Check Alternating Patterns: A Physical Zero-Knowledge Proof for Moon-or-Sun*

Samuel Hand[1], Alexander Koch[2], Pascal Lafourcade[3], Daiki Miyahara[4,5], and Léo Robert[6]

[1] University of Glasgow, Glasgow, UK
[2] Paris Cité University, Paris, France
[3] University Clermont Auvergne, LIMOS, CNRS UMR 6158, Aubière France
`pascal.lafourcade@uca.fr`
[4] The University of Electro-Communications, Tokyo, Japan
`miyahara@uec.ac.jp`
[5] National Institute of Advanced Industrial Science and Technology, Tokyo, Japan
[6] University of Limoges, XLIM, Limoges, France

Abstract. A zero-knowledge proof (ZKP) allows a party to prove to another party that it knows some secret, such as the solution to a difficult puzzle, without revealing any information about it. We propose a physical zero-knowledge proof using only a deck of playing cards for solutions to a pencil puzzle called *Moon-or-Sun*. In this puzzle, one is given a grid of cells on which rooms, marked by thick black lines surrounding a connected set of cells, may contain a number of cells with a moon or a sun symbol. The goal is to find a loop passing through all rooms exactly once, and in each room either passes through all cells with a moon, or all cells with a sun symbol.

Finally, whenever the loop passes from one room to another, it must go through all cells with a moon if in the previous room it passed through all cells with a sun, and visa-versa. This last rule constitutes the main challenge for finding a physical zero-knowledge proof for this puzzle, as this must be verified without giving away through which borders the loop enters or leaves a given room. We design a card-based zero-knowledge proof of knowledge protocol for Moon-or-Sun solutions, together with an analysis of their properties. Our technique of verifying the alternation of a pattern along a non-disclosed path might be of independent interest for similar puzzles.

Keywords: Physical Zero-knowledge Proof · Pencil Puzzle · Card-based Cryptography · Moon-or-Sun · Nikoli Puzzle

* This paper appears in Proceedings of IWSEC 2023. 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: `https://doi.org/10.1007/978-3-031-41326-1_14`. 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 Introduction

A Zero-Knowledge Proof (ZKP) protocol is a cryptographic tool enabling a party to prove a statement without revealing information about it. Due to their versatility, numerous variants of these protocols exist with different possible applications. For instance, a ZKP could help to determine if a database contains information without revealing it. A ZKP protocol is also used for e-voting system to ensure that ballots are correctly shuffled. Lastly, ZKP protocols are also used for cryptocurrencies like Monero or ZCash to allow anonymous transactions.

We focus on a particular ZKP: interactive Zero-Knowledge Proof of Knowledge protocols. In this context, there are two parties involved: a prover P and a verifer V. The prover wishes to convince the verifier that it knows specific information without revealing it. These protocols have three properties:

- Completeness: if P knows a secret s then the protocol ends without failure (meaning that V is convinced P has s);
- (Perfect) Soundness: if P does not have the solution s then the protocol will abort (with probability 1);
- Zero-Knowledge: V learns nothing about s.

We design a ZKP protocol for the Moon-or-Sun puzzle. The goal of our protocol is that if a prover P has the solution for a given instance of the Moon-or-Sun puzzle, then it will be able to convince a verfier V of this fact, and the protocol will end (as stated by the *completeness* property). Further, any information revealed during the protocol should not leak any information about the solution (as stated by the *zero-knowledge* property). Finally, if P does not have the solution, then the protocol should abort (as stated by the *soundness* property).

In a nutshell, the protocol is done in two major steps: (1) the prover P commits its solution, and (2) the verifier V checks that the commited values are respecting the rules.

The Moon-or-Sun rules are given in Fig. 1. We also illustrate an example in Fig. 2 taken from Nikoli's webgpage[7].

In [10], the Moon-or-Sun puzzle is proven to be NP-complete which implies that a ZKP protocol exists, as proved in [2]. While the latter work is a constructive proof which implies the existence of a ZKP protocol, there is always the need to design a specific protocol for a given problem. Indeed, the generic construction is not efficient, nor interesting in itself.

[7] `https://www.nikoli.co.jp/en/puzzles/moon_or_sun/`

Moon-or-Sun Rules:

1. Construct a loop.
2. The loop never crosses itself, branches off, or goes through the same cell twice.
3. The loop goes through each *room* (*i.e.*, continuous cells delimited by thick edges) only once.
4. The loop goes through all moon or all sun for each room. This means that the loop cannot pass through moon and sun cells for a given room.
5. After the loop goes through the moons in one room it has to go through all the suns in the next room it enters and visa versa.

Fig. 1. Rules for Moon-or-Sun [1].

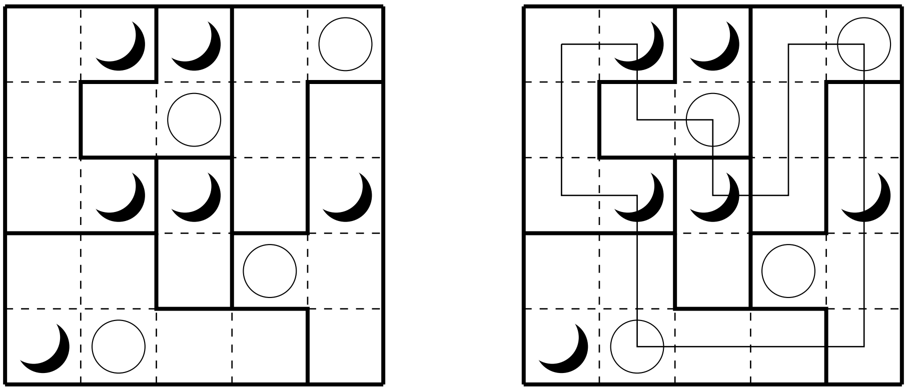


Fig. 2. Example of a Moon-or-Sun instance, with initial values on the left and the solution on the right.

Related work. The first physical ZKP protocol [8] for a Sudoku grid was constructed using a deck of cards. Since this novel protocol was devised, several papers have proposed physical ZKP protocols using a deck of cards for pencil puzzles, such as Sudoku [29,30], Akari [4], Takuzu [4], Kakuro [4,14], KenKen [4], Makaro [5], Norinori [7], Slitherlink [13], Suguru [21], Nurikabe [18], Ripple Effect [26], Numberlink [24], Bridges [25], Cryptarithmetic [9], and Nonogram [6, 23]. More recent puzzles have been considered such as Shikaku [27], Makaro (using a standard deck of cards) [28], Nurimisaki [19], Topswops [11], Pancake Sorting [12], and Usowan [20].

