2, \ldots, \sigma_m \in S_n$ and a sequence z of length n are public, the prover wants to convince the verifier that the prover knows $(y_1, y_2, \ldots, y_\ell)$ such that

$$\sigma_{y_\ell} \circ \sigma_{y_{\ell-1}} \circ \cdots \circ \sigma_{y_1}(x) = z,$$

where $\pi(x)$ for a permutation π represents the permuted sequence according to π. This general problem includes solving the Rubik's Cube, for instance.

The graph obtained by connecting two vertices that can be transitioned by a prefix reversal with edges, where each sequence is regarded as a vertex, is called a *pancake network* [19], and is considered to be the origin of the *reconfiguration problem* (e.g., [23, 24]), which is currently popular in the study of algorithm theory. Our protocol can be regarded as a technique to show that a vertex can be transitioned from one vertex to another in a pancake network without leaking any information, and we believe that it is an attractive topic to investigate whether card-based cryptography and zero-knowledge proofs can be applied to various other reconfiguration problems.

Acknowledgements

We thank the anonymous referees, whose comments have helped us improve the presentation of the paper. This work was supported by Grant-in-Aid for Scientific Research (JP18H05289, JP21K11881). We thank Koji Nuida for advising us to generalize the problem, as described in the third paragraph of Sect. 6.

References

1. Yoshiki Abe, Takeshi Nakai, Yoshihisa Kuroki, Shinnosuke Suzuki, Yuta Koga, Yohei Watanabe, Mitsugu Iwamoto, and Kazuo Ohta. Efficient card-based majority voting protocols. *New Gener. Comput.*, 40:173–198, 2022. URL: `https://doi.org/10.1007/s00354-022-00161-7`.
2. Shogo Asai, Yuusuke Kounoike, Yuji Shinano, and Keiichi Kaneko. Computing the diameter of 17-pancake graph using a pc cluster. In Wolfgang E. Nagel, Wolfgang V. Walter, and Wolfgang Lehner, editors, *Euro-Par 2006 Parallel Processing*, pages 1114–1124, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
3. Laurent Bulteau, Guillaume Fertin, and Irena Rusu. Pancake flipping is hard. In Branislav Rovan, Vladimiro Sassone, and Peter Widmayer, editors, *Mathematical Foundations of Computer Science 2012*, pages 247–258, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
4. Laurent Bulteau, Guillaume Fertin, and Irena Rusu. Pancake flipping is hard. *Journal of Computer and System Sciences*, 81(8):1556–1574, 2015. URL: `https://www.sciencedirect.com/science/article/pii/S0022000015000124`, `doi:https://doi.org/10.1016/j.jcss.2015.02.003`.

[7] This generalization was pointed out by Koji Nuida.

