Efficient Card-Based ZKP for Single Loop Condition and Its Application to Moon-or-Sun

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

[1]University of Glasgow, Glasgow, UK.
[2]CNRS/IRIF, Université Paris Cité, Paris, France.
[3]LIMOS, University Clermont Auvergne, CNRS UMR 6158, Aubière, Flance.
[4*]The University of Electro-Communications, Tokyo, Japan.
[5*]National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan.
[6]University of Picardie Jules Verne, Amiens, France.

*Corresponding author(s). E-mail(s): miyahara@uec.ac.jp;
Contributing authors: s.hand.1@research.gla.ac.uk; koch@irif.fr;
pascal.lafourcade@uca.fr; leo.robert@u-picardie.fr;

Abstract

A zero-knowledge proof (ZKP) allows a prover to prove to a verifier that it knows some secret, such as a solution to a difficult puzzle, without revealing any information about it. In recent years, ZKP protocols using only a deck of playing cards for solutions to various pencil puzzles have been proposed. The previous work of Lafourcade et al. deals with a famous puzzle called Slitherlink. Their proposed protocol can verify that a solution forms a single loop without revealing anything about the solution, except this fact. Their protocol guarantees that the solution satisfies the single-loop condition, by interactively constructing a solution starting from a state that holds a simple single loop, and proceeding via steps that preserve the invariant of encoding a single loop, until the proper solution is reached. A drawback of their protocol is that it requires additional verifications to guarantee a single loop. In this study, we propose a more efficient ZKP protocol for such a puzzle with fewer additional verifications. For this, we employ the previous work of Robert et al., which addressed the connectivity property in a puzzle. That is, we verify that a solution is connected but not split, to be a single loop. Applying our proposal, we construct a card-based ZKP

protocol for Moon-or-Sun, which has its specific rule of alternating pattern in addition to the single-loop condition.

Keywords: Physical zero-knowledge proof, Pencil puzzle, Card-based cryptography, Moon-or-Sun, Nikoli puzzle

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 [39]. Lastly, ZKP protocols are also used for cryptocurrencies like Monero or ZCash to allow anonymous transactions.

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

- Completeness: if the statement is true, then P who knows a secret s can convince V that P knows s without failure;
- (Perfect) Soundness: any party in the prover role cannot convince V, if there is no solution;
- Zero-Knowledge: any party in the role of the verifier learns nothing about s. That is, a probabilistic polynomial-time simulator that does not know s exists, such that the outputs of both the protocol and the simulator follow the same probability distribution.

In this paper, we study ZKP protocols using only a deck of playing cards. In recent years, many theoretical studies on proposing such physical ZKP protocols for solutions to pencil puzzles have been addressed. For the most famous puzzle called Sudoku, various efficient ZKP protocols [38, 40, 46] have been proposed after the appearance of the first physical ZKP protocol by Gradwohl *et al.* [8]. In this line of research, the previous work of Lafourcade *et al.* [16] deals with a famous puzzle called Slitherlink. Given lattice dots, the goal of a Slitherlink puzzle is to connect adjacent dots to draw a *single loop* satisfying other rules. Hereinafter, we call this rule the loop condition. Their proposed protocol can verify that a solution P knows forms a single loop without revealing anything other than it. The novelty of their protocol is that it does not first make P place a solution on a given board and make V verify it but makes P interactively create a solution whose form is guaranteed to be a single loop. This idea has been applied to ZKP protocols for other puzzles, such as Nurikabe [24], Nurimisaki [27], and Usowan [26]. However, a possible drawback of the protocol in [16] is that it requires additional verifications to guarantee a single loop when creating it

as discussed in Sect. 3.1. Therefore, there is room for improvement to have a better ZKP protocol for the loop condition.

Contributions. We design a more efficient ZKP protocol for puzzles having the loop condition with fewer additional verifications. For this, we employ the previous work of Robert *et al.* [24], which addressed the connectivity property in a puzzle. Their protocol is an interactive protocol that can verify that given a grid, colored cells are connected to form a continuous block. We employ this existing protocol to verify that lines of a solution P knows is connected but not split, to be a single loop.

Applying our proposal, we construct a card-based ZKP protocol for *Moon-or-Sun*, which has its specific rule of alternating pattern in addition to the loop condition. We rely on some existing techniques, such as computing the sum of multiple commitments [35, 41, 42], but also propose original and simple sub-protocols, such as showing alternating pattern, to obtain a ZKP protocol. Our description is also accompanied by security proofs to show the completeness, perfect soundness, and zero-knowledge of our protocol. 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.

This study is an extended version of a published conference paper [9]; we construct a new card-based protocol for puzzles with the loop condition, which is more efficient than the existing protocol [16]. More specifically, Sect. 3 contains new research results.

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

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 cells 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 vice versa.

Fig. 1: Rules for Moon-or-Sun.

In [12], the Moon-or-Sun puzzle is proven to be NP-complete, which implies that a ZKP protocol exists because any problem in NP has a ZKP protocol [7]. Note that the result of [7] is based on the Turing machines, but it also implies the existence of card-based ZKP protocol because any problem in NP can be reduced to a SAT problem in polynomial time, and any Boolean circuit can be computed with a deck

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

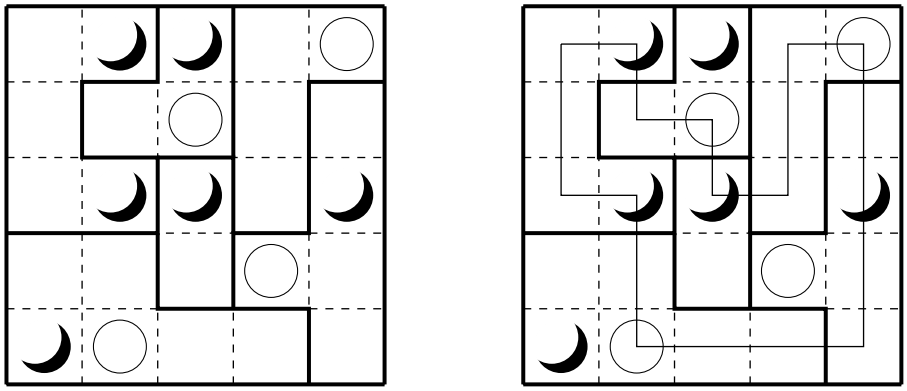


Fig. 2: Example of a Moon-or-Sun instance, with initial values on the left and the solution on the right.

of cards without leaking any information about private inputs [22]. While [7] showed a constructive proof that 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 in terms of the number of cards required due to the overhead caused by the reduction[2] nor interesting in itself.