Contributions. We design a card-based, interactive ZKP protocol for the Moon-or-Sun puzzle. We rely on some existing techniques, such as constructing a non-branching loop or computing the sum of multiple commitments, but also propose original and simple sub-protocols, such as showing alternating pattern, to obtain a secure ZKP protocol. Our description is also accompanied of security proofs to show the completeness, perfect soundness and zero-knowledge of our protocol. An overview of our proposed protocol is depicted in Fig. 3.

We also demonstrate that our proposed ZKP protocol is related to a well-known NP-hard problem in graph theory. This may prove the significance of our protocol for a Moon-or-Sun puzzle.

Outline. We begin by introducing notations and existing protocols used in our ZKP protocol in Sect. 2. In Sect. 3, we present our ZKP protocol for a Moon-or-Sun puzzle. In Sect. 4, we prove that our ZKP protocol satisfies the ZKP properties. In Sect. 5, we discuss about our ZKP protocol. We conclude this study in Sect. 6.

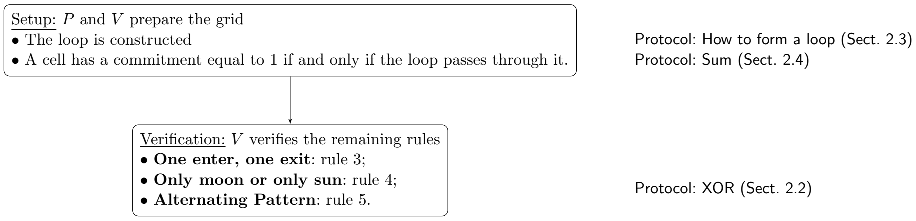

Setup: P and V prepare the grid
• The loop is constructed
• A cell has a commitment equal to 1 if and only if the loop passes through it.
Protocol: How to form a loop (Sect. 2.3)
Protocol: Sum (Sect. 2.4)
Verification: V verifies the remaining rules
• **One enter, one exit**: rule 3;
• **Only moon or only sun**: rule 4;
• **Alternating Pattern**: rule 5.
Protocol: XOR (Sect. 2.2)

Fig. 3. Overview of our protocol. On the left with rounded corners are the main steps of our construction, and on the right are the main sub-protocols used.

2 Preliminaries

We present the general notions needed for our ZKP protocol, such as encoding and sub-protocols.

Cards and Encoding. We use a deck of cards consisting of two suits: clubs $\boxed{\clubsuit}$ and hearts $\boxed{\heartsuit}$. We then let an ordered pair of these cards represent a bit value according to the following encoding:

$$\boxed{\clubsuit}\boxed{\heartsuit} \to 0, \quad \boxed{\heartsuit}\boxed{\clubsuit} \to 1. \tag{1}$$

Each card in the deck has an identical back $\boxed{?}$, and we refer to an ordered pair of face-down cards satisfying encoding (1) for a bit $x \in \{0, 1\}$

as a *commitment* to x. Such a commitment to a bit x is then denoted by:

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

We also define two converse encodings for integers modulo p [26]:

- $\clubsuit$-scheme: to encode $x \in \mathbb{Z}/p\mathbb{Z}$ use a row of p cards with one $\clubsuit$ in position $(x+1)$ from the left and the remaining $p-1$ positions occupied by $\heartsuit$s. As an example, we would represent 2 with $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}$ in $\mathbb{Z}/4\mathbb{Z}$.
- $\heartsuit$-scheme: equivalently as above, but with $\heartsuit$ and $\clubsuit$ exchanged. Here 2 is instead represented by $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}$ in $\mathbb{Z}/4\mathbb{Z}$.

2.1 Shuffle

We explain two types of shuffles that introduce randomness into the order of a sequence of cards. These shuffles are usually employed in card-based cryptography, particularly within ZKP protocols.

Consider a *pile* consisting of ℓ cards, where $\ell > 0$. Both shuffles are applied to multiple piles of cards, making the order of the piles unknown to everyone, while preserving the order of cards within each pile. Suppose that we have m piles denoted by $(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$, each containing ℓ cards.

Pile-scramble shuffle. This shuffling method, initially introduced in [16], completely randomizes the order of piles. Applying a pile-scramble shuffle to $(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$ yields $(\boldsymbol{p}_{r^{-1}(1)}, \boldsymbol{p}_{r^{-1}(2)}, \ldots, \boldsymbol{p}_{r^{-1}(m)})$, where r is a random permutation uniformly distributed in a symmetric group of degree m, denoted by S_m. This shuffling is denoted by $[\cdot|\ldots|\cdot]$.

Pile-shifting shuffle. This shuffling method, initially introduced in [31], randomly and cyclically shifts the order of piles. Applying a *pile-shifting* shuffle to $(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$ yields $(\boldsymbol{p}_{s+1}, \boldsymbol{p}_{s+2}, \ldots, \boldsymbol{p}_{s+m})$, where s is chosen randomly and uniformly from $\mathbb{Z}/m\mathbb{Z}$. This shuffling is denoted by $\langle\cdot|\ldots|\cdot\rangle$.

When $m = 2$, this shuffle is called a *random bisection cut* [17], i.e., bisecting a sequence of cards and randomly swapping the two halves. When $\ell = 1$, this shuffle is known as a *random cut* invented by Den Boer [3].

2.2 XOR and Copy Protocols

Our protocol uses the existing card-based protocols for computing a logical function of two-input XOR and duplicating an input commitment [17].

Here, we briefly introduce them, and their full descriptions are in Appendix A.

Given commitments to $a, b \in \{0, 1\}$, the Mizuki–Sone XOR protocol [17] outputs a commitment to $a \oplus b$:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\ \underbrace{\boxed{?}\boxed{?}}_{b} \rightarrow \cdots \rightarrow \underbrace{\boxed{?}\boxed{?}}_{a \oplus b}.$$

Given a commitment to $a \in \{0, 1\}$ along with two commitments to 0, the Mizuki–Sone copy protocol [17] outputs two commitments to a:

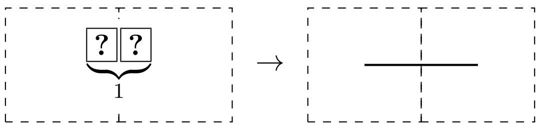
$$\underbrace{\boxed{?}\boxed{?}}_{a}\ \underbrace{\boxed{?}\boxed{?}}_{0}\ \underbrace{\boxed{?}\boxed{?}}_{0} \rightarrow \cdots \rightarrow \underbrace{\boxed{?}\boxed{?}}_{a}\ \underbrace{\boxed{?}\boxed{?}}_{a}.$$

2.3 How to Form a Loop

To verify rule 2, i.e., the loop condition, we use the existing protocol from [13]. Let us introduce an overview of the protocol [13]. This protocol enables a prover P to create any single loop without revealing information about the loop shape, while simultaneously convincing a verifier V that the resulting loop is indeed a single loop. That is, this protocol creates a figure respecting rule 2 rather than verifying it.

