Physical Zero-knowledge Proof Protocol for Topswops[⋆]

Yuichi Komano[1] and Takaaki Mizuki[2]

[1] Toshiba Corporation, 1 Komukai-Toshiba-Cho, Saiwai-ku, Kawasaki, Japan
yuichi1.komano[atmark]toshiba.co.jp
[2] Tohoku University, 6–3 Aramaki-Aza-Aoba, Aoba-ku, Sendai, Japan
mizuki+lncs[atmark]tohoku.ac.jp

Abstract. Suppose that a sequence of n cards, numbered 1 to n, is placed face up in random order. Let k be the number on the first card in the sequence. Then take the first k cards from the sequence, rearrange that subsequence of k cards in reverse order, and return them to the original sequence. Repeat this prefix reversal until the number on the first card in the sequence becomes 1. This is a one-player card game called Topswops. The computational complexity of Topswops has not been thoroughly investigated. For example, letting $f(n)$ denote the maximum number of prefix reversals for Topswops with n cards, values of $f(n)$ for $n \geq 20$ remain unknown. In general, there is no known efficient algorithm for finding an initial sequence of n cards that requires exactly ℓ prefix reversals for any integers n and ℓ. In this paper, we propose a physical zero-knowledge proof protocol that allows a prover to convince a verifier that the prover knows an initial sequence of n cards that requires ℓ prefix reversals without leaking knowledge of that sequence.

Keywords: Zero-knowledge Proof, Card-based Cryptography, Topswops

1 Introduction

Topswops is a one-player card game in which one randomly arranges n cards, numbered 1 to n. Given such an initial sequence of n cards, the player rearranges the sequence as follows. After looking at the number written on the first card, denoted as k, the player rearranges the first k cards into their reverse order while leaving the remaining $n-k$ cards unchanged. The player repeatedly applies such prefix reversals to the current sequence until the number on the first card in the sequence becomes 1. For example, consider the case where $n = 5$, and

[⋆] This paper appears in Proceedings of ISPEC 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/10.1007/978-3-031-21280-2_30. 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.

Table 1. Values for $f(n)$ and $g(n)$.

n		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$f(n)$	(OEIS A000375)	0	1	2	4	7	10	16	22	30	38	51	65	80	101	113	139	159	191	221
$g(n)$	(OEIS A123398)	1	1	2	2	2	1	5	2	1	1	1	1	1	4	6	1	2	1	4

let $(3,1,4,5,2)$ be an initial sequence. In Topswops, the player rearranges the sequence as

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

so the game ends in seven steps (namely, seven prefix reversals). In this example, the cards are finally sorted in ascending order. Note that the number of steps depends on an individual initial sequence and that the n cards may not be finally sorted in ascending order. For instance, if the initial sequence is $(2,1,3,5,4)$, the game ends with $(1,2,3,5,4)$ after one step, with the cards unsorted.

1.1 Open Problems in Topswops

Topswops is said to be invented by J.H. Conway (see the Introduction in [38] or p. 116 of [17]). The question of what initial sequences produce the maximum number of steps has been investigated. For example, recall that the initial sequence $(3,1,4,5,2)$ shown above requires exactly seven steps; it is known that in the case of $n = 5$, there is no initial sequence that results in more than seven steps, and hence, seven is the maximum number of steps for Topswops with five cards. Therefore, letting $f(n)$ denote the maximum number of steps for Topswops with n cards, we have $f(5) = 7$.

Generally, the currently known upper bound on $f(n)$ is $O(1.62^n)$ [17,19] and the currently known lower bound is $\Omega(n^2)$ [38]. Thus, there is an exponential gap between the (currently known) upper and lower bounds on $f(n)$, and finding better bounds to reduce that gap is an open problem.

Since there is no known efficient algorithm for finding the values of $f(n)$, several studies have used brute-force searches to report values for $f(n)$ and the corresponding initial sequences (e.g., [16,41]). Specifically, values are known up to $n \leq 19$. Table 1 shows known values for $f(n)$, which are registered as OEIS A000375[3] in *The On-Line Encyclopedia of Integer Sequences* (OEIS). That table also shows as $g(n)$ the number of initial sequences requiring $f(n)$ steps, which are registered as OEIS A123398[4]. The most recent result was that Kimura et al. [16] obtained values for $f(18), f(19)$ (and $g(18), g(19)$) in 2021, using 172 threads on nine computers over about six days[5]. Finding values of $f(n)$ and $g(n)$ for $n \geq 20$

[3] `https://oeis.org/A000375`

[4] `https://oeis.org/A123398`

[5] According to Zimmermann's Programming Contests (`http://azspcs.com/Contest/Cards/FinalReport`, accessed 16 Aug 2022), four initial sequences which require 221

has remained an important open problem for many years, along with narrowing the gap between the upper and lower bounds on $f(n)$, as described above.