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 [38, 40, 46], Akari [2], Takuzu [2], Kakuro [2, 17], KenKen [2], Makaro [3], Norinori [5], Slitherlink [16], Suguru [23], Nurikabe [24], Ripple Effect [35], Numberlink [33], Bridges [34], Cryptarithmetic [11], and Nonogram [4, 29]. More recent puzzles have been considered such as Shikaku [36], Makaro (using a standard deck of cards) [37], Nurimisaki [25], Topswops [14], Pancake Sorting [15], Usowan [26], ABC End View [6, 32], Ball Sort [30], Goishi Hiroi [32], Five Cells [31], 15-puzzle [45], and Sumplete [10].

Outline. We begin by introducing notations and existing protocols used in our ZKP protocol in Sect. 2. In Sect. 3, we design our card-based protocol for puzzles having the loop condition. In Sect. 4, we present our ZKP protocol for a Moon-or-Sun puzzle. In Sect. 5, we prove that our ZKP protocol satisfies the required properties. In Sect. 6, we discuss our ZKP protocol. We conclude this study in Sect. 7.

2 Preliminaries

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

[2]The number of shuffles required can be one (but complicated) regardless of the size of a given Boolean circuit [43], and hence, there is a tradeoff between such a generic construction and specific protocols. We note that a reduction from any Moon-or-Sun puzzle to a SAT problem is not addressed, because the NP-hardness proof in [12] shows the reduction from any Hamiltonian cycle problem to a Moon-or-Sun puzzle.

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 [35]:

- $\clubsuit$-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 [21], 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 the symmetric group of degree m, denoted by S_m. This shuffling is denoted by $[\cdot||\ldots||\cdot]$.

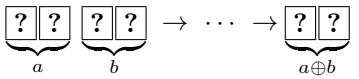
Pile-shifting shuffle. This shuffling method, initially introduced in [44], randomly and cyclically shifts the order of piles. Applying a *pile-shifting* shuffle to $(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$ yields $(\boldsymbol{p}_{s+1}, \boldsymbol{p}_{s+2}, \ldots, \boldsymbol{p}_{s+m})$, where s is chosen randomly and uniformly from $\{1, 2, \ldots, m\}$ and $\boldsymbol{p}_k$ means $\boldsymbol{p}_{k-m}$ for $k > m$. This shuffling is denoted by $\langle\cdot||\ldots||\cdot\rangle$.

When $m = 2$, this shuffle is called a *random bisection cut* [20], *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 [1].

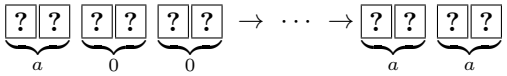
2.2 XOR and Copy Protocols

Our protocol uses the existing card-based protocols for computing a logical function of two-input XOR and duplicating an input commitment [20]. Here, we briefly introduce them, and their full descriptions are in Appx. A.

Given commitments to $a, b \in \{0, 1\}$, the Mizuki–Sone XOR protocol [20] outputs a commitment to $a \oplus b$:



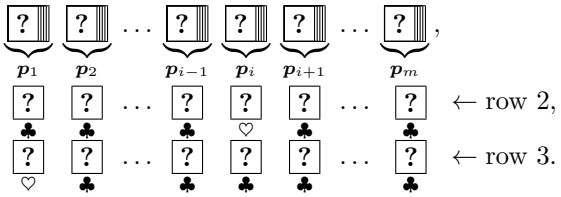
Given a commitment to $a \in \{0, 1\}$ along with two commitments to 0, the Mizuki–Sone copy protocol [20] outputs two commitments to a:



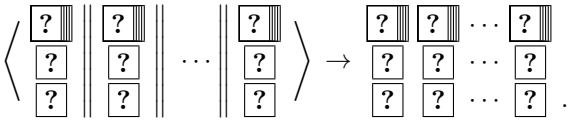
2.3 Chosen Pile Protocol [5]

This protocol is an extended version of the "chosen pile cut" proposed in [13] and used in various card-based ZKP protocols. Given m piles $(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$ with $2m$ additional cards, the *chosen pile protocol* consists of two rounds; first, it allows P and V to obtain the i-th pile $\boldsymbol{p}_i$ previously chosen by P without revealing the index i. Second, after performing some operations on the chosen pile $\boldsymbol{p}_i$, it reverts the order of the sequence of m piles to their original order.

1. P places m face-down cards (denoted by *row 2*) below the given piles, such that only the i-th card (from the left) is $\boxed{\heartsuit}$. We further place m face-down cards (denoted by *row 3*) below the cards, such that only the first card is $\boxed{\heartsuit}$:



2. Apply a pile-shifting shuffle to the sequence of piles where the cards in the same column are regarded as a single pile.



3. Reveal all face-down cards in row 2. This should result in exactly one $\boxed{\heartsuit}$ appearing at a random position; the pile above the revealed $\boxed{\heartsuit}$ is $\boldsymbol{p}_i$ because a pile-shifting

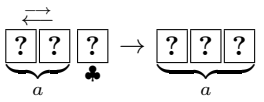
shuffle is a cyclic shuffling and the m cards in row 2 comprise exactly one $\boxed{\heartsuit}$ on the i-th in step 1. After this step is invoked, other operations may be applied to $\boldsymbol{p}_i$ and those around $\boldsymbol{p}_i$. Consequently, it is reverted back to its position in the sequence.

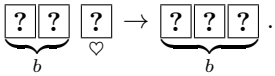
4. Turn over the revealed cards, *i.e.*, the cards in row 2. Subsequently, apply a pile-shifting shuffle again.
5. Reveal all face-down cards in row 3. Similarly, this should result in exactly one $\boxed{\heartsuit}$ appearing, and the pile above the revealed $\boxed{\heartsuit}$ is $\boldsymbol{p}_1$. Therefore, shifting the sequence of piles (such that $\boldsymbol{p}_1$ becomes the leftmost pile in the sequence) results in a sequence of piles of the same order as the original one.

2.4 Sum of Commitments

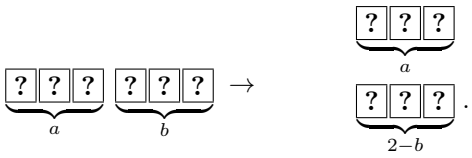
This protocol is defined in [35]; we give a general description given as an example. Suppose that we have commitments to $a, b \in \{0, 1\}$, and we want to output $a + b \in \mathbb{Z}/3\mathbb{Z}$ (in the $\heartsuit$-scheme, see the begining of Sect. 2):

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 $\mathbb{Z}/3\mathbb{Z}$:

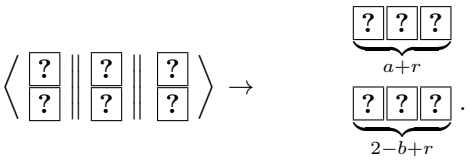


2. Add a $\boxed{\heartsuit}$ on the right of the commitment to b. Those three cards represent b in the $\clubsuit$-scheme in $\mathbb{Z}/3\mathbb{Z}$: 

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:



   (b) Apply a pile-shifting shuffle as follows:



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 $b - r$: $\underbrace{?\,?\,?}_{a+r}\ \underbrace{?\,?\,?}_{b-r}$.