Briefly, this protocol starts from the single loop going along the boundary of the board. P and V interactively create the solution P has from the single loop. During this process, V cannot obtain information other than that the process proceeds correctly, and hence, the resulting shape is indeed a single loop.

To represent a loop with a sequence of cards, we place a commitment *between* each cell. The value of such a commitment represents the existence of line, *i.e.*, line passes through them if the value is 1, and no line passes if it is 0 as follows:



Thus, the protocol [13] begins by placing a commitment to 1 between each cell adjacent to the border of a given board and commitments to 0 on the remaining positions, representing the single loop. Refer to [13] for specific methods on creating the solution P has.

2.4 Sum of Commitments

This protocol is defined in [26]; we give a general description given as an example. Suppose that we have commitments to $a, b \in \{0,1\}$, and we want to output $a + b \in \mathbb{Z}/3\mathbb{Z}$ (in the $\heartsuit$-scheme, see encoding Eq. (2)):

$$\underbrace{\boxed{?}\boxed{?}}_{a}\;\underbrace{\boxed{?}\boxed{?}}_{b}\;\boxed{\clubsuit}\boxed{\heartsuit} \;\rightarrow\; \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{a+b}.$$

1. Swap the two cards of the commitment to a and add a $\boxed{\clubsuit}$ face-down to the right. Those three cards represent a in the $\heartsuit$-scheme in $\mathbb{Z}/3\mathbb{Z}$:

$$\underbrace{\overset{\rightleftarrows}{\boxed{?}\boxed{?}}}_{a}\;\underset{\clubsuit}{\boxed{?}} \;\rightarrow\; \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{a}.$$

2. Add a $\boxed{\heartsuit}$ on the right of the commitment to b. Those three cards represent b in the $\clubsuit$-scheme in $\mathbb{Z}/3\mathbb{Z}$: $\underbrace{\boxed{?}\boxed{?}}_{b}\;\underset{\heartsuit}{\boxed{?}} \rightarrow \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{b}$.
3. Obtain three cards representing $a + r$ and those representing $b - r$ for a uniformly random value $r \in \mathbb{Z}/3\mathbb{Z}$ as follows.
   (a) Place in *reverse* order the three cards obtained in Step 2 below the three cards obtained in Step 1:

$$\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{a}\;\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{b} \;\rightarrow\; \begin{array}{c}\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{a}\\ \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{2-b}\end{array}.$$

   (b) Apply a pile shifting shuffle as follows:

$$\left\langle \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \right\rangle \;\rightarrow\; \begin{array}{c}\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{a+r}\\ \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{2-b+r}\end{array}.$$

   For a uniformly random value $r \in \mathbb{Z}/3\mathbb{Z}$, we obtain three cards representing $a + r$ and $2 - b + r$.

   (c) Reverse the order of the three cards representing $2-b+r$ to obtain $b - r$: $\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{a+r}\;\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{b-r}$.

4. Reveal the three cards representing $b - r$, and shift to the right the three cards representing $a + r$ to obtain those representing $a + b$ in the ♡-scheme; apply the same routine for the remaining elements to compute the final sum.

Notice that we described the sum protocol for an output of two bit commitments in $\mathbb{Z}/3\mathbb{Z}$. We can generalize by inductively applying the protocol for n bit commitments giving an output in $\mathbb{Z}/(n+1)\mathbb{Z}$.

3 ZKP Protocol for Moon-or-Sun

We present a card-based ZKP protocol for a Moon-or-Sun puzzle. As shown in Fig. 3, our protocol has two phases: the setup and verification phases. The setup phase constructs a loop with the interaction between a prover P and a verifier V. The verification phase verifies all the rules other than rules 1 and 2.

3.1 Setup

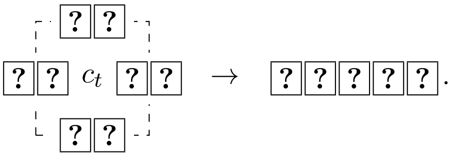
As introduced in Sect. 2.3, a solution is represented with a commitment between each adjacent cells. Moreover, we place a commitment on each cell to represent the loop passing through a symbol (*i.e.*, moon or sun symbol). The setup is done in two steps:

1. Constructing the loop using [13];
2. Placing the commitments inside the cells. Notice that we modify only cells with moon or sun symbol.

Forming the loop. We directly use the construction of [13] described in Sect. 2.3. At this point, there are commitments between the cells but no commitment inside them.

Filling the grid. We want to put commitments inside the cells to model the line passing through it (or not). For each cell (corresponding to a moon or sun symbol), if the line passes through it, we observe that the sum of the values of the four *neighbour* commitments on its edge is always equal to two (otherwise, zero). Based on this observation, we place a commitment inside every cell as follows. We note that we execute the Mizuki–Sone copy protocol introduced in Sect. 2.2 whenever a commitment on the board is taken, so that the same commitment can be used for several times.

1. Apply the sum protocol (Sect. 2.4) to the four neighbours of the targeted cell $\boxed{c_t}$. The result is in ♡-scheme:



Remember that if the sum is two, the third card from the left in the resulting sequence is a $\boxed{\heartsuit}$.

2. Make a commitment consisting of the third and first cards in the resulting sequence (in this order) by taking them and place it on $\boxed{c_t}$.

V is convinced that each cell (containing a moon or a sun) is equal to $1 = \boxed{\heartsuit}\boxed{\clubsuit}$ if and only if the line passes through it, *i.e.*, there are two neighbours equal to 1, exactly.

At this point, P has placed commitments according to its solution, and V wants to check that each rule is respected.

3.2 Verification Phases

The loop has been constructed in the previous step, so V wants to check if the other rules are respected.