5. Xavier Bultel, Jannik Dreier, Jean-Guillaume Dumas, and Pascal Lafourcade. Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen. In Erik D. Demaine and Fabrizio Grandoni, editors, *Fun with Algorithms*, volume 49 of *LIPIcs*, pages 8:1–8:20, Dagstuhl, Germany, 2016. Schloss Dagstuhl. URL: https://doi.org/10.4230/LIPIcs.FUN.2016.8.
6. Xavier Bultel, Jannik Dreier, Jean-Guillaume Dumas, Pascal Lafourcade, Daiki Miyahara, Takaaki Mizuki, Atsuki Nagao, Tatsuya Sasaki, Kazumasa Shinagawa, and Hideaki Sone. Physical zero-knowledge proof for Makaro. In *Stabilization, Safety, and Security of Distributed Systems*, volume 11201 of *LNCS*, pages 111–125, 2018. URL: https://doi.org/10.1007/978-3-030-03232-6_8.
7. Yu-Feng Chien and Wing-Kai Hon. Cryptographic and physical zero-knowledge proof: From Sudoku to Nonogram. In Paolo Boldi and Luisa Gargano, editors, *Fun with Algorithms*, volume 6099 of *LNCS*, pages 102–112, Berlin, Heidelberg, 2010. Springer. URL: https://doi.org/10.1007/978-3-642-13122-6_12.
8. B. Chitturi, W. Fahle, Z. Meng, L. Morales, C.O. Shields, I.H. Sudborough, and W. Voit. An (18/11)n upper bound for sorting by prefix reversals. *Theoretical Computer Science*, 410(36):3372–3390, 2009. Graphs, Games and Computation: Dedicated to Professor Burkhard Monien on the Occasion of his 65th Birthday. URL: https://www.sciencedirect.com/science/article/pii/S0304397508003575, doi:https://doi.org/10.1016/j.tcs.2008.04.045.
9. Josef Cibulka. On average and highest number of flips in pancake sorting. *Theoretical Computer Science*, 412(8):822–834, 2011. URL: https://www.sciencedirect.com/science/article/pii/S0304397510006663, doi:https://doi.org/10.1016/j.tcs.2010.11.028.
10. David S. Cohen and Manuel Blum. On the problem of sorting burnt pancakes. *Discrete Applied Mathematics*, 61(2):105–120, 1995. URL: https://www.sciencedirect.com/science/article/pii/0166218X94000093, doi:https://doi.org/10.1016/0166-218X(94)00009-3.
11. Jean-Guillaume Dumas, Pascal Lafourcade, Daiki Miyahara, Takaaki Mizuki, Tatsuya Sasaki, and Hideaki Sone. Interactive physical zero-knowledge proof for Norinori. In Ding-Zhu Du, Zhenhua Duan, and Cong Tian, editors, *Computing and Combinatorics*, volume 11653 of *LNCS*, pages 166–177, Cham, 2019. Springer. URL: https://doi.org/10.1007/978-3-030-26176-4_14.
12. William H. Gates and Christos H. Papadimitriou. Bounds for sorting by prefix reversal. *Discret. Math.*, 27(1):47–57, 1979. doi:10.1016/0012-365X(79)90068-2.
13. S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Annual ACM Symposium on Theory of Computing*, STOC'85, pages 291–304, New York, 1985. ACM. URL: https://doi.org/10.1145/22145.22178.
14. Ronen Gradwohl, Moni Naor, Benny Pinkas, and Guy N. Rothblum. Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. *Theory of Computing Systems*, 44(2):245–268, 2009. URL: https://doi.org/10.1007/s00224-008-9119-9.
15. Rikuo Haga, Yuichi Hayashi, Daiki Miyahara, and Takaaki Mizuki. Card-minimal protocols for three-input functions with standard playing cards. In Lejla Batina and Joan Daemen, editors, *Progress in Cryptology—AFRICACRYPT 2022*, volume 13503 of *LNCS*, pages 448–468, Cham, 2022. Springer.
16. Rikuo Haga, Kodai Toyoda, Yuto Shinoda, Daiki Miyahara, Kazumasa Shinagawa, Yuichi Hayashi, and Takaaki Mizuki. Card-based secure sorting protocol. In Chen-Mou Cheng and Mitsuaki Akiyama, editors, *Advances in Information and Com-*

puter Security, volume 13504 of *LNCS*, pages 224–240, Cham, 2022. Springer. URL: https://doi.org/10.1007/978-3-031-15255-9_12.