1.2 Zero-knowledge Proof Protocol for Topswops

As mentioned above, the computational complexity of Topswops has been insufficiently revealed, and many open problems related to Topswops remain. In particular, for any positive integers n and ℓ (such that $1 \leq \ell \leq f(n)$), there is no known efficient algorithm for finding an initial sequence of n cards that requires exactly ℓ steps. Therefore, there are possible situations where it is invaluable that only a single person knows a particular initial sequence (and its number of steps) while the others do not know anything about it. Hence, let us try to apply the concept of *zero-knowledge proof* [8] to Topswops. In other words, let us consider a zero-knowledge proof protocol for Topswops whereby a prover P can convince a verifier V that P knows an initial sequence requiring exactly ℓ steps without leaking any information about the sequence. The following illustrates two such situations.

- Suppose that prover P has discovered an initial sequence of 20 cards requiring 250 steps, which is longer than the currently known lower bound $f(20) \geq 249$[6], and wants to publish it as a new achievement. However, if P shows the initial sequence to a (possibly malicious) third party, that party might claim it as its own achievement. If, however, a zero-knowledge proof protocol could be applied to Topswops, P could prove that P discovered the initial sequence while still keeping it secret. After gaining sufficient recognition, P could claim the initial sequence by disclosing it without any trouble.
- Suppose that several players want to play a game in which they try to find an initial sequence of Topswops. The rules of the game state that the winner is the player who finds an initial sequence with a larger number of steps, or one with a predetermined number of steps. Assume that the game may be played many times, with different players and different conditions. If the initial sequences are disclosed every time to determine a game winner, those sequences lead to another sequence corresponding to a smaller number of steps, or it will serve as a clue for finding another sequence having a larger number of steps, which makes it impossible to fairly or enjoyably play subsequent games. Using a zero-knowledge proof protocol to determine the game winner, however, would make it possible to enjoy the game.

1.3 Contribution

We propose a zero-knowledge proof protocol for Topswops. Namely, assuming that positive integers n and ℓ (such that $1 \leq \ell \leq f(n)$) are public, when a prover

steps for $n = 19$ were discovered in 2011. Hence, it seems that at that time the lower bounds on $f(19)$ and $g(19)$ were known to be 221 and 4, respectively.

[6] At the present moment, of course, it is unknown whether such an initial sequence exists, i.e., whether $f(20) \geq 250$ or $f(20) = 249$.

P knows an initial sequence of n cards requiring exactly ℓ steps, P can convince a verifier V that P knows the initial sequence without disclosing it. Our protocol is a so-called *physical* zero-knowledge proof protocol that can be executed using a physical deck of cards.

1.4 Related Work

In the literature, many physical zero-knowledge proof protocols have been constructed using a physical deck of cards, especially those for pencil puzzles such as Sudoku. Examples include Akari [2], Cryptarithmetic [15], Hashiwokakero (Bridges) [56], Heyawake [43], Hitori [43,46], Juosan [30], Kakuro [2,31], KenKen [2], Makaro [3, 59], Masyu [24], Nonogram [4, 48], Norinori [6], Numberlink [52, 53], Nurikabe [43, 46], Nurimisaki [44], Ripple Effect [54, 55], Shikaku [58], Slitherlink [24,25], Sudoku [49,51,60,61], Suguru [42,45], Takuzu [2,30], and Usowan [47].

Since our protocol uses a physical deck of cards, this study falls into the research area of *card-based cryptography* (refer to [20, 37, 63] for surveys), which has been rapidly growing recently, e.g., [33, 34]. The recent results on card-based cryptography include proposing efficient protocols for multi-valued-output symmetric functions [57, 62], analyzing information leakage due to operative errors [35], introducing graph automorphism shuffles [32], designing a secure sorting protocol [10], lowering the numbers of required cards [9] and shuffles [23], constructing multi-valued protocols with a direction encoding [64], introducing new encoding schemes for integers [50], exploring half-open-action protocols [29], and determining the landscape of card-minimal protocols in terms of running time requirements [21].

2 Preliminaries

In this section, we first explain the notations used in this paper, then describe how and what physical cards are utilized, and introduce the "pile-scramble shuffle [14]" and the "sort sub-protocol [22]," both of which are used in our protocol as sub-protocols.

2.1 Notations

Let n denote the Topswops size (the number of cards in an initial sequence), as in Section 1.

We regard any initial sequence of n cards and its succeeding sequences that appear in Topswops as permutations on $\{1, 2, \ldots, n\}$. That is, a permutation $\pi \in S_n$, i.e.,

$$\pi = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ \pi(1) & \pi(2) & \pi(3) & \cdots & \pi(n) \end{pmatrix},$$

represents a sequence of n cards

$$(\pi(1), \pi(2), \pi(3), \ldots, \pi(n)),$$

where S_n denotes the symmetric group of degree n.

For the initial sequence corresponding to a permutation $\pi \in S_n$, denote by $\alpha(\pi)$ the number of steps required in Topswops. Using this notation $\alpha(\pi)$, we formally define $f(n)$, which is the maximum number of steps for n cards, as

$$f(n) := \max\left\{\alpha(\pi) \mid \pi \in S_n\right\}.$$

Using a permutation, let us represent each prefix reversal operation in Topswops, which rearranges the first i cards in their reverse order: Define a permutation $\mathsf{sw}_i \in S_n$ for every i, $2 \le i \le n$, as

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

which is sometimes referred to as the *swap* sw_i. Note that sw_i is equal to its inverse permutation sw_i^{-1}. That is, for every i such that $2 \le i \le n$,

$$\mathsf{sw}_i = \mathsf{sw}_i^{-1} \tag{1}$$

holds.