Only moon or only sun (rule 4). The loop must pass through only moon or only sun symbols in a given room but exactly one of them. The following verification is done for each room:

1. Consider all the commitments on sun cells, and place them in a sequence (in any order). Apply a random cut introduced in Sect. 2.1 and reveal it. If the result has alternating pattern, then continue; otherwise, abort.
   We show an example when the room has three sun cells as follows:

$$\left\langle \boxed{?}\boxed{?}\;\boxed{?}\boxed{?}\;\boxed{?}\boxed{?} \right\rangle \rightarrow \boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit} \text{ or } \boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}.$$

2. Repeat the previous step for moon cells.
3. Execute the Mizuki–Sone XOR protocol [17] with a commitment on any sun cell and a commitment on any moon cell. If the protocol outputs a commitment to 1, then V continues; otherwise, V aborts.

Note that no information is leaked if the rule 4 is respected. Indeed, if the commitments are equal (for a given symbol), the random cut *hides* the initial values of commitments (V does not know if they are 0s or 1s). However, if the rule is not respected, then V knows the number of commitments that are different (*i.e.*, it can deduce the Hamming weight of the sequence).

One enter, one exit (rule 3). The loop must be passing through a room only once. This means that for each room, the loop crosses its edge exactly twice (one for entering and one for exiting the room). The idea is thus to shuffle the commitments located at the edge of a room and reveal them. Formally, we proceed as follows:

1. Consider a room and take all the commitments located at the edge.
2. Apply a pile-scramble shuffle to them.
3. Reveal all the commitments. If exactly two commitments to 1 appears, then continue; otherwise, abort.
4. Repeat the previous step until visiting all rooms.

Alternating pattern (rule 5). The loop must pass through a different symbol to the one in the previous room it enters. Let us first present the idea behind our verification for this rule.

Given a solution as in Fig. 2, consider verifying whether a room (referred to as the target) satisfies rule 5 or not. For this, we examine the two rooms connected to the target room (*i.e.*, those through which line passes) and ensure that the loop passes through different symbols within the target room and the connected rooms. That is, we determine whether the two connected rooms are either both "sun rooms" or "moon rooms" and both of them differ to the target room. Our approach follows a similar logic: for every adjacent room, we collect a commitment on any sun cell[8]. Subsequently, from among the commitments, we somehow choose two commitments corresponding to the two connected rooms without leaking any information. The remaining steps are simple; we confirm that the values of the chosen commitments XORed with a commitment on any sun cell within the target room both yields ones.

Now we are ready to describe the verification method. Suppose that we verify the rule 5 for a target room R_0 with $k\,(\geq 2)$ adjacent rooms, $R_1, R_2, \ldots, R_k$. Let n_i, $1 \leq i \leq k$, denote the number of commitments on the border between R_0 and R_i. The verification proceeds as follows.

1. For every adjacent room R_i, let c_j denote each of the n_i commitments on the border between R_0 and R_i, for $1 \leq j \leq n_i$ (in any order). Collect one c_j for every j and add "dummy" commitments to 0 so

[8] Remember that the value of a commitment on a cell indicates the presence of line passing through the cell.

that the total number of collected commitments (*i.e.*, n_i) becomes $n_{\max} = \max(n_1, \dots, n_k)$ as follows:

$$\overbrace{\boxed{?}\boxed{?}\,\boxed{?}\boxed{?}\,\cdots\,\boxed{?}\boxed{?}}^{n_i\text{ commitments}} \rightarrow \overbrace{\boxed{?}\boxed{?}\,\boxed{?}\boxed{?}\,\cdots\,\boxed{?}\boxed{?}}^{n_i\text{ commitments}}\,\overbrace{\underbrace{\boxed{?}\boxed{?}}_{0}\underbrace{\boxed{?}\boxed{?}}_{0}\,\cdots\,\underbrace{\boxed{?}\boxed{?}}_{0}}^{n_{\max}-n_i\text{ commitments}}.$$

Let s_i denote the sequence of the $n_{\max}$ commitments. Apply a pile-scramble shuffle to s_i as follows:

$$s_i : \left[\,\boxed{?}\boxed{?}\,\middle\|\,\boxed{?}\boxed{?}\,\middle\|\cdots\middle\|\,\boxed{?}\boxed{?}\,\right] \;\rightarrow\; s_i : \boxed{?}\boxed{?}\,\boxed{?}\boxed{?}\,\cdots\,\boxed{?}\boxed{?}.$$

2. For every adjacent room R_i, $1 \le i \le k$, let c'_i denote a commitment on any sun cell within R_i. Place one c'_i above s_i. (If R_i has no sun, then let c'_i denote a commitment on any moon cell and swap the two cards constituting c'_i before placing it.)

$$s_i : \boxed{?}\boxed{?}\,\boxed{?}\boxed{?}\,\cdots\,\boxed{?}\boxed{?} \;\rightarrow\; \begin{array}{r l} c'_i : & \boxed{?}\boxed{?} \\ s_i : & \boxed{?}\boxed{?}\,\boxed{?}\boxed{?}\,\cdots\,\boxed{?}\boxed{?}. \end{array}$$

3. Apply a pile-scramble shuffle to the n_k piles consisting of s_i and c'_i, $1 \le i \le k$, as follows:

$$\left[\begin{array}{r l} c'_1 : & \boxed{?}\boxed{?} \\ s_1 : & \boxed{?}\boxed{?}\,\boxed{?}\boxed{?}\,\cdots\,\boxed{?}\boxed{?} \end{array}\middle\|\cdots\middle\|\begin{array}{r l} c'_k : & \boxed{?}\boxed{?} \\ s_k : & \boxed{?}\boxed{?}\,\boxed{?}\boxed{?}\,\cdots\,\boxed{?}\boxed{?} \end{array}\right].$$

4. Reveal all the commitments constituting s_i for all i, $1 \le i \le k$. Then exactly two commitments to 1 should be revealed, each appearing in different s_i (rule 3)[9]. Denote these positions as a and b ($a < b$) as follows:

$$\cdots \begin{array}{r l} c'_{r^{-1}(a)} : & \boxed{?}\boxed{?} \\ s_{r^{-1}(a)} : & \boxed{\clubsuit}\boxed{\heartsuit}\cdots\boxed{\heartsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}\boxed{\heartsuit} \end{array} \cdots$$

$$\cdots \begin{array}{r l} c'_{r^{-1}(b)} : & \boxed{?}\boxed{?} \\ s_{r^{-1}(b)} : & \boxed{\clubsuit}\boxed{\heartsuit}\cdots\boxed{\heartsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}\boxed{\heartsuit} \end{array} \cdots,$$

where $r \in S_k$ is a random permutation generated through the application of a pile-scramble shuffle at step 3.