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.

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 $\mathbb{Z}/(n+1)\mathbb{Z}$.

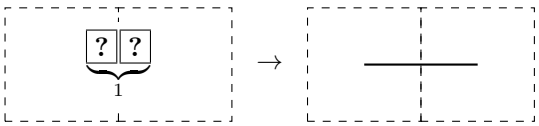
3 Card-Based Protocol for Single Loop Property

In this section, we present our card-based protocol for puzzles with the loop condition. (For the sake of clarity, this section deals with a Moon-or-Sun puzzle.) Before presenting it, let us overview the existing protocol [16] and discuss its drawback.

3.1 Existing Protocol

This existing protocol proposed by Lafourcade *et al.* [16] 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 the loop condition 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.

Setup. To represent a loop with a sequence of cards, the protocol places a commitment *between* each cell in a Moon-or-Sun puzzle. 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:



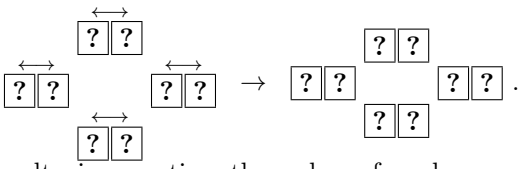
Thus, the protocol begins by placing a commitment to 1 between each cell adjacent to the border of a given board and commitments to 0 on the remaining positions, representing a single loop.

Idea and Procedure. Let us briefly introduce its idea and procedure[3]. After placing commitments in such a way, the protocol repeats to let P "dent" the loop to reduce the loop size one by one so that the loop represents a solution P has (where we

[3]Refer to [16] for specific methods on creating the solution P has.

assume that the size of each cell is one). In this iteration, the protocol ensures that the resulting figure should represent a single loop. Its procedure is as follows.

1. P uses the chosen pile protocol introduced in Sect. 2.3 to select four commitments around an intersection of a given Moon-or-Sun board.
2. V swaps two cards comprising each commitment selected in the previous step as follows:



   This swapping results in negating the value of each commitment, *i.e.*, the existence of a line, and the loop size is decreased by one.
3. V returns the four commitments and ends the remaining steps of the chosen pile protocol. The above steps are repeated.

Drawback. As seen before, the idea behind the protocol [16] is simple, and reducing the loop size is easy-to-implement, just swapping two cards comprising each commitment. However, to ensure that the resulting figure represents a single loop, two additional verifications are required in each iteration above, which affects the efficiency of the protocol. The two additional verifications are as follows.

- In step 1, because V cannot see the position of the commitments selected by P, any position can be selected, such as the one where no line exists, without V noticing it[4]. Therefore, an additional verification is required to ensure that the position selected by P via the chosen pile protocol is correct. This verification requires the application of the chosen pile protocol once, *i.e.*, two pile-shifting shuffles.
- In step 2, even if P selected a correct position in step 1, the resulting figure after swapping may be split into two loops because of loop denting. To prevent this, another verification is required, which comprises the applications of the chosen pile protocol four times, *i.e.*, eight shuffles.

To summarize, the existing protocol [16] requires the two additional verifications of 10 shuffles in each iteration as well as two pile-shifting shuffles for the chosen pile protocol in step 1.

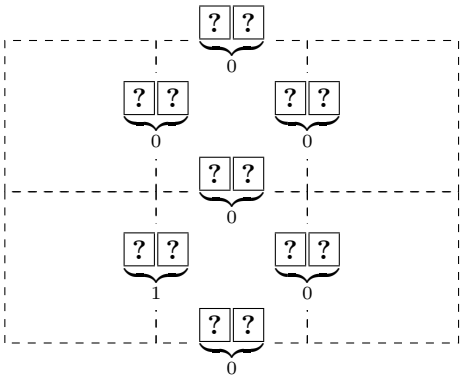
3.2 Idea

To construct an efficient protocol, we employ another existing protocol proposed by Robert *et al.* [24], which deals with puzzles with the *connectivity* condition, such as Nurikabe and Hitori, where the goal is to color cells to be a continuous wall with satisfying some other rules given a grid. This protocol is an interactive protocol between P and V and repeats to let P "color" a cell to represent a solution using the chosen pile protocol. In each iteration, V verifies that a new cell to be colored is next to a colored cell, so that the resulting figure represents a continuous wall.

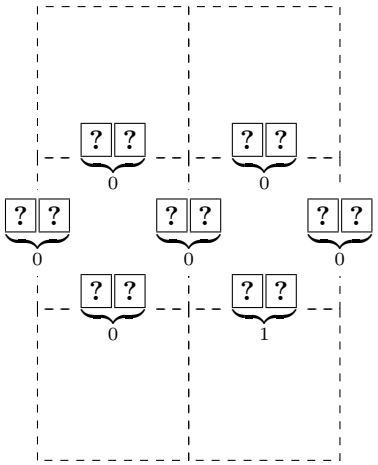
In this study, we apply this existing protocol [24] for the loop condition; we let P "draw" a line one by one, *i.e.*, selecting and negating a commitment to 0, so that the

[4] If P selects such a position, another loop would be created, violating the loop condition.

resulting figure represents a single path, and at the last iteration, P connects the start and end of the path to be a single loop. For this, only one simple verification is required in each iteration to ensure that a new line to be drawn is next to a previously drew line and that the new line is at the tip. That is, V confirms that the value of exactly one commitment is 1 among six commitments around the selected commitment. The positions of those six commitments depend on whether the selected commitment is between two horizontally or vertically adjacent cells; if it is between two vertically adjacent cells, those six commitments are as follows:



Here, the position of the commitment to 1 is denoted as an example. If the selected commitment is between two horizontally adjacent cells, those six commitments are as follows:



However, information about the position of the selected commitment should be secret. We elaborate on achieving such a verification using the chosen pile protocol twice as seen later.

Because this verification requires four pile-shifting shuffles (and two pile-shifting shuffles for executing the chosen pile protocol), our protocol is efficient compared to the existing protocol [16], which requires 12 shuffles in each iteration. As will be discussed

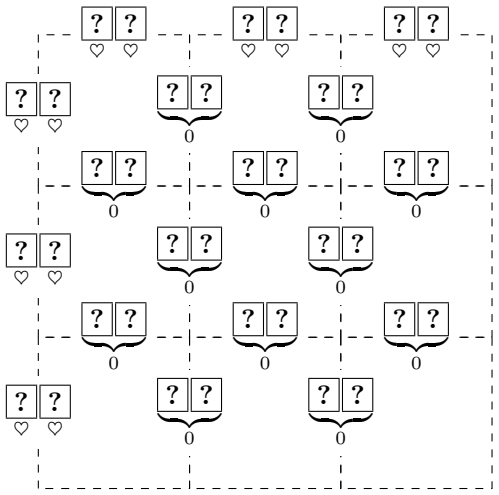
in Sect. 3.3, the number of iterations required for the above protocol is almost the same as in the existing protocol [16], *i.e.*, the number of cells in a given board. Therefore, our improvement results in half the number of shuffles required.