17. Yuji Hashimoto, Kazumasa Shinagawa, Koji Nuida, Masaki Inamura, and Goichiro Hanaoka. Secure grouping protocol using a deck of cards. In Junji Shikata, editor, *Information Theoretic Security*, volume 10681 of *LNCS*, pages 135–152, Cham, 2017. Springer. URL: https://doi.org/10.1007/978-3-319-72089-0_8.
18. Yuji Hashimoto, Kazumasa Shinagawa, Koji Nuida, Masaki Inamura, and Goichiro Hanaoka. Secure grouping protocol using a deck of cards. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E101.A(9):1512–1524, 2018. URL: https://doi.org/10.1587/transfun.E101.A.1512.
19. Mohammad H. Heydari and I.Hal Sudborough. On the diameter of the pancake network. *Journal of Algorithms*, 25(1):67–94, 1997. URL: https://www.sciencedirect.com/science/article/pii/S0196677497908749, doi:https://doi.org/10.1006/jagm.1997.0874.
20. T. Ibaraki and Y. Manabe. A more efficient card-based protocol for generating a random permutation without fixed points. In *Mathematics and Computers in Sciences and in Industry (MCSI)*, pages 252–257, 2016. URL: https://doi.org/10.1109/MCSI.2016.054.
21. Rie Ishikawa, Eikoh Chida, and Takaaki Mizuki. Efficient card-based protocols for generating a hidden random permutation without fixed points. In Cristian S. Calude and Michael J. Dinneen, editors, *Unconventional Computation and Natural Computation*, volume 9252 of *LNCS*, pages 215–226, Cham, 2015. Springer. URL: https://doi.org/10.1007/978-3-319-21819-9_16.
22. Raimu Isuzugawa, Daiki Miyahara, and Takaaki Mizuki. Zero-knowledge proof protocol for Cryptarithmetic using dihedral cards. In Irina Kostitsyna and Pekka Orponen, editors, *Unconventional Computation and Natural Computation*, volume 12984 of *LNCS*, pages 51–67, Cham, 2021. Springer. URL: https://doi.org/10.1007/978-3-030-87993-8_4.
23. Takehiro Ito, Erik D. Demaine, Nicholas J.A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12):1054–1065, 2011. URL: https://www.sciencedirect.com/science/article/pii/S0304397510006961, doi:https://doi.org/10.1016/j.tcs.2010.12.005.
24. Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, and Yoshio Okamoto. Shortest reconfiguration of perfect matchings via alternating cycles. *SIAM Journal on Discrete Mathematics*, 36(2):1102–1123, 2022. arXiv:https://doi.org/10.1137/20M1364370, doi:10.1137/20M1364370.
25. Kento Kimura, Atsuki Takahashi, Tetsuya Araki, and Kazuyuki Amano. Maximum number of steps of topswops on 18 and 19 cards. *arXiv:2103.08346*, 2021. URL: https://arxiv.org/abs/2103.08346, doi:10.48550/ARXIV.2103.08346.
26. Murray S. Klamkin. *Problems in Applied Mathematics: Selections from SIAM Review*. 1990. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611971729.ch4, arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611971729.ch4, doi:10.1137/1.9781611971729.ch4.
27. D. J. Kleitman, Edvard Kramer, J. H. Conway, Stroughton Bell, and Harry Dweighter. Elementary problems: E2564-e2569. *The American Mathematical Monthly*, 82(10):1009–1010, 1975. URL: http://www.jstor.org/stable/2318260.
28. Alexander Koch. The landscape of optimal card-based protocols. *Mathematical Cryptology*, 1(2):115–131, 2022. URL: https://journals.flvc.org/mathcryptology/article/view/130529.