[9] This means that rule 3 can be simultaneously verified for the target room.

5. Let c'_0 denote a commitment on any sun cell in the target room R_0. (If there is no sun, then denote a commitment on any moon cell by c'_0 and swap the two cards constituting c'_0.) Execute an extended version of the Mizuki–Sone XOR protocol [17] with c'_0, $c'_{r^{-1}(a)}$, and $c'_{r^{-1}(b)}$ as follows.
   (a) Place c'_0, $c'_{r^{-1}(a)}$, and $c'_{r^{-1}(b)}$ and apply a random bisection cut as follows:

$$\begin{array}{rl} c'_0 & : \boxed{?}\boxed{?} \\ c'_{r^{-1}(a)} & : \boxed{?}\boxed{?} \\ c'_{r^{-1}(b)} & : \boxed{?}\boxed{?} \end{array} \rightarrow \left[\begin{array}{c||c} \boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?} \end{array}\right] \rightarrow \begin{array}{c} \boxed{?}\boxed{?} \\ \boxed{?}\boxed{?} \\ \boxed{?}\boxed{?} \end{array}.$$

   (b) Reveal all the cards. If the values of the middle and bottommost commitments both differ from the value of the topmost commitment, then continue; otherwise, abort.

$$\begin{array}{c} \boxed{?}\boxed{?} \\ \boxed{?}\boxed{?} \\ \boxed{?}\boxed{?} \end{array} \rightarrow \begin{array}{c} \boxed{\clubsuit}\boxed{\heartsuit} \\ \boxed{\heartsuit}\boxed{\clubsuit} \\ \boxed{\heartsuit}\boxed{\clubsuit} \end{array} \text{ or } \begin{array}{c} \boxed{\heartsuit}\boxed{\clubsuit} \\ \boxed{\clubsuit}\boxed{\heartsuit} \\ \boxed{\clubsuit}\boxed{\heartsuit} \end{array} \rightarrow \text{Continue.}$$

We execute these steps for all the rooms. If V does not abort, then V is convinced that the commitments on the board respect rule 5. We discuss on reducing the number of executions of these steps in Sect. 5.

3.3 Efficiency

Let us evaluate the number of required shuffles for our proposed ZKP protocol for efficiency. Because the verification for rule 5 (alternating pattern) can also verify rule 3 (one enter, one exit) as mentioned, our protocol does not execute the verification for rule 3 in this evaluation. Also, let us omit the evaluation of the part of constructing the single loop [13] because it cannot be included in this paper due to the page length limit.

Let n_r denote the number of rooms in a given Moon-or-Sun puzzle and $p \times q$ denote the size of the puzzle. For verifying rule 4, our protocol uses two random cuts and one random bisection cut (for the XOR protocol [17]) for each room, *i.e.*, $3n_r$. For verifying rule 5 for each room, our protocol uses one pile-scramble shuffle, one random bisection cut, and a number of pile-scramble shuffles corresponding to the number of adjacent rooms. For duplicating commitments, our protocol applies the copy protocol [17], *i.e.*, one random bisection cut, to each commitment between each pair of adjacent rooms and on moon and sun cells. For making a commitment placed on each of moon and sun cells, our protocol applies the sum protocol [26] to the four neighbour commitments, *i.e.*, three pile-shifting shuffles. In total, because the number of commitments between each pair of adjacent rooms (and on moon and sun cells) is less than $p^2 \times q^2$, our protocol uses $\mathcal{O}(p^2q^2)$ shuffles.

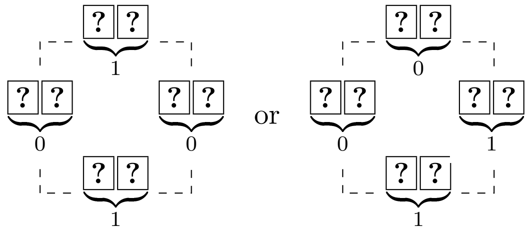
4 Security Proofs

Our protocol needs to verify three security properties given as theorems.

Theorem 1 (Completeness). *If P knows a solution of a Moon-or-Sun grid, then P can always convince V (*i.e.*, V does not abort).*

Proof. In the setup phase described in Sect. 3.1, constructing the loop is directly taken from [13], so we refer readers to this paper for the proof.

For placing a commitment inside a cell, we use the sum protocol [26] so that the value of the commitment represents the presence of line. Because the configuration of the four neighbors is either the following two (up to rotation), the resulting sequence at step 1 is always ♣♣♡♣♣, representing two if line passes through the cell (the loop never branches off):



If line does not pass, then the resulting sequence is ♡♣♣♣♣ because all the four neighbors are commitments to 0. Therefore, constructing a commitment with the first and third cards, from the previous sequence, correctly represents the presence of line for a cell.

For the verification phase, described in Sect. 3.2, we divide the proof into three parts, each corresponding to one rule.

- **Only moon or only sun (rule 4):** Because P knows a solution, the values of all the commitments on sun cells considered at step 1 are either 0s or 1s, *i.e.*, ♣♡ ♣♡ ··· or ♡♣ ♡♣ ···. Thus, applying a random cut to them always yields a sequence having an alternating pattern. This holds true for moon cells at step 2 as well. Finally, because rule 4 implies that the value of a commitment on any sun cell must differ to that on any moon cell within the same room, the XOR protocol [17] always outputs a commitment to 1 at step 3.

One enter, one exit (rule 3): The number of commitments to 1 among all the commitments located at the edge must be two for every room. Therefore, two commitments to 1 always appear when revealing all of them at step 3.