The number of iterations leaks (is equal to) information about the length of a solution loop. To hide this information, we use the same strategy used in the existing protocols [16, 24], in which the number of iterations is equal to the maximum length of possible solution loops in a given board. Observe that our drawing, *i.e.*, negating a commitment, can be applied to "erase" a line, and the condition that the resulting path is a single path even if a single line of the path is erased can be verified exactly in the same way, *i.e.*, , a line to be erased should be at the tip. Therefore, we let P repeat to execute the above steps to draw and erase a line, leaking no information about the length of a solution loop.

3.3 Procedure

We are ready to present the procedure of our card-based protocol.

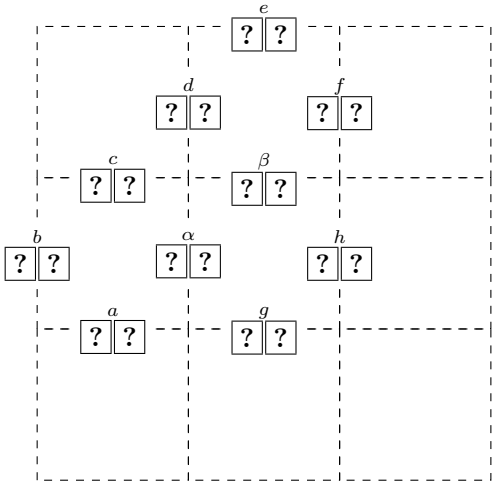
Setup. A prover P and a verifier V place a commitment to 0 between each cell in a given Moon-or-Sun board. As in the existing protocols [16, 24], we further place "dummy" commitments on the top and left part of the outer line of a given grid. Such a dummy commitment consists of two heart cards. For example, commitments on a 3×3 board are placed as follows, where descriptions of rooms and symbols of moon and sun are omitted:



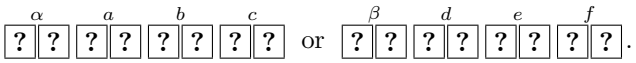
Drawing and Verification. After the setup, the protocol proceeds to draw a single loop as follows.

1. P uses the chosen pile protocol introduced in Sect. 2.3 to select and negate a commitment to 0 and ends the chosen pile protocol. For this, P selects a commitment placed on the solution loop P has (in any position).

2. Let ℓ denote the maximum length of possible loops in the given Moon-or-Sun puzzle, namely the maximum number of commitments to 1. (The value of ℓ will be given in the next paragraph.) P and V repeat the following steps $\ell - 2$ times to draw lines, such that the resulting path require one more line to be a single loop, as follows.
   (a) P uses the chosen pile protocol to select two commitments at the top and left of a cell, as in the existing protocols [16, 24]. For this, P selects two commitments such that they include either a commitment to 0 placed around the edge of the current path drawn by P (if P wants to draw a line), or a commitment to 1 placed on the edge of the current path (if P wants to erase a line). Note that P should memorize the current path/line drawn by P in the previous steps. The detail of how to execute the chosen pile protocol is described in Appx. B.
   (b) V also picks eight commitments around the two commitments selected via the chosen pile protocol. Assuming that the two selected commitments are shifted so that they are placed on the middle cell, the positions of the eight commitments are specified as follows (The detail is described in Appx. B):



   Here, each commitment is denoted by a letter above it; the commitments a, b, and c (d, e, and f) are used to verify the selected commitment α (β) because it is between two horizontally (vertically) adjacent cells. Note that the commitments g and h are common to α and β for the verification.
   (c) P uses the chosen pile protocol to select either the commitments α, a, b, and c or β, d, e, and f:



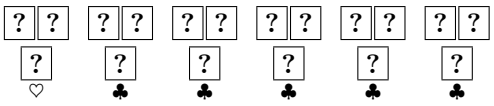
   Note that P has already decided whether to select α or β in mind as described in step 2a.

(d) Let γ denote the selected commitment among the commitments α and β. V negates the commitment γ.

(e) V confirms that the value of exactly one commitment is 1 among the six commitments around the commitment γ as follows.

(i) V places the following six commitments at any order: g, h, the three commitments along with γ selected in step 2c, and the non-selected commitment among α and β.

(ii) V places a face-down card below each commitment as follows:



(iii) V applies a pile-shifting shuffle to the six piles where cards in the same column are regarded as a single pile.

(iv) V reveals only the right card of each commitment to confirm that exactly one $\boxed{\clubsuit}$ appears; otherwise, V aborts. Note that this means that the value of exactly one commitment among the six commitments is 1 because of the encoding rule defined in Eq. (1).

(v) V turns over revealed cards and applies a pile-shifting shuffle again.

(vi) V reveals the card below each commitment to revert the order of the six commitments to their original order similar to the chosen pile protocol.

(f) After returning the commitments, V ends the chosen pile protocol executed in steps 2c and 2a.

3. After repeating the previous step $\ell-2$ times, P uses the chosen pile protocol twice to select the commitment γ and pick the six commitments around γ (*i.e.*, g, h, the three commitments along with γ, and the non-selected commitment among α and β) similar to the previous step. For this, P selects the commitment γ such that the current path becomes a single loop if the value of γ becomes 1.
4. V confirms that the value of the commitment γ is 0 and negates it. If not, V aborts.
5. V confirms that the value of exactly one commitment is 1 among the three commitments of those selected along with γ and also confirms that the value of exactly one commitment is 1 among the three commitments of g, h, and the non-selected commitment among α and β. This verification is similar to step 2e and is omitted.
6. V ends the above chosen pile protocols.

As seen above, each iteration requires a verification with six pile-shifting shuffles. The correctness and security proof are given in Sect. 3.4.

An Upper Bound on the Length of the Loop in Moon-Or-Sun. For concreteness, we determine how long a loop in a Moon-or-Sun puzzle can maximally be, *i.e.*, ℓ. Given such a puzzle with $n > 1$ rows and $m > 1$ columns, the longest possible loop would be one that visits at most every cell. If we view the puzzle as an $n \times m$-grid graph, where each cell of the puzzle is a vertex and each line between two neighboring cells is an edge, the number of lines between the neighboring cells that are crossed by

our loop then translates to the number of edges of the respective cycle on this grid graph.

To get an upper bound, we assume this cycle visits all $n \times m$ vertices (*i.e.*, the graph is Hamiltonian). Then, the number of edges in this cycle would be also $n \times m$. Note that if the graph is not Hamiltonian, $n \times m$ is still an upper bound, as then a possible loop just passes through fewer vertices and hence uses fewer edges. Overall, it is safe to allow P at most $\ell = n \times m$ steps, where they can choose between extending the length of the current path by one, or not, to be able to draw any allowed loop in the puzzle. Note that ℓ is almost the same as in the ZKP protocol for Slitherlink [16], where the protocol repeats to "dent" the loop $n \times m - 1$ times to hide information about the size, because the maximum size of a possible loop is clearly $n \times m$ and the minimum size is one.[5]

3.4 Security Proof

Our proposed protocol satisfies the following theorems. These theorems will be used in part to prove the three properties of our ZKP protocol for Moon-or-Sun.