2.2 Commitment Based on Physical Cards

The goal of this paper is to construct a zero-knowledge proof protocol for Topswops, which should keep an initial sequence and its subsequent sequences secret. Therefore, as usual in card-based cryptography, we place cards face down and perform a series of operations such as shuffling, rearranging, and turning over cards.

In this paper, we use a deck of physical cards numbered 1 to n, denoted by $\boxed{1}\boxed{2}\boxed{3}\cdots\boxed{n}$, whose backs are all identical $\boxed{?}$. We assume that cards $\boxed{i}\boxed{i}\cdots\boxed{i}$ having the same number i are all indistinguishable. By

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

we denote a face-down card whose face is $\boxed{i}$ for a number i.

As explained in the previous subsection, an initial sequence of n cards can be represented by a permutation $\pi \in S_n$, so a prover P can take n physical cards $\boxed{1}\boxed{2}\boxed{3}\cdots\boxed{n}$ and place them face down according to the initial sequence π:

$$\underset{\pi(1)}{\boxed{?}}\;\underset{\pi(2)}{\boxed{?}}\;\underset{\pi(3)}{\boxed{?}}\;\cdots\;\underset{\pi(n)}{\boxed{?}}\,.$$

It thus seems possible to keep the initial sequence and its succeeding sequences secret. However, it is difficult to apply a swap sw_i to such a physical sequence while keeping the integer i secret, and hence, such a simple representation does not work.

We therefore introduce a novel encoding of integers based on swaps sw_i: an integer $x \in \{2, 3, \ldots, n\}$ is encoded with n cards as

$$\boxed{\mathsf{sw}_x(1)}\boxed{\mathsf{sw}_x(2)}\boxed{\mathsf{sw}_x(3)}\cdots\boxed{\mathsf{sw}_x(n)}. \tag{2}$$

In other words, each integer $x \in \{2, 3, \ldots, n\}$ is represented by

$$\begin{array}{l}
\boxed{2}\boxed{1}\boxed{3}\boxed{4}\cdots\boxed{n} = 2\\
\boxed{3}\boxed{2}\boxed{1}\boxed{4}\cdots\boxed{n} = 3\\
\quad\vdots\\
\boxed{i}\boxed{i-2}\boxed{i-3}\cdots\boxed{1}\boxed{i+1}\cdots\boxed{n} = i\\
\quad\vdots\\
\boxed{n}\boxed{n-1}\cdots\boxed{1} = n.
\end{array}$$

In addition, $x = 1$ is represented with n cards of $\boxed{1}$ as

$$\boxed{1}\boxed{1}\cdots\boxed{1} = 1. \tag{3}$$

We call a sequence of n face-down cards encoding an integer $x \in \{1, 2, \ldots, n\}$ (according to the encoding rules (2) and (3) above) a *commitment* to x. For example, a commitment to 1 is a sequence of n face-down cards

$$\underset{1}{\boxed{?}}\underset{1}{\boxed{?}}\cdots\underset{1}{\boxed{?}},$$

and a commitment to x such that $x \geq 2$ is

$$\underset{\mathsf{sw}_x(1)}{\boxed{?}}\;\underset{\mathsf{sw}_x(2)}{\boxed{?}}\;\cdots\;\underset{\mathsf{sw}_x(n)}{\boxed{?}}.$$

Without distinguishing between the case $x = 1$ and the case $x > 1$, a commitment to x is denoted by

$$\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{x} \quad \text{or} \quad \underbrace{\boxed{?}\!|||}_{x}.$$

2.3 Pile-scramble Shuffle

A *pile-scramble shuffle* [14] is a shuffling operation by which several piles of cards of the same size are shuffled.

As an example, suppose we have n commitments and a sequence of n face-down cards corresponding to a permutation $\pi \in S_n$ as follows:

$$\begin{array}{cccc}
1 & 2 & & n\\
\boxed{?}\!||| & \boxed{?}\!||| & \cdots & \boxed{?}\!|||\\
\boxed{?} & \boxed{?} & \cdots & \boxed{?}\\
\pi(1) & \pi(2) & & \pi(n)
\end{array}, \tag{4}$$

where numbers above the commitments represent indexes for convenience. Considering each commitment and the card below it as a single pile, apply a pile-scramble shuffle to the n piles. The transition is

$$\left[\begin{array}{c|c|c|c} \boxed{?|||} & \boxed{?|||} & \cdots & \boxed{?|||} \\ \boxed{?} & \boxed{?} & \cdots & \boxed{?} \end{array}\right] \rightarrow \begin{array}{cccc} r^{-1}(1) & r^{-1}(2) & \cdots & r^{-1}(n) \\ \boxed{?|||} & \boxed{?|||} & & \boxed{?|||} \\ \boxed{?} & \boxed{?} & \cdots & \boxed{?} \\ \pi(r^{-1}(1)) & \pi(r^{-1}(2)) & & \pi(r^{-1}(n)) \end{array}, \tag{5}$$

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

To implement a pile-scramble shuffle, we fix each pile of cards and shuffle the piles [14].