Alternating pattern (rule 5): At step 5, $c'_{r^{-1}(a)}$ and $c'_{r^{-1}(b)}$ come from commitments on any sun cell within different rooms such that lines exist between each of them and R_0. This is because a commitment to 1 is revealed among each of $s_{r^{-1}(a)}$ and $s_{r^{-1}(b)}$, and s_i comes from commitments on the border between R_0 and R_i. Rule 5 implies that the values of $c'_{r^{-1}(a)}$ and $c'_{r^{-1}(b)}$ must be both different to the value of c'_0, *i.e.*, the first configuration at step 5(a) is as follows:

$$\begin{array}{rl} c'_0 & : \boxed{\heartsuit}\boxed{\clubsuit} \\ c'_{r^{-1}(a)} & : \boxed{\clubsuit}\boxed{\heartsuit} \\ c'_{r^{-1}(b)} & : \boxed{\clubsuit}\boxed{\heartsuit} \end{array} \quad \text{or} \quad \begin{array}{rl} c'_0 & : \boxed{\clubsuit}\boxed{\heartsuit} \\ c'_{r^{-1}(a)} & : \boxed{\heartsuit}\boxed{\clubsuit} \\ c'_{r^{-1}(b)} & : \boxed{\heartsuit}\boxed{\clubsuit} \end{array}.$$

Thus, V never aborts when revealing all the cards at step 5(b). □

Theorem 2 (Soundness). *If P does not know a solution of a Moon-or-Sun grid, then V always rejects (*i.e.*, the protocol aborts).*

Proof. Our protocol is a proof-of-knowledge because commitments placed on the grid after the setup phase represent a solution. Thus, in the remaining part of this proof, we prove that V always aborts if P does not provide a solution, *i.e.*, at least one rule is not respected. Because rule 2 is always respected due to [13], we consider the case that each of the remaining rules is not respected as follows.

Only moon or only sun (rule 4): For a given room, two cases are considered: (1) the loop does not pass through all suns (or moons) but only some of them, and (2) the loop passes through all moons and suns (or nothing). The first case can be detected at either step 1 or step 2 because a sequence of commitments on all sun (moon) cells does not have an alternating pattern, *e.g.*, $\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit}$. The second case can be detected at step 3 because the value of commitment on any sun cell is the same as on any moon cell.

One enter, one exit (rule 3): Because all commitments located at the edge of a given room and the target room are revealed at step 3, the number of times the loop enters the given room is revealed. Thus, V always detect the case in which rule 3 is violated.

Alternating pattern (rule 5): If this rule is violated, it means that for a given room, there is at least one adjacent room such that line exists between them but the loop passes through the same symbol (assuming that rule 4 is respected). As stated in the above proof, because $c'_{r^{-1}(a)}$ and $c'_{r^{-1}(b)}$ come from commitments on any sun cell within such rooms at step 5, V learns whether the values of them are equal to c'_0 using the XOR protocol [17]. Thus, V always aborts.

In any case, the verifier always rejects. □

Theorem 3 (Zero-knowledge). *V learns nothing about P's solution of the given grid G.*

Proof. We use the same proof technique as in [8], namely the description of an efficient *simulator* which simulates the interaction between an honest prover and a cheating verifier. As described in [8], this simulator does not have a correct solution, but has an ability that a sequence of cards can be swapped with the same number of cards during the application of shuffling; this ability is the replaced one with the rewind ability in cryptographic ZKP protocols.

Informally, our protocol is zero-knowledge because it applies an appropriate shuffling to a sequence of cards before revealing them. The simulator can always swap the sequence such that the real and simulated protocols are indistinguishable.

Formally, in the setup phase, the simulator first constructs arbitrary loop executing [13]. Subsequently, it applies the sum protocol [26] introduced in Sect. 2.4. Note that this existing protocol [26] is proved to be zero-knowledge. In the verification phase, for each of the remaining rules, it acts as follows.

Only moon or only sun (rule 4): At steps 1 and 2, during each application of a random cut, the simulator swaps the commitments with commitments to 1. Because a random cut cyclically and randomly shifts a sequence of cards, this swapping results in any of the alternating patterns with a probability of 1/2, which is indistinguishable from a real execution. At step 3, it executes the Mizuki–Sone XOR protocol [17] that is zero-knowledge.

One enter, one exit (rule 3): At step 2, the simulator swaps the commitments with the ones where the number of commitments to 1 is exactly two. Because a pile-scramble shuffle randomly rearranges the order of piles consisting cards, the two commitments to 1 appear in random positions.

Alternating pattern (rule 5): At step 1, the simulator swaps the $n_{\max}$ commitments with the ones having exactly one commitment to 1 if $i = 1, 2$ and with $n_{\max}$ commitments to 0 otherwise. At step 3, it acts nothing, but applying pile-scramble shuffles results in the case where the two commitments to 1 appears in different sequences of random positions. Finally, at step 5, it swaps c'_0, $c'_{r^{-1}(a)}$, and $c'_{r^{-1}(b)}$ with commitments to 1, 0, and 0, respectively. Because applying a random bisection cut to them results in either commitments to 1, 0, and 0 or commitments to 0, 1, and 1 with a probability of $1/2$, V learns nothing other than that the value of c'_0 differs to those of $c'_{r^{-1}(a)}$ and $c'_{r^{-1}(b)}$. □

5 Discussion

Here, we discuss whether we can reduce the number of executions of our method for rule 5 described in Sect. 3.2. Suppose that we execute the verification phase described in Sect. 3.2 for all rooms surrounding a given room. Then we prove that such a room does not need to be verified for rule 5 as in the following theorem.

Theorem 4. *A room always satisfies rule 5 if all rooms surrounding the room satisfy all the rules.*

Proof. Suppose, for the sake of contradiction, that there exists a room R that does not satisfy rule 5, while all rooms surrounding R are verified to satisfy all the rules through the execution of our verification phase described in Sect. 3.2. Then, as R does not satisfy rule 5, there should exist a room R' such that the line passes between R and R', passing through the same symbol in both.

However, R' surrounds R, and this contradicts our assumption that all rooms surrounding R satisfy all the rules. Therefore, our initial assumption must be false, and hence, R satisfies rule 5. □

Theorem 4 implies that we do not need to verify all rooms for rule 5. We observe that optimally reducing the number of rooms for which rule 5 is verified in our protocol is related to one of the classical NP-hard problems, namely, the *minimum vertex cover problem.* This connection emerges if we consider a Moon-or-Sun puzzle as an undirected graph, wherein a vertex set comprises rooms, and an edge denotes the adjacency of rooms. A vertex cover of graph is a set of vertices where every edge of the graph has at least one vertex in the set. The minimum vertex cover problem asks the minimum size of such vertex covers if they exist.

Because a vertex cover represents rooms surrounding all the remaining rooms, it suffices to verify whether such rooms satisfy rule 5, as indicated in Theorem 4. However, if we wish to verify rule 5 for a minimum number of rooms we must initially find a minimum vertex cover. As mentioned, finding such a cover is an NP-hard problem, even on planar graphs, thus we are unable to perform this initial step efficently. Although techniques for evaluating the execution time of card-based protocols exist [15], doing so in this case is non-trivial, due to this additional computationally intensive step. Additionally, constructing ZKP protocols for a Moon-or-Sun puzzle may prove more challenging than those in existing work because in essence, rule 5 involves not verifying a given room itself but comparing a given room with all of its adjacent rooms.