Theorem 1. *A prover P can represent any single loop using our protocol described in Sect. 3.3.*

Proof. Remember that our protocol uses the chosen pile protocol to let P select a position in which P wants to draw a line one by one, *i.e.*, negate a commitment to 0. Briefly, P can represent any single loop simply by drawing its lines.

Because P can select either draw or erase a line, the parity of the length of a solution loop and ℓ should be the same. Note that the length of any single loop, *i.e.*, perimeter of any rectilinear polygon, is even if the length of a single line is assumed to be one. Therefore, P can always connect a single path to be a single loop in the last step. □

Theorem 2. *The resulting figure after running our protocol described in Sect. 3.3 always represents a single loop, no matter how P acts.*

Proof. We prove that our protocol never produces a loop crossing and/or branching off and a path not being closed. Remember that our protocol lets P draw a new line next to a previously drawn line, so that the new line becomes the tip because the protocol confirms that there is only a single line around the new line. That is, if P selects α (resp. β) to draw a line in step 2c, the protocol reveals the values of a, b, c, β, h, g (resp. d, e, f, α, g, h) to confirm that there is only a single line in step 2e. Therefore, multiple loops cannot be drawn in the protocol, because if a malicious P tries to do so, P needs to draw a new line such that there is no line around it. Similarly, a loop branching off cannot occur because if so, P needs to draw a line such that there are two lines around it. This discussion also holds when P selects to erase a line. Finally, because lines P draws are connected to be a loop at the last iteration, it results in a loop that does not cross and does not branch off, *i.e.*, single loop.

[5] Clearly, the minimum loop cannot satisfy other rules, and we can further reduce the number of iterations for the existing ZKP protocol for Slitherlink [16] in this sense, but it was not discussed.

We note that adding dummy commitments in the setup phase is necessary to delimit the action area. Also note that a dummy commitment should consist of $\boxed{\heartsuit}\boxed{\heartsuit}$ to prevent cheating. P could select a dummy commitment to draw a line, but this does not matter because negating a dummy commitment results in the dummy commitment itself. □

Theorem 3. *Our protocol described in Sect. 3.3 leaks no information about P's solution of a given board to a verifier V.*

Proof. We follow the security definition formalized in the computational model for card-based protocols [19]. Here, the security of card-based protocols means that both the input and output are stochastically independent to a visible sequence trace, *i.e.*, what we can observe from a sequence of cards operated during the protocol. In our protocol, the input is the solution P has; more precisely, it is a sequence of commitments placed on the board and a sequence of cards placed by P when executing the chosen pile protocol. The output is a sequence of commitments on the board after executing the protocol, and hence, it is obviously independent because they are face-down cards. Therefore, we prove that the input is independent to a visible sequence trace. We note that the number of iterations ℓ can be computed from the board. That is, the number of actions on the sequence of cards is independent to the input. Thus, let us focus on steps where face-down cards are revealed as follows.

- The security of step 1 is obvious, as it just uses the chosen pile protocol once and negates the selected commitment. The security of the chosen pile protocol is also obvious from its description in Sect. 2.3; revealing all cards in row 2 (and row 3) leaks no information about the position of $\boxed{\heartsuit}$ initially placed by P in row 2, because before revealing, a pile-shifting shuffle is applied, resulting in the $\boxed{\heartsuit}$ appearing at a random position. We note that the output of the chosen pile protocol is a sequence of commitments, which is visually the same as the sequence of commitments initially given to the protocol.
- The security of steps 2a, 2c, and 3 (and also 2f and 6) is clear from the above description.
- In step 2(e)iv, one $\boxed{\clubsuit}$ is revealed in a random position among (the right cards of) the six commitments, because a pile-shifting shuffle is applied in step 2(e)iii. This means that the value of only one commitment among them is 1, *i.e.*, $\boxed{\heartsuit}\boxed{\clubsuit}$, which is exactly what V wants to confirm.
- In step 2(e)vi, one $\boxed{\heartsuit}$ appears in a random position because of the application of a pile-shifting shuffle.
- The security of step 5 is clear from the above description.

□

4 ZKP Protocol for Moon-or-Sun

We present a card-based ZKP protocol for a Moon-or-Sun puzzle. Our protocol has two phases: the setup and verification phases. The setup phase uses our protocol proposed in Sect. 3 to construct a loop. The verification phase verifies all the rules other than rules 1 and 2.

4.1 Setup

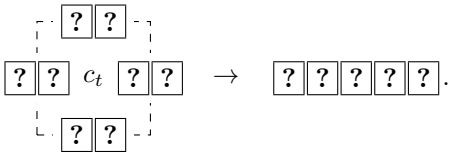
As constructed in Sect. 3, a solution is represented with a commitment between each adjacent cells. Moreover, we place a commitment on each cell to represent the loop passing through a symbol (*i.e.*, moon or sun symbol). The setup is done in two steps:

1. Draw the loop using our protocol presented in Sect. 3;
2. Place the commitments inside the cells. Notice that we modify only cells with moon or sun symbol.

Forming the loop. We directly use the construction described in Sect. 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 *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 neighbors of the targeted cell $\boxed{c_t}$. The result is in ♡-scheme:



   Here, the number of neighbors around $\boxed{c_t}$ is four, and the number of cards in the resulting sequence is five. If it is at the border of the grid, the number of cards will be either four or three; in any case, 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.

4.2 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:

$$\langle \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?} \rangle \;\to\; \boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit} \text{ or } \boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}.$$

2. Repeat the previous step for moon cells.
3. Execute the Mizuki–Sone XOR protocol [20] 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 appear, 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.

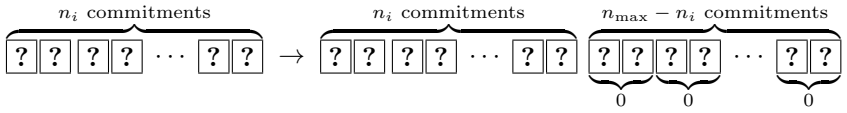
Given a solution as in Fig. 2, consider verifying whether a room (referred to as the target) satisfies rule 5 or not. For this, we examine the two rooms connected to the target room (*i.e.*, those through which line passes) and ensure that the loop passes through different symbols within the target room 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[6]. 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.