2.4 Sort Sub-protocol

Recall arrangement (4) above, where we have a sequence of n commitments and a sequence of cards corresponding to a permutation $\pi \in S_n$. Suppose that, after applying a pile-scramble shuffle to them as in transition (5), we turn over all n cards on the second row. Each number 1 to n should appear. We then sort the piles (keeping the order within each pile unchanged) so that the revealed numbers are in ascending order. We then obtain the sequence of piles

$$\begin{array}{cccc} \pi^{-1}(1) & \pi^{-1}(2) & \cdots & \pi^{-1}(n) \\ \boxed{?|||} & \boxed{?|||} & & \boxed{?|||} \\ \boxed{1} & \boxed{2} & \cdots & \boxed{n} \end{array}$$

(because, for example, if $\pi(r^{-1}(i)) = 1$, then $r^{-1}(i) = \pi^{-1}(1)$ and the commitment at position $\pi^{-1}(1)$ moves to the first). We have thus rearranged the sequence of n commitments according to the permutation $\pi \in S_n$ while keeping π secret.

This technique has been commonly used in constructions of zero-knowledge proof protocols for pencil puzzles since the introduction of a zero-knowledge proof protocol for Sudoku [61] that made full use of card-based cryptography. Koch and Walzer [22] formulated this technique and named it the *sort sub-protocol*. Note that the idea of applying a pile-scramble shuffle to two sequences of cards corresponding to some permutations originally comes from [13], followed by [11].

3 Proposed Protocol

In this section, we propose a zero-knowledge proof protocol for Topswops. Let positive integers n and ℓ (such that $1 \leq \ell \leq f(n)$) be public information. Suppose that a prover P knows $\sigma \in S_n$ such that $\alpha(\sigma) = \ell$, i.e., an initial sequence σ of n cards requiring exactly ℓ steps. P wants to convince a verifier V that P knows an initial sequence requiring ℓ steps without revealing any information about the initial sequence σ. We will construct a card-based protocol to achieve this.

First, Section 3.1 describes the idea behind the proposed protocol, then Section 3.2 presents a copy protocol we use as a sub-protocol. Finally, Section 3.3 presents our protocol.

3.1 Idea

This subsection describes the idea behind the proposed protocol using the initial $n = 5$ sequence $(3, 1, 4, 5, 2)$ (which has seven steps) as a working example.

First, prover P, who knows the initial sequence, secretly creates a sequence of commitments corresponding to $(3, 1, 4, 5, 2)$:

$$\underbrace{\boxed{?}\!\|\!\|}_{3}\;\underbrace{\boxed{?}\!\|\!\|}_{1}\;\underbrace{\boxed{?}\!\|\!\|}_{4}\;\underbrace{\boxed{?}\!\|\!\|}_{5}\;\underbrace{\boxed{?}\!\|\!\|}_{2}\,. \tag{6}$$

Recall that each commitment follows the encoding (2) and (3) defined in Section 2.2. For example, the first commitment

$$\underbrace{\boxed{?}\!\|\!\|}_{3}\,,$$

i.e., the commitment to 3, is

$$\underset{3}{\boxed{?}}\underset{2}{\boxed{?}}\underset{1}{\boxed{?}}\underset{4}{\boxed{?}}\underset{5}{\boxed{?}}\,,$$

which is a permutation corresponding to the swap sw_3.

Suppose that we can somehow copy this commitment to 3. We then place the five copied face-down cards below the sequence of commitments as

$$\begin{array}{ccccc} \underbrace{\boxed{?}\!\|\!\|}_{3} & \underbrace{\boxed{?}\!\|\!\|}_{1} & \underbrace{\boxed{?}\!\|\!\|}_{4} & \underbrace{\boxed{?}\!\|\!\|}_{5} & \underbrace{\boxed{?}\!\|\!\|}_{2} \\ \underset{3}{\boxed{?}} & \underset{2}{\boxed{?}} & \underset{1}{\boxed{?}} & \underset{4}{\boxed{?}} & \underset{5}{\boxed{?}} \end{array}.$$

Considering each commitment and the card below as a single pile, we then apply a pile-scramble shuffle. After that, as in the sort sub-protocol described in Section 2.4, we turn over the five cards in the second row and sort the piles so that the five cards are rearranged in ascending order. We thus obtain the following:

$$\begin{array}{ccccc} \underbrace{\boxed{?}\!\|\!\|}_{4} & \underbrace{\boxed{?}\!\|\!\|}_{1} & \underbrace{\boxed{?}\!\|\!\|}_{3} & \underbrace{\boxed{?}\!\|\!\|}_{5} & \underbrace{\boxed{?}\!\|\!\|}_{2} \\ \boxed{1} & \boxed{2} & \boxed{3} & \boxed{4} & \boxed{5} \end{array}.$$

The order of the first three commitments is reversed in the resulting sequence of commitments, which means that the swap sw_3 was applied to the initial sequence without leaking any information about sw_3.