29. Yuichi Komano and Takaaki Mizuki. Physical zero-knowledge proof protocol for Topswops. In Chunhua Su, Dimitris Gritzalis, and Vincenzo Piuri, editors, *Information Security Practice and Experience - 17th International Conference, ISPEC 2022*, volume 13620 of *Lecture Notes in Computer Science*, pages 537–553. Springer, 2022. URL: https://doi.org/10.1007/978-3-031-21280-2_30.
30. Y. Kounoike, K. Kaneko, and Y. Shinano. Computing the diameters of 14- and 15-pancake graphs. In *8th International Symposium on Parallel Architectures,Algorithms and Networks (ISPAN'05)*, pages 6 pp.–, 2005. doi:10.1109/ISPAN.2005.31.
31. Tomoki Kuzuma, Kodai Toyoda, Daiki Miyahara, and Takaaki Mizuki. Card-based single-shuffle protocols for secure multiple-input AND and XOR computations. In *ASIA Public-Key Cryptography*, pages 51–58, NY, 2022. ACM. URL: https://doi.org/10.1145/3494105.3526236.
32. Pascal Lafourcade, Daiki Miyahara, Takaaki Mizuki, Léo Robert, Tatsuya Sasaki, and Hideaki Sone. How to construct physical zero-knowledge proofs for puzzles with a "single loop" condition. *Theor. Comput. Sci.*, 888:41–55, 2021. URL: https://doi.org/10.1016/j.tcs.2021.07.019.
33. Pascal Lafourcade, Daiki Miyahara, Takaaki Mizuki, Tatsuya Sasaki, and Hideaki Sone. A physical ZKP for Slitherlink: How to perform physical topology-preserving computation. In Swee-Huay Heng and Javier Lopez, editors, *Information Security Practice and Experience*, volume 11879 of *LNCS*, pages 135–151, Cham, 2019. Springer. URL: https://doi.org/10.1007/978-3-030-34339-2_8.
34. Yoshifumi Manabe and Hibiki Ono. Secure card-based cryptographic protocols using private operations against malicious players. In Diana Maimut, Andrei-George Oprina, and Damien Sauveron, editors, *Innovative Security Solutions for Information Technology and Communications*, volume 12596 of *LNCS*, pages 55–70, Cham, 2021. Springer. URL: https://doi.org/10.1007/978-3-030-69255-1_5.
35. Yoshifumi Manabe and Hibiki Ono. Card-based cryptographic protocols with malicious players using private operations. *New Gener. Comput.*, 40:67–93, 2022. URL: https://doi.org/10.1007/s00354-021-00148-w.
36. Daiki Miyahara, Hiromichi Haneda, and Takaaki Mizuki. Card-based zero-knowledge proof protocols for graph problems and their computational model. In Qiong Huang and Yu Yu, editors, *Provable and Practical Security*, volume 13059 of *Lecture Notes in Computer Science*, pages 136–152, Cham, 2021. Springer. URL: https://doi.org/10.1007/978-3-030-90402-9_8.
37. Daiki Miyahara, Yu ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. Practical card-based implementations of Yao's millionaire protocol. *Theor. Comput. Sci.*, 803:207–221, 2020. URL: https://doi.org/10.1016/j.tcs.2019.11.005.
38. Daiki Miyahara and Takaaki Mizuki. Secure computations through checking suits of playing cards. In *Frontiers in Algorithmics*, Lecture Notes in Computer Science, Cham, 2022. Springer. to appear.
39. Daiki Miyahara, Léo Robert, Pascal Lafourcade, So Takeshige, Takaaki Mizuki, Kazumasa Shinagawa, Atsuki Nagao, and Hideaki Sone. Card-based ZKP protocols for Takuzu and Juosan. In Martin Farach-Colton, Giuseppe Prencipe, and Ryuhei Uehara, editors, *Fun with Algorithms*, volume 157 of *LIPIcs*, pages 20:1–20:21, Dagstuhl, Germany, 2020. Schloss Dagstuhl. URL: https://doi.org/10.4230/LIPIcs.FUN.2021.20.
40. Daiki Miyahara, Tatsuya Sasaki, Takaaki Mizuki, and Hideaki Sone. Card-based physical zero-knowledge proof for Kakuro. *IEICE Trans. Fundam. Electron. Com-*

mun. Comput. Sci., 102(9):1072–1078, 2019. URL: https://doi.org/10.1587/transfun.E102.A.1072.