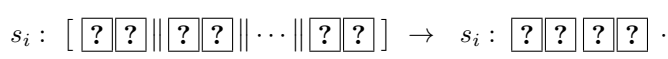
1. For every adjacent room R_i, let c_j denote each of the n_i commitments on the border between R_0 and R_i, for $1 \leq j \leq n_i$ (in any order). Collect one c_j for every j and add "dummy" commitments to 0 so that the total number of collected

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

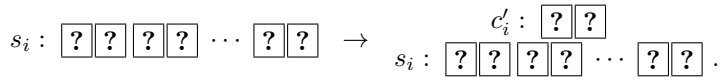
commitments becomes $n_{\max} = \max(n_1, \ldots, n_k)$ as follows:



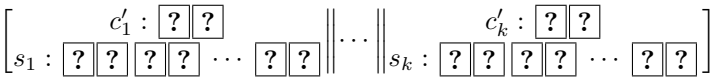
Let s_i denote the sequence of the $n_{\max}$ commitments. Apply a pile-scramble shuffle to s_i as follows:



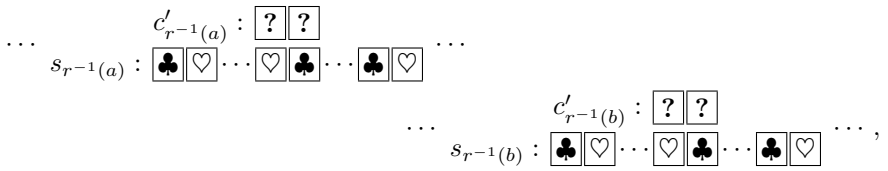
2. For every adjacent room R_i, $1 \leq i \leq k$, let c'_i denote a commitment on any sun cell within R_i. Place one c'_i above s_i. (If R_i has no sun, then let c'_i denote a commitment on any moon cell and swap the two cards constituting c'_i before placing it.)



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



4. Reveal all the commitments constituting s_i for all i, $1 \leq i \leq k$. Then exactly two commitments to 1 should be revealed, each appearing in different s_i (rule 3) [7]; if not, V aborts. Denote these positions as a and b ($a < b$) as follows:

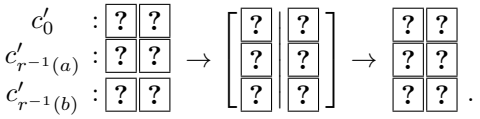


where $r \in S_k$ is the random permutation generated through the application of a pile-scramble shuffle at step 3.

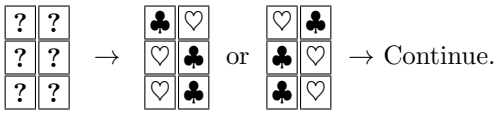
5. Let c'_0 denote a commitment on any sun cell in the target room R_0. (If there is no sun, then denote a commitment on any moon cell by c'_0 and swap the two cards constituting c'_0.) Execute an extended version of the Mizuki–Sone XOR protocol [20] with c'_0, $c'_{r^{-1}(a)}$, and $c'_{r^{-1}(b)}$ as follows.

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

(a) Place c'_0, $c'_{r^{-1}(a)}$, and $c'_{r^{-1}(b)}$ and apply a random bisection cut as follows:



(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.



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. 6.

4.3 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. The evaluation of the part of constructing a single loop is given in Sect. 3.

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 [20]) 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 [20], *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 [35] 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.

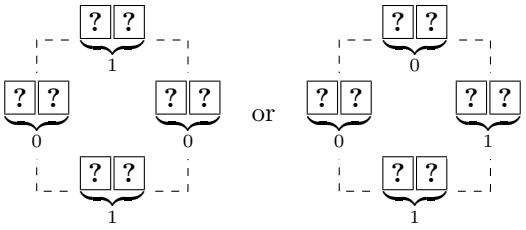
5 Security Proofs

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

Theorem 4 (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, P knowing the solution can first form any loop using our proposed protocol described in Sect. 3 as in Theorem 1, and hence, P can form the

solution loop. For placing a commitment inside a cell, we use the sum protocol [35] 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. 4.2, 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.*, ♣♡♣♡ $\cdots$ or ♡♣♡♣ $\cdots$. 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 [20] 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:

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

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

□

Theorem 5 (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. We consider the case that each of the 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.*, ♣♡♣♡♡♣. 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 [20]. Thus, V always aborts.

In any case, the verifier always rejects. Note that the loop condition cannot be violated as in Theorem 2. □

Theorem 6 (Zero-knowledge)**.** *Any party in the verifier role learns nothing about P's solution.*

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 in any time; 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 our protocol for the loop condition. Subsequently, it applies the sum protocol [35] introduced in Sect. 2.4. Note that our protocol and this existing protocol are proved to leak no information as in Theorem 3 and [35], respectively. 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 [20], which leaks no information as proved in [20].

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 $c'_{r^{-1}(b)}$.

□

6 Discussion

Here, we discuss whether we can reduce the number of executions of our method for rule 5 described in Sect. 4.2. Suppose that we execute the verification phase described in Sect. 4.2 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 7. *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. 4.2. 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 7 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 7. 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 efficiently. Although techniques for evaluating the execution time of card-based protocols exist [18], doing so in this case is non-trivial, due to this additional computationally expensive 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 8. *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 7 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 [28].

7 Conclusion

We proposed a card-based protocol for pencil-and-paper puzzles with the single loop property, which is more efficient than the existing protocol [16]. This proposed protocol is applied to the construction of 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.

Acknowledgments. We thank the anonymous referees, whose comments have helped us to improve the presentation of the paper. The fourth author thanks the discussions held in Institute of Mathematics for Industry, Joint Usage/Research Center in Kyushu University (FY2024a035).

Declarations

Funding. 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.

Conflict of Interest. The third author (Pascal Lafourcade) is one of the board members of this special issue on card-based cryptography.

Ethics Approval and Consent to Participate. Not applicable.

References

[1] den Boer B (1989) More efficient match-making and satisfiability: *The Five Card Trick*. In: Quisquater J, Vandewalle J (eds) EUROCRYPT 1989, LNCS, vol 434. Springer, Berlin, Heidelberg, pp 208–217

[2] Bultel X, Dreier J, Dumas J, et al (2016) Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen. In: Fun with Algorithms, LIPIcs, vol 49. Schloss Dagstuhl, Dagstuhl, pp 8:1–8:20

[3] Bultel X, Dreier J, Dumas J, et al (2018) Physical zero-knowledge proof for Makaro. In: SSS 2018, LNCS, vol 11201. Springer, Cham, pp 111–125

[4] Chien YF, Hon WK (2010) Cryptographic and physical zero-knowledge proof: From Sudoku to Nonogram. In: Boldi P, Gargano L (eds) Fun with Algorithms, LNCS, vol 6099. Springer, Berlin, Heidelberg, pp 102–112

[5] Dumas JG, Lafourcade P, Miyahara D, et al (2019) Interactive physical zero-knowledge proof for Norinori. In: Du DZ, Duan Z, Tian C (eds) COCOON 2019, LNCS, vol 11653. Springer, Cham, pp 166–177