The first commitment in the current sequence is a commitment to 4, which corresponds to the swap sw_4. Hence, similar to sw_3 above, if we can copy and place it below the commitments and apply the sort sub-protocol, the swap sw_4 is applied while the integer corresponding to the first commitment remains secret. If we repeat such operations seven times, the first commitment in the sequence should be a commitment to 1. By turning over the first commitment, we can finally confirm that the prover P placed a correct initial sequence.

This is the idea behind our protocol. However, the above procedure cannot guarantee that the sequence of commitments (6) secretly set by P correctly satisfies the encoding rules. Thus, the proposed protocol, shown in Section 3.3, requires another step to ensure that the sequence is correctly rearranged.

3.2 Copy Protocol

As mentioned in Section 3.1, our main protocol requires a copied commitment. The following describes how to make two identical commitments from a single commitment while its value is kept secret.

Given a commitment to x such that $2 \leq x \leq n$,

$$\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{x} \quad = \quad \underset{\mathsf{sw}_x(1)}{\boxed{?}}\;\underset{\mathsf{sw}_x(2)}{\boxed{?}}\;\cdots\;\underset{\mathsf{sw}_x(n)}{\boxed{?}},$$

along with two sets of $\boxed{1}$ through $\boxed{n}$ as additional cards, our copy protocol proceeds as follows.

Protocol 1 (Copy protocol)

1. *Commitments and additional cards are placed as follows:*

$$\begin{array}{cccc} \boxed{1} & \boxed{2} & \cdots & \boxed{n} \\ \boxed{1} & \boxed{2} & \cdots & \boxed{n} \\ \underset{\mathsf{sw}_x(1)}{\boxed{?}} & \underset{\mathsf{sw}_x(2)}{\boxed{?}} & \cdots & \underset{\mathsf{sw}_x(n)}{\boxed{?}} \end{array}.$$

2. *Turn over the cards in the first and second rows and apply a pile-scramble shuffle:*

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

Then, a uniformly distributed random permutation $r \in S_n$ corresponding to this pile-scramble shuffle occurs, and the resulting cards can be represented as

$$\begin{array}{cccc} \underset{r^{-1}(1)}{\boxed{?}} & \underset{r^{-1}(2)}{\boxed{?}} & \cdots & \underset{r^{-1}(n)}{\boxed{?}} \\ \underset{r^{-1}(1)}{\boxed{?}} & \underset{r^{-1}(2)}{\boxed{?}} & \cdots & \underset{r^{-1}(n)}{\boxed{?}} \\ \underset{\mathsf{sw}_x(r^{-1}(1))}{\boxed{?}} & \underset{\mathsf{sw}_x(r^{-1}(2))}{\boxed{?}} & \cdots & \underset{\mathsf{sw}_x(r^{-1}(n))}{\boxed{?}} \end{array}.$$

3. *Turn over all the cards in the third row. If cards 1 through n appear without omission, sort the piles so that the revealed numbered cards are in ascending order, resulting in the following:*

$$\begin{array}{cccc} \underset{\mathsf{sw}_x^{-1}(1)}{\boxed{?}} & \underset{\mathsf{sw}_x^{-1}(2)}{\boxed{?}} & \cdots & \underset{\mathsf{sw}_x^{-1}(n)}{\boxed{?}} \\ \underset{\mathsf{sw}_x^{-1}(1)}{\boxed{?}} & \underset{\mathsf{sw}_x^{-1}(2)}{\boxed{?}} & \cdots & \underset{\mathsf{sw}_x^{-1}(n)}{\boxed{?}} \\ \boxed{1} & \boxed{2} & \cdots & \boxed{n} \end{array}.$$

Since $\mathsf{sw}_x^{-1} = \mathsf{sw}_x$ from Equation (1), the first and second rows are commitments to x and we have copied commitments

$$\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{x} \qquad \underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{x}.$$

In the last step, the n cards in the third row are turned over, but no information about x is leaked because these cards were uniformly randomized by the random permutation r in Step 2. Note also that if an input sequence is a commitment to 1, or if it is not a valid commitment, this protocol detects that state when the cards are turned over in Step 3.

A copy protocol for a permutation was first proposed in [61], but that protocol requires two pile-scramble shuffles. Our proposed copy protocol is simpler with only one pile-scramble shuffle, because by Equation (1), the permutation sw_i to be copied is equal to its reverse permutation.

3.3 Our Protocol

We are now ready to describe our protocol, which is a so-called *non-interactive* physical zero-knowledge proof protocol that requires no knowledge of the prover after the prover places an input card sequence (see [28] for details).