It is still possible to bound the number of rooms that we must verify rule 5 for, without requiring a computational step of infeasible running time. To begin, we note that any planar graph always has a vertex cover with at most $\frac{3n}{4}$ vertices, and thus we have the following theorem:

Theorem 5. *It is possible to convince the verifier that all rooms satisfy rule 5 by checking this rule for at most $\frac{3}{4}$ of the rooms.*

Proof. It follows from theorem 4 that it suffices to verify rule 5 only for rooms in a vertex cover. Furthermore every planar graph has a vertex cover containing at most $\frac{3}{4}$ of the vertices. Thus it is only ever necessary to verify rule 5 for only $\frac{3}{4}$ of the rooms. □

Furthermore, we know that we can find a cover of this size in polynomial time. To do so, we find a four coloring of the graph, and then take our cover to be the union of the three smallest color classes, yielding a cover that is of size most $\frac{3n}{4}$. It is possible to compute a four coloring for a planar graph in polynomial (quadratic) time [22].

6 Conclusion

We proposed a ZKP protocol for Moon-or-Sun, which has an interesting rule: the loop must pass through different symbols within two consecutive rooms. Through the construction, we found this rule to be related to a well-known problem in graph theory, which leads some challenging problems.

Acknowledgements. We thank the anonymous referees, whose comments have helped us improve the presentation of the paper. The fourth author was supported in part by Kayamori Foundation of Informational Science Advancement and JSPS KAKENHI Grant Number JP23H00479. The third and fifth authors were partially supported by the French ANR project ANR-18-CE39-0019 (MobiS5). Other programs also fund to write this paper, namely the French government research program "Investissements d´Avenir" through the IDEX-ISITE initiative 16-IDEX-0001 (CAP 20-25) and the IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01). Finally, the French ANR project DECRYPT (ANR-18-CE39-0007) and SEVERITAS (ANR-20-CE39-0009) also subsidize this work.

References

1. https://www.nikoli.co.jp/en/puzzles/moon_or_sun/, Nikoli, Moon-or-Sun.
2. Ben-Or, M., Goldreich, O., Goldwasser, S., Håstad, J., Kilian, J., Micali, S., Rogaway, P.: Everything provable is provable in zero-knowledge. In: Goldwasser, S. (ed.) Advances in Cryptology—CRYPTO'88. LNCS, vol. 403, pp. 37–56. Springer (1988)
3. den Boer, B.: More efficient match-making and satisfiability: *The Five Card Trick*. In: Quisquater, J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 208–217. Springer, Berlin, Heidelberg (1989)
4. Bultel, X., Dreier, J., Dumas, J., Lafourcade, P.: Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen. In: Fun with Algorithms. LIPIcs, vol. 49, pp. 8:1–8:20. Schloss Dagstuhl, Dagstuhl (2016)
5. Bultel, X., Dreier, J., Dumas, J., Lafourcade, P., Miyahara, D., Mizuki, T., Nagao, A., Sasaki, T., Shinagawa, K., Sone, H.: Physical zero-knowledge proof for Makaro. In: SSS 2018. LNCS, vol. 11201, pp. 111–125. Springer, Cham (2018)
6. Chien, Y.F., Hon, W.K.: Cryptographic and physical zero-knowledge proof: From Sudoku to Nonogram. In: Boldi, P., Gargano, L. (eds.) Fun with Algorithms. LNCS, vol. 6099, pp. 102–112. Springer, Berlin, Heidelberg (2010)
7. Dumas, J.G., Lafourcade, P., Miyahara, D., Mizuki, T., Sasaki, T., Sone, H.: Interactive physical zero-knowledge proof for Norinori. In: Du, D.Z., Duan, Z., Tian, C. (eds.) COCOON 2019. LNCS, vol. 11653, pp. 166–177. Springer, Cham (2019)
8. Gradwohl, R., Naor, M., Pinkas, B., Rothblum, G.N.: Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. Theory Comput. Syst. **44**(2), 245–268 (2009)
9. Isuzugawa, R., Miyahara, D., Mizuki, T.: Zero-knowledge proof protocol for Cryptarithmetic using dihedral cards. In: Kostitsyna, I., Orponen, P. (eds.) UCNC 2021. LNCS, vol. 12984, pp. 51–67. Springer, Cham (2021)
10. Iwamoto, C., Ide, T.: Moon-or-Sun, Nagareru, and Nurimeizu are NP-complete. IEICE Trans. Fundamentals **105**(9), 1187–1194 (2022)
11. Komano, Y., Mizuki, T.: Physical zero-knowledge proof protocol for Topswops. In: Su, C., Gritzalis, D., Piuri, V. (eds.) Information Security Practice and Experience. LNCS, vol. 13620, pp. 537–553. Springer (2022)
12. Komano, Y., Mizuki, T.: Card-based zero-knowledge proof protocol for Pancake Sorting. In: Bella, G., Doinea, M., Janicke, H. (eds.) SecITC. LNCS, vol. 13809, pp. 222–239. Springer (2023)