[6] Fukasawa T, Manabe Y (2022) Card-based zero-knowledge proof for the nearest neighbor property: Zero-knowledge proof of ABC End View. In: Batina L, Picek S, Mondal M (eds) Security, Privacy, and Applied Cryptography Engineering, LNCS, vol 13783. Springer, Cham, pp 147–161

[7] Goldreich O, Micali S, Wigderson A (1991) Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. J ACM 38(3):691–729

[8] Gradwohl R, Naor M, Pinkas B, et al (2009) Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. Theory Comput Syst 44(2):245–268

[9] Hand S, Koch A, Lafourcade P, et al (2023) Check alternating patterns: A physical zero-knowledge proof for Moon-or-Sun. In: Shikata J, Kuzuno H (eds) Advances in Information and Computer Security, LNCS, vol 14128. Springer, Cham, pp 255–272

[10] Hatsugai K, Asano K, Abe Y (2024) A physical zero-knowledge proof for Sumplete, a puzzle generated by ChatGPT. In: Wu W, Tong G (eds) Computing and Combinatorics, LNCS, vol 14422. Springer, Cham, pp 398–410

[11] Isuzugawa R, Miyahara D, Mizuki T (2021) Zero-knowledge proof protocol for Cryptarithmetic using dihedral cards. In: Kostitsyna I, Orponen P (eds) UCNC 2021, LNCS, vol 12984. Springer, Cham, pp 51–67

[12] Iwamoto C, Ide T (2022) Moon-or-Sun, Nagareru, and Nurimeizu are NP-complete. IEICE Trans Fundamentals 105(9):1187–1194

[13] Koch A, Walzer S (2021) Foundations for actively secure card-based cryptography. In: Farach-Colton M, Prencipe G, Uehara R (eds) Fun with Algorithms, LIPIcs, vol 157. Schloss Dagstuhl, Dagstuhl, pp 17:1–17:23

[14] Komano Y, Mizuki T (2022) Physical zero-knowledge proof protocol for Topswops. In: Su C, Gritzalis D, Piuri V (eds) Information Security Practice and Experience, LNCS, vol 13620. Springer, pp 537–553

[15] Komano Y, Mizuki T (2023) Card-based zero-knowledge proof protocol for Pancake Sorting. In: Bella G, Doinea M, Janicke H (eds) SecITC, LNCS, vol 13809. Springer, pp 222–239

[16] Lafourcade P, Miyahara D, Mizuki T, et al (2021) How to construct physical zero-knowledge proofs for puzzles with a "single loop" condition. Theor Comput Sci 888:41–55

[17] Miyahara D, Sasaki T, Mizuki T, et al (2019) Card-based physical zero-knowledge proof for Kakuro. IEICE Trans Fundamentals 102-A(9):1072–1078

[18] Miyahara D, Ueda I, Hayashi Y, et al (2021) Evaluating card-based protocols in terms of execution time. Int J Inf Secur 20:729–740

[19] Mizuki T, Shizuya H (2014) A formalization of card-based cryptographic protocols via abstract machine. Int J Inf Sec 13(1):15–23

[20] Mizuki T, Sone H (2009) Six-card secure AND and four-card secure XOR. In: Deng X, Hopcroft JE, Xue J (eds) FAW 2009, LNCS, vol 5598. Springer, Berlin, Heidelberg, pp 358–369

[21] Mizuki T, Asiedu IK, Sone H (2013) Voting with a logarithmic number of cards. In: Mauri G, Dennunzio A, Manzoni L, et al (eds) Unconventional Computation and Natural Computation, LNCS, vol 7956. Springer, pp 162–173

[22] Nishida T, Hayashi Y, Mizuki T, et al (2015) Card-based protocols for any boolean function. In: Jain R, Jain S, Stephan F (eds) Theory and Applications of Models of Computation, LNCS, vol 9076. Springer, Cham, pp 110–121

[23] Robert L, Miyahara D, Lafourcade P, et al (2022) Physical zero-knowledge proof and NP-completeness proof of Suguru puzzle. Inf Comput 285:1–14

[24] Robert L, Miyahara D, Lafourcade P, et al (2022) Card-based ZKP for connectivity: Applications to Nurikabe, Hitori, and Heyawake. New Gener Comput 40:149–171

[25] Robert L, Miyahara D, Lafourcade P, et al (2022) Card-based ZKP protocol for Nurimisaki. In: Devismes S, Petit F, Altisen K, et al (eds) Stabilization, Safety, and Security of Distributed Systems, LNCS, vol 13751. Springer, pp 285–298

[26] Robert L, Miyahara D, Lafourcade P, et al (2022) Hide a liar: Card-based ZKP protocol for Usowan. In: Du D, Du D, Wu C, et al (eds) Theory and Applications of Models of Computation, vol 13571. Springer, pp 201–217

[27] Robert L, Miyahara D, Lafourcade P, et al (2023) Physical ZKP protocols for Nurimisaki and Kurodoko. Theor Comput Sci 972:114071

[28] Robertson N, Sanders DP, Seymour PD, et al (1996) Efficiently four-coloring planar graphs. In: Miller GL (ed) ACM Symposium on the Theory of Computing. ACM, pp 571–575

[29] Ruangwises S (2021) An improved physical ZKP for Nonogram. In: Du DZ, Du D, Wu C, et al (eds) COCOA 2021, Cham, pp 262–272

[30] Ruangwises S (2023) Physical zero-knowledge proof for ball sort puzzle. In: Della Vedova G, Dundua B, Lempp S, et al (eds) Unity of Logic and Computation, LNCS, vol 13967. Springer, Cham, pp 246–257

[31] Ruangwises S (2023) Physical zero-knowledge proofs for Five Cells. In: Aly A, Tibouchi M (eds) Progress in Cryptology – LATINCRYPT 2023, LNCS, vol 14168. Springer, Cham, pp 315–330

[32] Ruangwises S (2023) Physically verifying the first nonzero term in a sequence: Physical ZKPs for ABC end view and Goishi Hiroi. In: Li M, Sun X, Wu X (eds) Frontiers of Algorithmics, LNCS, vol 13933. Springer, Cham, pp 171–183

[33] Ruangwises S, Itoh T (2021) Physical zero-knowledge proof for Numberlink puzzle and k vertex-disjoint paths problem. New Gener Comput 39(1):3–17

[34] Ruangwises S, Itoh T (2021) Physical ZKP for connected spanning subgraph: Applications to Bridges puzzle and other problems. In: Kostitsyna I, Orponen P (eds) UCNC 2021, LNCS, vol 12984. Springer, Cham, pp 149–163

[35] Ruangwises S, Itoh T (2021) Securely computing the n-variable equality function with 2n cards. Theor Comput Sci 887:99–110

[36] Ruangwises S, Itoh T (2022) 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. Schloss Dagstuhl, pp 24:1–24:12