According to an initial sequence σ such that $\alpha(\sigma) = \ell$, the prover P places n face-down cards on the table without anyone seeing the order as

$$\underset{\sigma^{-1}(1)}{\boxed{?}}\ \underset{\sigma^{-1}(2)}{\boxed{?}}\ \underset{\sigma^{-1}(3)}{\boxed{?}}\ \cdots\ \underset{\sigma^{-1}(n)}{\boxed{?}}\ . \tag{7}$$

The input to our protocol is this sequence (7) along with additional $2n$ cards of $\boxed{1}$ and n cards from each of $\boxed{2}$ to $\boxed{n}$. (The protocol may be executed by a prover or a verifier, or even by a third party[7].)

Protocol 2 (Physical zero-knowledge proof protocol for Topswops)

1. *Using* $2n-1$ *cards of* $\boxed{1}$ *and* $n-1$ *cards of each from* $\boxed{2}$ *to* $\boxed{n}$ *from the additional cards, make a commitment to each of* 1 *through* n *and place them above the input sequence (7) as follows:*

$$\begin{array}{cccc} \underbrace{\boxed{?}\!\|\!\|\!\|}_{1} & \underbrace{\boxed{?}\!\|\!\|\!\|}_{2} & \cdots & \underbrace{\boxed{?}\!\|\!\|\!\|}_{n} \\ \underset{\sigma^{-1}(1)}{\boxed{?}} & \underset{\sigma^{-1}(2)}{\boxed{?}} & \cdots & \underset{\sigma^{-1}(n)}{\boxed{?}} \end{array} .$$

2. *Apply a pile-scramble shuffle:*

$$\left[\begin{array}{c|c|c|c} \underbrace{\boxed{?}\!\|\!\|\!\|}_{1} & \underbrace{\boxed{?}\!\|\!\|\!\|}_{2} & \cdots & \underbrace{\boxed{?}\!\|\!\|\!\|}_{n} \\ \underset{\sigma^{-1}(1)}{\boxed{?}} & \underset{\sigma^{-1}(2)}{\boxed{?}} & \cdots & \underset{\sigma^{-1}(n)}{\boxed{?}} \end{array}\right] .$$

[7] More precisely, it is specified as a card-based protocol formulated in the standard model of card-based cryptography [36, 37]. (Note that there are other models, e.g., [26, 27, 39, 40].)

3. *In the manner of the sort sub-protocol described in Section 2.4, turn over and sort the cards in the bottom row so that the cards in the bottom row are in ascending order:*

$$\begin{array}{cccc} \underbrace{\boxed{?|||}}_{\sigma(1)} & \underbrace{\boxed{?|||}}_{\sigma(2)} & \cdots & \underbrace{\boxed{?|||}}_{\sigma(n)} \\ \boxed{1} & \boxed{2} & \cdots & \boxed{n} \end{array} .$$

   (If any card from 1 *to* n *does not appear in the bottom row, the protocol halts.)*

4. *We have two cards each from* $\boxed{1}$ *to* $\boxed{n}$ *(i.e., the cards revealed in Step 3 and the remaining additional cards not used in Step 1). Using these as additional cards, copy the first commitment,* $\sigma(1)$*, by executing the copy protocol described in Section 3.2 and place it below the sequence of commitments. Hereinafter, we use* x *for* $\sigma(1)$ *in subscripts for simplicity:*

$$\begin{array}{cccc} \underbrace{\boxed{?|||}}_{\sigma(1)} & \underbrace{\boxed{?|||}}_{\sigma(2)} & \cdots & \underbrace{\boxed{?|||}}_{\sigma(n)} \\ \underset{\mathsf{sw}_x(1)}{\boxed{?}} & \underset{\mathsf{sw}_x(2)}{\boxed{?}} & \cdots & \underset{\mathsf{sw}_x(n)}{\boxed{?}} \end{array} .$$

   (If an illegal card appears during the copy protocol, the protocol halts.)

5. *Apply a pile-scramble shuffle, turn over the cards in the bottom row, and sort the cards in the bottom row in ascending order:*

$$\left[\begin{array}{c|c|c|c} \underbrace{\boxed{?|||}}_{\sigma(1)} & \underbrace{\boxed{?|||}}_{\sigma(2)} & \cdots & \underbrace{\boxed{?|||}}_{\sigma(n)} \\ \underset{\mathsf{sw}_x(1)}{\boxed{?}} & \underset{\mathsf{sw}_x(2)}{\boxed{?}} & \cdots & \underset{\mathsf{sw}_x(n)}{\boxed{?}} \end{array}\right]$$
$$\downarrow$$
$$\begin{array}{cccc} \underbrace{\boxed{?|||}}_{\sigma(\mathsf{sw}_x^{-1}(1))} & \underbrace{\boxed{?|||}}_{\sigma(\mathsf{sw}_x^{-1}(2))} & \cdots & \underbrace{\boxed{?|||}}_{\sigma(\mathsf{sw}_x^{-1}(n))} \\ \boxed{1} & \boxed{2} & \cdots & \boxed{n} \end{array} .$$

   After this operation, the first row is a sequence of commitments where the order of the first x *commitments is reversed, i.e., the swap* sw_x *operation has been applied to the original sequence of commitments. (If any card from* 1 *to* n *does not appear, the protocol aborts.)*

6. *Repeat Steps 4 and 5 in the same manner* ℓ *times in total.*
7. *Turn over the first commitment* $\boxed{?|||}$ *and return 'accept' if* $\boxed{1}\boxed{1}\cdots\boxed{1}$ *appears. Otherwise, return 'reject.'*

4 Security and Performance

This section describes security and performance of the protocol proposed in the previous section.