41. Kengo Miyamoto and Kazumasa Shinagawa. Graph automorphism shuffles from pile-scramble shuffles. *New Gener. Comput.*, 40:199–223, 2022. URL: https://doi.org/10.1007/s00354-022-00164-4.
42. Takaaki Mizuki. Preface: Special issue on card-based cryptography. *New Gener. Comput.*, 39:1–2, 2021. URL: https://doi.org/10.1007/s00354-021-00127-1.
43. Takaaki Mizuki. Preface: Special issue on card-based cryptography 2. *New Gener. Comput.*, 40:47–48, 2022. URL: https://doi.org/10.1007/s00354-022-00170-6.
44. Takaaki Mizuki and Yuichi Komano. Information leakage due to operative errors in card-based protocols. *Inf. Comput.*, 285:104910, 2022. URL: https://doi.org/10.1016/j.ic.2022.104910.
45. Linda Morales and Hal Sudborough. A quadratic lower bound for topswops. *Theoretical Computer Science*, 411(44):3965–3970, 2010. URL: https://www.sciencedirect.com/science/article/pii/S0304397510004287, doi:https://doi.org/10.1016/j.tcs.2010.08.011.
46. Takeshi Nakai, Yuto Misawa, Yuuki Tokushige, Mitsugu Iwamoto, and Kazuo Ohta. Secure computation for threshold functions with physical cards: Power of private permutations. *New Gener. Comput.*, 40:95–113, 2022. URL: https://doi.org/10.1007/s00354-022-00153-7.
47. Akihiro Nishimura, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. Pile-shifting scramble for card-based protocols. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 101(9):1494–1502, 2018. URL: https://doi.org/10.1587/transfun.E101.A.1494.
48. Léo Robert, Daiki Miyahara, Pascal Lafourcade, Luc Libralesso, and Takaaki Mizuki. Physical zero-knowledge proof and NP-completeness proof of Suguru puzzle. *Information and Computation*, page 104858, 2021. in press. URL: https://www.sciencedirect.com/science/article/pii/S0890540121001905, doi:https://doi.org/10.1016/j.ic.2021.104858.
49. Léo Robert, Daiki Miyahara, Pascal Lafourcade, and Takaaki Mizuki. Card-based ZKP for connectivity: Applications to Nurikabe, Hitori, and Heyawake. *New Gener. Comput.*, pages 1–23, 2022. in press. URL: https://doi.org/10.1007/s00354-022-00155-5.
50. Léo Robert, Pascal Lafourcade, Daiki Miyahara, and Takaaki Mizuki. Card-based ZKP protocol for Nurimisaki. In Stéphane Devismes, Franck Petit, Karine Altisen, Giuseppe Antonio Di Luna, and Antonio Fernandez Anta, editors, *Stabilization, Safety, and Security of Distributed Systems*, volume 13751 of *LNCS*, pages 285–298, Cham, 2022. Springer.
51. Léo Robert, Daiki Miyahara, Pascal Lafourcade, and Takaaki Mizuki. Physical zero-knowledge proof for Suguru puzzle. In Stéphane Devismes and Neeraj Mittal, editors, *Stabilization, Safety, and Security of Distributed Systems*, volume 12514 of *LNCS*, pages 235–247, Cham, 2020. Springer. URL: https://doi.org/10.1007/978-3-030-64348-5_19.
52. Léo Robert, Daiki Miyahara, Pascal Lafourcade, and Takaaki Mizuki. Interactive physical ZKP for connectivity: Applications to Nurikabe and Hitori. In Liesbeth De Mol, Andreas Weiermann, Florin Manea, and David Fernández-Duque, editors, *Connecting with Computability*, volume 12813 of *LNCS*, pages 373–384, Cham, 2021. Springer. URL: https://doi.org/10.1007/978-3-030-80049-9_37.
53. Léo Robert, Daiki Miyahara, Pascal Lafourcade, and Takaaki Mizuki. Hide a liar: Card-based ZKP protocol for Usowan. In *Theory and Applications of Models of Computation*, LNCS, Cham, 2022. Springer. to appear.