[37] Ruangwises S, Itoh T (2022) Physical ZKP for Makaro using a standard deck of cards. In: Du D, Du D, Wu C, et al (eds) Theory and Applications of Models of Computation, LNCS, vol 13571. Springer, pp 43–54

[38] Ruangwises S, Itoh T (2022) Two standard decks of playing cards are sufficient for a ZKP for Sudoku. New Gener Comput 40(1):49–65

[39] Sako K, Kilian J (1995) Receipt-free mix-type voting scheme. In: Guillou LC, Quisquater JJ (eds) EUROCRYPT'95, LNCS, vol 921. Springer, Berlin, Heidelberg, pp 393–403

[40] Sasaki T, Miyahara D, Mizuki T, et al (2020) Efficient card-based zero-knowledge proof for Sudoku. Theor Comput Sci 839:135–142

[41] Shikata H, Toyoda K, Miyahara D, et al (2022) Card-minimal protocols for symmetric Boolean functions of more than seven inputs. In: Seidl H, Liu Z, Pasareanu CS (eds) Theoretical Aspects of Computing – ICTAC 2022, LNCS, vol 13572. Springer, Cham, pp 388–406

[42] Shikata H, Miyahara D, Mizuki T (2023) Few-helping-card protocols for some wider class of symmetric Boolean functions with arbitrary ranges. In: 10th ACM Asia Public-Key Cryptography Workshop. ACM, New York, APKC '23, pp 33–41

[43] Shinagawa K, Nuida K (2021) A single shuffle is enough for secure card-based computation of any Boolean circuit. Discrete Applied Mathematics 289:248–261

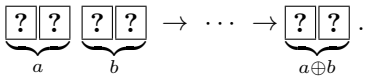
[44] Shinagawa K, Mizuki T, Schuldt JCN, et al (2017) Card-based protocols using regular polygon cards. IEICE Trans Fundamentals 100-A(9):1900–1909

[45] Tamura Y, Suzuki A, Mizuki T (2024) Card-based zero-knowledge proof protocols for the 15-puzzle and the token swapping problem. In: ACM Asia Public-Key Cryptography Workshop. ACM, New York, pp 11–22

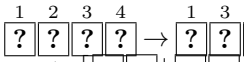
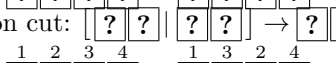
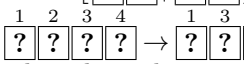
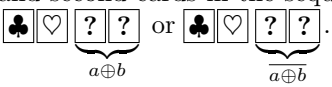
[46] Tanaka K, Mizuki T (2023) Two uno decks efficiently perform zero-knowledge proof for Sudoku. In: Fernau H, Jansen K (eds) Fundamentals of Computation Theory, LNCS, vol 14292. Springer, Cham, pp 406–420

Appendix A Full Description of XOR and Copy Protocols

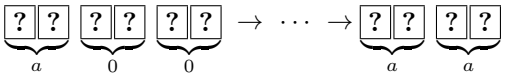
XOR protocol. Given commitments to $a, b \in \{0, 1\}$, the Mizuki–Sone XOR protocol [20] outputs a commitment to $a \oplus b$:



This protocol proceeds as follows.

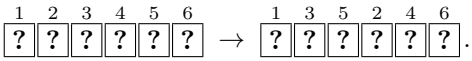
1. Rearrange the sequence:  .
2. Apply a random bisection cut:  .
3. Rearrange the sequence:  .
4. Reveal the first and second cards in the sequence to obtain the output commitment as follows:  or .

Copy protocol. Given a commitment to $a \in \{0, 1\}$ along with two commitments to 0, the Mizuki–Sone copy protocol [20] outputs two commitments to a:

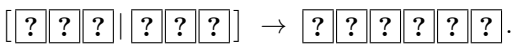


This protocol proceeds as follows.

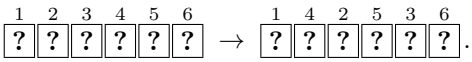
1. Rearrange the sequence as follows:



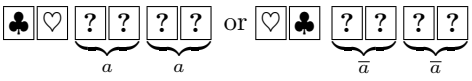
2. Apply a random bisection cut to the sequence as follows:



3. Rearrange the sequence as follows:

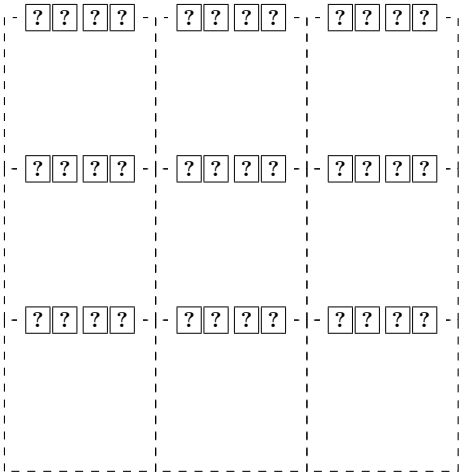


4. Reveal the first and second cards in the sequence to obtain the output commitments as follows:

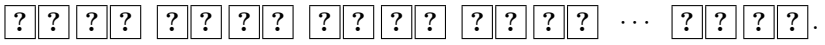


Appendix B How to Execute Chosen Pile Protocol

Let us describe the procedure with the example of commitments placed on a 3×3 board as shown in Sect. 3.3. To select two commitments at the top and left of a cell via the chosen pile protocol, we make a sequence of 4-card pile as follows: we pick a commitment at the left of each cell and place it right to the one at the top of the cell.



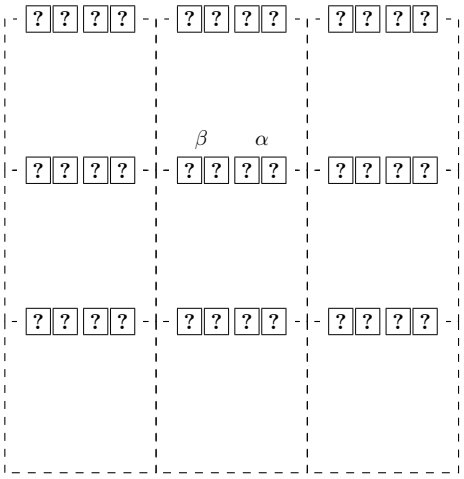
Then, each two commitments is picked from top to bottom to make a sequence of 4-card piles as follows:



Therefore, using the chosen pile protocol, P can select two commitments at the top and left of a cell, among which P wants to negate the value of a commitment.

After P selects the two commitments, V picks eight commitments around them. Since the chosen pile protocol uses the pile-shifting shuffle to select a pile, we can further shift the sequence of 4-card piles so that the selected pile (*i.e.*, the two

commitments) is placed in the middle cell as follows:



Here, the two selected commitments are specified by α and β to be consistent with Sect. 3.3. Therefore, V can always pick the eight commitments around them, without knowing their original positions.