4.1 Security

In this subsection, we prove that our protocol is a zero-knowledge proof protocol. In other words, we show that our protocol satisfies requirements for completeness, soundness, and zero knowledge.

Completeness: Suppose that a prover P correctly sets as input a sequence of n cards encoding an initial sequence $\sigma \in S_n$ such that $\alpha(\sigma) = \ell$. Then, as can be seen from the construction of the protocol, the sequence is not rejected at any step, but accepted at the final step.

Soundness: Assume that an input sequence of n cards is not the encoding of an initial sequence requiring exactly ℓ steps. There are three cases to consider; we will show that the protocol eventually rejects each.
(i) If the input sequence is not encoded for any initial sequence, it is detected and rejected because not every card from 1 to n appears when the cards are turned up in Step 3.
(ii) Assume an initial sequence requiring exactly j $(< \ell)$ steps. When Steps 4 and 5 of our protocol are executed j times, the first commitment will be a commitment to 1, and Step 4 of iteration $(j+1)$ will duplicate the commitment to 1, and hence, our protocol rejects the sequence because n cards of 1 appear in the final step of the copy protocol.
(iii) If an initial sequence requires j $(> \ell)$ steps, the first commitment in the last step is rejected because it is something other than a commitment to 1.

Zero knowledge: Suppose that the sequence set by prover P corresponds to $\sigma \in S_n$ such that $\alpha(\sigma) = \ell$. During the protocol (except for the final step), no information about σ is leaked because the order of revealed cards $\boxed{1}$ to $\boxed{n}$ is always randomized by a pile-scramble shuffle, which is applied immediately before the cards are turned over. In addition, the commitment to 1 opened in the final step is public information (from the definition of Topswops). Thus, the proposed protocol is information-theoretically secure.

4.2 Performance

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

Regarding the number of cards, we use $\boxed{1}$ to $\boxed{n}$ for the input card sequence, as described at the beginning of Section 3.3, with additional $2n$ cards of $\boxed{1}$ and n cards each of $\boxed{2}$ to $\boxed{n}$. Thus, $2n+1$ cards of $\boxed{1}$ and $n+1$ cards of each of $\boxed{2}$ to $\boxed{n}$ are required, which add up to $n^2 + 2n$.

Our protocol uses only pile-scramble shuffles as a shuffle operation. We therefore count the number of pile-scramble shuffles. In our protocol, Step 2 is executed once, and Steps 4 and 5 are repeated ℓ times. These steps include one pile-scramble shuffle, so our protocol requires $2\ell + 1$ pile-scramble shuffles in total.

5 Conclusion

We proposed a physical zero-knowledge proof protocol for Topswops. The main idea is that the permutation corresponding to “the operation of reversing the first i cards” encodes a positive integer i, which enables us to efficiently and secretly perform prefix reversals in Topswops. The proposed protocol uses $n^2 + 2n$ cards and $2\ell + 1$ shuffles, where n is the size of Topswops and ℓ is the number of steps.

In the “game of discovering initial sequences requiring a predefined number of steps” described in Section 1.2, we can expect to make practical use of our protocol when n is not so large. However, in the other example of claiming discovery of an initial sequence for $n = 20$ requiring 250 steps, like that described in Section 1.2, implementing our protocol would require 440 cards and 501 shuffles, which might not be practical to implement. Therefore, we address the following problem as another situation that may be practically useful.

The *pancake problem* [18] is another game similar to Topswops. Given an initial sequence $(\pi(1), \pi(2), \ldots, \pi(n))$, the goal in this game is to sort the sequence in ascending order by repeating swap operations sw_i. Related to this game, the problem of finding a sequence of the minimum number of swap operations needed to complete the sort is proved to be NP-hard [1]. Also, we denote as $h(n)$ the maximum number of such fewest swap operations among all initial sequences of n cards. Then, the best currently known lower and upper bounds[8] are $\frac{15}{14}n \leq h(n)$ [12] and $h(n) \leq \frac{18}{11}n$ [5], respectively.

We expect that the idea of our protocol can be applied to the pancake problem. As mentioned above, the upper bound of $h(n)$ is linearly suppressed by n, so we expect the required number of shuffles not to be excessively large as n increases. We also expect to find more value in keeping a sequence of swap operations secret, since the problem of finding a sequence of the minimum number of swap operations has been proven to be an NP-hard problem. This direction is a future work.

Acknowledgements

This work was supported by Grant-in-Aid for Scientific Research (JP18H05289, JP21K11881).

References

1. 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`.
2. Xavier Bultel, Jannik Dreier, Jean-Guillaume Dumas, and Pascal Lafourcade. Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen. In

[8] The upper and lower bounds were known to be $\frac{17}{16}n \leq h(n) \leq \frac{5n+5}{3}$ [7] in 1979.