54. Suthee Ruangwises. An improved physical ZKP for Nonogram. In Ding-Zhu Du, Donglei Du, Chenchen Wu, and Dachuan Xu, editors, *Combinatorial Optimization and Applications*, volume 13135 of *LNCS*, pages 262–272, Cham, 2021. Springer. URL: https://doi.org/10.1007/978-3-030-92681-6_22.
55. Suthee Ruangwises. Two standard decks of playing cards are sufficient for a ZKP for Sudoku. In Chi-Yeh Chen, Wing-Kai Hon, Ling-Ju Hung, and Chia-Wei Lee, editors, *Computing and Combinatorics*, volume 13025 of *LNCS*, pages 631–642, Cham, 2021. Springer. URL: https://doi.org/10.1007/978-3-030-89543-3_52.
56. Suthee Ruangwises. Using five cards to encode each integer in Z/6Z. In Peter Y. A. Ryan and Cristian Toma, editors, *Innovative Security Solutions for Information Technology and Communications - 14th International Conference, SecITC 2021*, volume 13195 of *Lecture Notes in Computer Science*, pages 165–177, Cham, 2021. Springer. URL: https://doi.org/10.1007/978-3-031-17510-7_12.
57. Suthee Ruangwises. Two standard decks of playing cards are sufficient for a ZKP for sudoku. *New Gener. Comput.*, pages 1–17, 2022. in press. URL: https://doi.org/10.1007/s00354-021-00146-y.
58. Suthee Ruangwises and Toshiya Itoh. Physical zero-knowledge proof for Numberlink. In Martin Farach-Colton, Giuseppe Prencipe, and Ryuhei Uehara, editors, *Fun with Algorithms*, volume 157 of *LIPIcs*, pages 22:1–22:11, Dagstuhl, Germany, 2020. Schloss Dagstuhl. URL: https://doi.org/10.4230/LIPIcs.FUN.2021.22.
59. Suthee Ruangwises and Toshiya Itoh. Securely computing the n-variable equality function with $2n$ cards. In Jianer Chen, Qilong Feng, and Jinhui Xu, editors, *Theory and Applications of Models of Computation*, volume 12337 of *LNCS*, pages 25–36, Cham, 2020. Springer. URL: https://doi.org/10.1007/978-3-030-59267-7_3.
60. Suthee Ruangwises and Toshiya Itoh. Physical zero-knowledge proof for Numberlink puzzle and k vertex-disjoint paths problem. *New Gener. Comput.*, 39(1):3–17, 2021. URL: https://doi.org/10.1007/s00354-020-00114-y.
61. Suthee Ruangwises and Toshiya Itoh. Physical zero-knowledge proof for Ripple Effect. In Seokhee Hong, Subhas Nandy, and Ryuhei Uehara, editors, *WALCOM: Algorithms and Computation*, volume 11737 of *LNCS*, pages 296–307, Cham, 2021. Springer. URL: https://doi.org/10.1007/978-3-030-68211-8_24.
62. Suthee Ruangwises and Toshiya Itoh. Physical zero-knowledge proof for Ripple Effect. *Theor. Comput. Sci.*, 895:115–123, 2021. URL: https://doi.org/10.1016/j.tcs.2021.09.034.
63. Suthee Ruangwises and Toshiya Itoh. Physical ZKP for connected spanning subgraph: Applications to Bridges Puzzle and other problems. In *Unconventional Computation and Natural Computation*, volume 12984 of *LNCS*, Cham, 2021. Springer. URL: https://doi.org/10.1007/978-3-030-87993-8_10.
64. Suthee Ruangwises and Toshiya Itoh. Securely computing the n-variable equality function with 2n cards. *Theor. Comput. Sci.*, 887:99–110, 2021. URL: https://doi.org/10.1016/j.tcs.2021.07.007.
65. Suthee Ruangwises and Toshiya Itoh. How to physically verify a rectangle in a grid: A physical ZKP for Shikaku. In Pierre Fraigniaud and Yushi Uno, editors, *Fun with Algorithms*, volume 226 of *LIPIcs*, pages 24:1–24:12, Dagstuhl, 2022. Schloss Dagstuhl. URL: https://doi.org/10.4230/LIPIcs.FUN.2022.24.
66. Suthee Ruangwises and Toshiya Itoh. Physical ZKP for Makaro using a standard deck of cards. In *Theory and Applications of Models of Computation*, LNCS, Cham, 2022. Springer. to appear.