13. Lafourcade, P., Miyahara, D., Mizuki, T., Robert, L., Sasaki, T., Sone, H.: How to construct physical zero-knowledge proofs for puzzles with a "single loop" condition. Theor. Comput. Sci. **888**, 41–55 (2021)
14. Miyahara, D., Sasaki, T., Mizuki, T., Sone, H.: Card-based physical zero-knowledge proof for Kakuro. IEICE Trans. Fundamentals **102-A**(9), 1072–1078 (2019)
15. Miyahara, D., Ueda, I., Hayashi, Y., Mizuki, T., Sone, H.: Evaluating card-based protocols in terms of execution time. Int. J. Inf. Secur. **20**, 729–740 (2021)
16. Mizuki, T., Asiedu, I.K., Sone, H.: Voting with a logarithmic number of cards. In: Mauri, G., Dennunzio, A., Manzoni, L., Porreca, A.E. (eds.) Unconventional Computation and Natural Computation. LNCS, vol. 7956, pp. 162–173. Springer (2013)
17. Mizuki, T., Sone, H.: Six-card secure AND and four-card secure XOR. In: Deng, X., Hopcroft, J.E., Xue, J. (eds.) FAW 2009. LNCS, vol. 5598, pp. 358–369. Springer, Berlin, Heidelberg (2009)
18. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Card-based ZKP for connectivity: Applications to Nurikabe, Hitori, and Heyawake. New Gener. Comput. **40**, 149–171 (2022)
19. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Card-based ZKP protocol for Nurimisaki. In: Devismes, S., Petit, F., Altisen, K., Luna, G.A.D., Anta, A.F. (eds.) Stabilization, Safety, and Security of Distributed Systems. LNCS, vol. 13751, pp. 285–298. Springer (2022)
20. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Hide a liar: Card-based ZKP protocol for Usowan. In: Du, D., Du, D., Wu, C., Xu, D. (eds.) Theory and Applications of Models of Computation. vol. 13571, pp. 201–217. Springer (2022)
21. Robert, L., Miyahara, D., Lafourcade, P., Libralesso, L., Mizuki, T.: Physical zero-knowledge proof and NP-completeness proof of Suguru puzzle. Inf. Comput. **285**, 1–14 (2022)
22. Robertson, N., Sanders, D.P., Seymour, P.D., Thomas, R.: Efficiently four-coloring planar graphs. In: Miller, G.L. (ed.) ACM Symposium on the Theory of Computing. pp. 571–575. ACM (1996)
23. Ruangwises, S.: An improved physical ZKP for Nonogram. In: Du, D.Z., Du, D., Wu, C., Xu, D. (eds.) COCOA 2021. LNCS, vol. 13135, pp. 262–272. Cham (2021)
24. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for Numberlink puzzle and k vertex-disjoint paths problem. New Gener. Comput. **39**(1), 3–17 (2021)
25. Ruangwises, S., Itoh, T.: Physical ZKP for connected spanning subgraph: Applications to Bridges puzzle and other problems. In: Kostitsyna, I., Orponen, P. (eds.) UCNC 2021. LNCS, vol. 12984, pp. 149–163. Springer, Cham (2021)
26. Ruangwises, S., Itoh, T.: Securely computing the n-variable equality function with 2n cards. Theor. Comput. Sci. **887**, 99–110 (2021)
27. Ruangwises, S., Itoh, T.: How to physically verify a rectangle in a grid: A physical ZKP for Shikaku. In: Fraigniaud, P., Uno, Y. (eds.) Fun with Algorithms. LIPIcs, vol. 226, pp. 24:1–24:12. Schloss Dagstuhl (2022)
28. Ruangwises, S., Itoh, T.: Physical ZKP for Makaro using a standard deck of cards. In: Du, D., Du, D., Wu, C., Xu, D. (eds.) Theory and Applications of Models of Computation. LNCS, vol. 13571, pp. 43–54. Springer (2022)
29. Ruangwises, S., Itoh, T.: Two standard decks of playing cards are sufficient for a ZKP for Sudoku. New Gener. Comput. **40**(1), 49–65 (2022)
30. Sasaki, T., Miyahara, D., Mizuki, T., Sone, H.: Efficient card-based zero-knowledge proof for Sudoku. Theor. Comput. Sci. **839**, 135–142 (2020)

31. Shinagawa, K., Mizuki, T., Schuldt, J.C.N., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Card-based protocols using regular polygon cards. IEICE Trans. Fundamentals **100-A**(9), 1900–1909 (2017)

A Full Description of XOR and Copy Protocols

XOR protocol. Given commitments to $a, b \in \{0, 1\}$, the Mizuki–Sone XOR protocol [17] outputs a commitment to $a \oplus b$:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\;\underbrace{\boxed{?}\boxed{?}}_{b} \rightarrow \cdots \rightarrow \underbrace{\boxed{?}\boxed{?}}_{a\oplus b}.$$

This protocol proceeds as follows.

1. Rearrange the sequence: $\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}} \rightarrow \overset{1}{\boxed{?}}\overset{3}{\boxed{?}}\overset{2}{\boxed{?}}\overset{4}{\boxed{?}}$.
2. Apply a random bisection cut: $\left[\boxed{?}\boxed{?}\,\middle|\,\boxed{?}\boxed{?}\right] \rightarrow \boxed{?}\boxed{?}\boxed{?}\boxed{?}$.
3. Rearrange the sequence: $\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}} \rightarrow \overset{1}{\boxed{?}}\overset{3}{\boxed{?}}\overset{2}{\boxed{?}}\overset{4}{\boxed{?}}$.
4. Reveal the first and second cards in the sequence to obtain the output commitment as follows: $\boxed{\clubsuit}\boxed{\heartsuit}\,\underbrace{\boxed{?}\boxed{?}}_{a\oplus b}$ or $\boxed{\clubsuit}\boxed{\heartsuit}\,\underbrace{\boxed{?}\boxed{?}}_{\overline{a\oplus b}}$.

Copy protocol. Given a commitment to $a \in \{0, 1\}$ along with two commitments to 0, the Mizuki–Sone copy protocol [17] outputs two commitments to a:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\;\underbrace{\boxed{?}\boxed{?}}_{0}\;\underbrace{\boxed{?}\boxed{?}}_{0} \rightarrow \cdots \rightarrow \underbrace{\boxed{?}\boxed{?}}_{a}\;\underbrace{\boxed{?}\boxed{?}}_{a}.$$

This protocol proceeds as follows.

1. Rearrange the sequence as follows:
$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{5}{\boxed{?}}\overset{6}{\boxed{?}} \rightarrow \overset{1}{\boxed{?}}\overset{3}{\boxed{?}}\overset{5}{\boxed{?}}\overset{2}{\boxed{?}}\overset{4}{\boxed{?}}\overset{6}{\boxed{?}}.$$
2. Apply a random bisection cut to the sequence as follows:
$$\left[\boxed{?}\boxed{?}\boxed{?}\,\middle|\,\boxed{?}\boxed{?}\boxed{?}\right] \rightarrow \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}.$$
3. Rearrange the sequence as follows:
$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{5}{\boxed{?}}\overset{6}{\boxed{?}} \rightarrow \overset{1}{\boxed{?}}\overset{4}{\boxed{?}}\overset{2}{\boxed{?}}\overset{5}{\boxed{?}}\overset{3}{\boxed{?}}\overset{6}{\boxed{?}}.$$
4. Reveal the first and second cards in the sequence to obtain the output commitments as follows:
$$\boxed{\clubsuit}\boxed{\heartsuit}\,\underbrace{\boxed{?}\boxed{?}}_{a}\,\underbrace{\boxed{?}\boxed{?}}_{a} \text{ or } \boxed{\heartsuit}\boxed{\clubsuit}\,\underbrace{\boxed{?}\boxed{?}}_{\overline{a}}\,\underbrace{\boxed{?}\boxed{?}}_{\overline{a}}.$$