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.

3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. Rikuo Haga, Yuichi Hayashi, Daiki Miyahara, and Takaaki Mizuki. Card-minimal protocols for three-input functions with standard playing cards. In *Progress in Cryptology—AFRICACRYPT 2022*, LNCS, Cham, 2022. Springer. to appear.
10. 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 Computer Security*, volume 13504 of *LNCS*, pages 224–240, Cham, 2022. Springer. URL: https://doi.org/10.1007/978-3-031-15255-9_12.
11. Yuji Hashimoto, Koji Nuida, Kazumasa Shinagawa, Masaki Inamura, and Goichiro Hanaoka. Toward finite-runtime card-based protocol for generating a hidden random permutation without fixed points. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E101.A(9):1503–1511, 2018. URL: https://doi.org/10.1587/transfun.E101.A.1503.
12. 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.
13. 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.

14. 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.
15. 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.
16. 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.
17. 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.
18. 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.
19. Donald E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 2: Generating All Tuples and Permutations (Art of Computer Programming)*. Addison-Wesley Professional, 2005.
20. Alexander Koch. *Cryptographic Protocols from Physical Assumptions*. PhD thesis, Karlsruhe Institute of Technology, 2019. URL: https://doi.org/10.5445/IR/1000097756.
21. 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.
22. Alexander Koch and Stefan Walzer. Private function evaluation with cards. *New Gener. Comput.*, pages 1–33, 2022. in press. URL: https://doi.org/10.1007/s00354-021-00149-9.
23. 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.
24. 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.*, 2021, in press. URL: https://doi.org/10.1016/j.tcs.2021.07.019.
25. 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.
26. Yoshifumi Manabe and Hibiki Ono. Card-based cryptographic protocols for three-input functions using private operations. In *Combinatorial Algorithms*, volume 12757 of *LNCS*, pages 469–484, Cham, 2021. Springer. URL: https://doi.org/10.1007/978-3-030-79987-8_33.

27. Yoshifumi Manabe and Hibiki Ono. Card-based cryptographic protocols with malicious players using private operations. *New Gener. Comput.*, pages 1–27, 2022. in press. URL: https://doi.org/10.1007/s00354-021-00148-w.
28. 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.
29. 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.
30. 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.
31. Daiki Miyahara, Tatsuya Sasaki, Takaaki Mizuki, and Hideaki Sone. Card-based physical zero-knowledge proof for Kakuro. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 102(9):1072–1078, 2019. URL: https://doi.org/10.1587/transfun.E102.A.1072.
32. 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.
33. 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.
34. 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.
35. 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.
36. Takaaki Mizuki and Hiroki Shizuya. A formalization of card-based cryptographic protocols via abstract machine. *Int. J. Inf. Secur.*, 13(1):15–23, 2014. URL: https://doi.org/10.1007/s10207-013-0219-4.
37. Takaaki Mizuki and Hiroki Shizuya. Computational model of card-based cryptographic protocols and its applications. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E100.A(1):3–11, 2017. URL: https://doi.org/10.1587/transfun.E100.A.3.
38. 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.
39. Takeshi Nakai, Satoshi Shirouchi, Yuuki Tokushige, Mitsugu Iwamoto, and Kazuo Ohta. Secure computation for threshold functions with physical cards: Power of private permutations. *New Gener. Comput.*, pages 1–19, 2022. in press. URL: https://doi.org/10.1007/s00354-022-00153-7.
40. Hibiki Ono and Yoshifumi Manabe. Card-based cryptographic logical computations using private operations. *New Gener. Comput.*, 39(1):19–40, 2021. URL: https://doi.org/10.1007/s00354-020-00113-z.
41. Andy Pepperdine. 73.23 topswops. *The Mathematical Gazette*, 73(464):131–133, 1989. URL: http://www.jstor.org/stable/3619674.

42. 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.
43. 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.
44. Léo Robert, Pascal Lafourcade, Daiki Miyahara, and Takaaki Mizuki. Card-based ZKP protocol for nurimisaki. In *Stabilization, Safety, and Security of Distributed Systems*, LNCS, Cham, 2022. Springer. to appear.
45. 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.
46. 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.
47. 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.
48. 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.
49. 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.
50. Suthee Ruangwises. Using five cards to encode each integer in Z/6Z. In *Innovative Security Solutions for Information Technology and Communications*, LNCS, Cham, 2021. Springer. to appear.
51. 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.
52. 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.
53. 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.
54. 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.

55. 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.
56. 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.
57. 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.
58. 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.
59. 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.
60. 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.
61. 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.
62. Hayato Shikata, Kodai Toyoda, Daiki Miyahara, and Takaaki Mizuki. Card-minimal protocols for symmetric boolean functions of more than seven inputs. In *Theoretical Aspects of Computing – ICTAC 2022*, LNCS, Cham, 2022. Springer. to appear.
63. Kazumasa Shinagawa. *On the Construction of Easy to Perform Card-Based Protocols*. PhD thesis, Tokyo Institute of Technology, 2020.
64. 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.