67. Tatsuya Sasaki, Daiki Miyahara, Takaaki Mizuki, and Hideaki Sone. Efficient card-based zero-knowledge proof for Sudoku. *Theor. Comput. Sci.*, 839:135–142, 2020. URL: https://doi.org/10.1016/j.tcs.2020.05.036.
68. Tatsuya Sasaki, Takaaki Mizuki, and Hideaki Sone. Card-based zero-knowledge proof for Sudoku. In Hiro Ito, Stefano Leonardi, Linda Pagli, and Giuseppe Prencipe, editors, *Fun with Algorithms*, volume 100 of *LIPIcs*, pages 29:1–29:10, Dagstuhl, Germany, 2018. Schloss Dagstuhl. URL: https://doi.org/10.4230/LIPIcs.FUN.2018.29.
69. Joe Sawada and Aaron Williams. Greedy flipping of pancakes and burnt pancakes. *Discret. Appl. Math.*, 210:61–74, 2016. doi:10.1016/j.dam.2016.02.005.
70. Joe Sawada and Aaron Williams. Successor rules for flipping pancakes and burnt pancakes. *Theor. Comput. Sci.*, 609:60–75, 2016. doi:10.1016/j.tcs.2015.09.007.
71. Hayato Shikata, Kodai Toyoda, Daiki Miyahara, and Takaaki Mizuki. Card-minimal protocols for symmetric boolean functions of more than seven inputs. In Helmut Seidl, Zhiming Liu, and Corina S. Pasareanu, editors, *Theoretical Aspects of Computing – ICTAC 2022*, pages 388–406, Cham, 2022. Springer International Publishing.
72. Kazumasa Shinagawa, Takaaki Mizuki, Jacob Schuldt, Koji Nuida, Naoki Kanayama, Takashi Nishide, Goichiro Hanaoka, and Eiji Okamoto. Card-based protocols using regular polygon cards. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E100.A(9):1900–1909, 2017. URL: https://doi.org/10.1587/transfun.E100.A.1900.
73. Kazumasa Shinagawa and Koji Nuida. A single shuffle is enough for secure card-based computation of any Boolean circuit. *Discrete Applied Mathematics*, 289:248–261, 2021. URL: https://doi.org/10.1016/j.dam.2020.10.013.
74. Kazumasa Shinagawa and Koji Nuida. Single-shuffle full-open card-based protocols imply private simultaneous messages protocols. Cryptology ePrint Archive, Paper 2022/1306, 2022. https://eprint.iacr.org/2022/1306. URL: https://eprint.iacr.org/2022/1306.
75. Yuto Shinoda, Daiki Miyahara, Kazumasa Shinagawa, Takaaki Mizuki, and Hideaki Sone. Card-based covert lottery. In Diana Maimut, Andrei-George Oprina, and Damien Sauveron, editors, *Innovative Security Solutions for Information Technology and Communications*, volume 12596 of *LNCS*, pages 257–270, Cham, 2021. Springer. URL: https://doi.org/10.1007/978-3-030-69255-1_17.
76. Yuji Suga. A classification proof for commutative three-element semigroups with local AND structure and its application to card-based protocols. In *2022 IEEE International Conference on Consumer Electronics - Taiwan*, pages 171–172, NY, 2022. IEEE. URL: https://doi.org/10.1109/ICCE-Taiwan55306.2022.9869063.
77. Ken Takashima, Yuta Abe, Tatsuya Sasaki, Daiki Miyahara, Kazumasa Shinagawa, Takaaki Mizuki, and Hideaki Sone. Card-based protocols for secure ranking computations. *Theor. Comput. Sci.*, 845:122–135, 2020. URL: https://doi.org/10.1016/j.tcs.2020.09.008.