Card-Based ZKP Protocols for Connectivity-Based Puzzles: Extending to Tree Structures with Application to Nurimeizu

Daiki Miyahara
The University of Electro-Communications, Japan
National Institute of Advanced Industrial Science and Technology, Japan

Pascal Lafourcade
Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Mines Saint-Etienne, LIMOS, 63000 Clermont-Ferrand, France

Maxime Puys
Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Mines Saint-Etienne, LIMOS, 63000 Clermont-Ferrand, France

Abstract

Card-based zero-knowledge proof (ZKP) protocols allow a prover to convince a verifier that it knows a witness of a given statement, without revealing any information, using a physical deck of playing cards. Previous studies have focused on puzzles with a specific connected component, such as a simple cycle and a polyomino. In this study, we propose a unified approach to handle a family of connected components, including a tree, path, cycle, and polyomino. This approach achieves this verification in $O(mn)$ steps relative to a given grid size $m \times n$. Using this approach, we construct a card-based ZKP protocol for Nurimeizu, where the goal is to find the shortest path on a given grid.

2012 ACM Subject Classification Theory of computation → Cryptographic protocols

Keywords and phrases Card-based cryptography, Nurimeizu, Nicoli, ZKP.

Digital Object Identifier 10.4230/LIPIcs.FUN.2026.40

Acknowledgements We thank the anonymous referees, whose comments have helped us to improve the presentation of the paper. This work was supported in part by JSPS KAKENHI Grant Number JP23H00479, by JSPS Bilateral Joint Research Projects JPJSBP120253206, by Institute of Mathematics for Industry, Joint Usage/Research Center in Kyushu University, 2023a020, 2024a035, and 2025a036, by PHC Sakura (53290WD), and by ANR Project PRIVA-SIQ (ANR-23-CE39-0008).

1 Introduction

A zero-knowledge proof (ZKP) protocol is a cryptographic protocol that allows a prover P to convince a verifier V that P knows a witness of a given statement, without revealing anything other than the validity of the statement itself. In the context of cryptocurrencies, ZKPs enable the network to verify transactions without revealing any underlying private data. In contrast to these digital approaches, we study *card-based ZKP protocols* that use a physical deck of playing cards to realize ZKPs. One of the primary advantages is that they do not require trust in computers or electronic devices, which is, for instance, important in the context of elections [12].

Previous studies have developed ZKP protocols for various puzzles; for example, given a puzzle instance such as Sudoku, P can convince V that P knows a valid solution without leaking any information about the solution (e.g., [4, 5, 7, 36]). Solving games is a really clever way to design basic blocks depending on the rules of the game, which could later be introduced in more realistic use-cases. Recent research has focused on puzzles with a specific

Table 1 Connectivity-based classification of Nikoli puzzles. Refer to [39] for their computational complexity.

Category	Structure	Representative Nikoli Puzzles
Single Tree	Tree	Nurimeizu (white)
Single Path	Path	Nurimeizu (goal)
Single Loop	Cycle	Dotchi-Loop, Masyu [18], Moon-or-Sun [8], Mid-loop, Nagareru Slitherlink [18], Suraromu, Yajilin
Single Connected Region	Polyomino	Heyawake [29], Hitori [29], Kurdoko [30], Kurotto, LITS, Nurikabe (black) [29], Nurimisaki [30], Nuritwin, Sukoro [35], Yajisan Kazusan
Multiple Connected Regions	Polyominoes	Evolomino, Fillomino, LITS (tetrominoes), Nurikabe (white) [29], Ripple Effect [32], Sashigane Shakashaka [22], Shikaku [34]
Multiple Paths	Disjoint open simple paths	Hebi, Herugolf [23], Hotaru Beam [28], Pencils
Connected Islands	Connected spanning subgraph	Hashiwokakero [33]
Chain	Corner-adjacent polyominoes	Chained Block

connected component; for example, [8, 18] studied the famous puzzle Slitherlink, which asks a player to form a single loop (i.e., cycle) on a given grid. Another target shape is a single connected region (i.e., a polyomino), and Sasaki and Shinagawa [35] proposed a ZKP protocol for Sukoro, in which all numbered cells must be connected.

1.1 Contribution

In this study, we propose a unified approach to handle a family of connected components on a grid, including a tree, (simple) path, (simple) cycle, and polyomino. Table 1 classifies Nikoli puzzles in terms of their required connected components, where Nikoli is a famous Japanese company for puzzles. As summarized in this table, our contribution is to extend previous approaches to include a tree and path and to construct a general protocol that unifies these connectivity requirements. We note that our study does not address *multiple* connected components. Our general protocol builds on previous findings [8] that showed protocols for a polyomino [29] could be modified to handle a simple cycle.

Furthermore, we construct a ZKP protocol for the puzzle *Nurimeizu*, which is an NP-complete problem proved in [11]. As will be explained in Section 2.1, Nurimeizu asks a player to color a given grid so that white cells form a tree and a shortest path between the start and goal. In general, proving the validity of a shortest path is a computationally intensive task for ZKP, as it typically requires executing an algorithm like Dijkstra's in a zero-knowledge manner. However, our observation is that the above requirement can be naturally satisfied by proving the existence of a path within a tree structure using our general protocol.

Remarks in Table 1. The existing ZKP protocol for Herugolf [23] cannot not be directly applied to Hotaru Beam due to differences in path constraints. While a Herugolf puzzle

provides both the number of turns and path lengths, a Hotaru Beam puzzle provides only the number of turns. Therefore, the existing ZKP protocol for Hotaru Beam [28] is applicable to Herugolf, making it superior in this sense.

1.2 Related Work

A closely related recent study by Nuida [27] proposed a card-based protocol for counting the number of connected components in general graphs. This protocol can be adopted to construct a ZKP protocol for puzzles involving polyomino placement, by verifying that the number of connected components is exactly one. For a $k \times k$ grid, this approach requires $O(k^3 \log k)$ steps. A notable feature of the protocol is that it is *non-interactive*, meaning that P only needs to place cards representing a solution at the beginning. In contrast, while our protocol is interactive and requires P to place additional cards during the execution, it is specifically optimized for grid structures and achieves an efficient complexity of $O(k^2)$.

Various ZKP protocols have been proposed for diverse problems, such as Sumplete [9], Ball Sort [31], the 15-puzzle [37], the 3-coloring [20], the Hamiltonian cycle [33], Topswops [16], and Pancake sorting [17]. Another interesting topic is the use of physical objects to realize cryptographic protocols, such as a PEZ dispenser [1, 2, 25], balls in bags [21], coins [15, 19], a custom-made box [3, 6], and a balance [12, 13].

2 Preliminaries

We provide the necessary background to understand our proposed protocols, including the basic rules of Nurimeizu and the fundamental operations of card-based ZKP protocols.

2.1 Rules of Nurimeizu

Figure 1 depicts an example of a Nurimeizu puzzle. Given an $m \times n$ grid, the goal of Nurimeizu is to color certain cells (called "Nuri" in Japanese) based on the following rules:

- Create a maze from the start (S) to the goal (G) across only white cells.
- **Room**: All cells within a room (enclosed by bold lines) have to be either colored or left white.
- **Tree**: The set of all white cells must form a tree structure (i.e., they must be connected and contain no cycles).
- **Shortest Path**: There must exist a shortest path between S and G through white cells. The shortest path include all ◯-cells (circle) and no △-cell (triangle). In addition, rooms with S, G, ◯, and △ are left white.[1]
- **Square of** 2×2: No 2×2 area of cells can be entirely of the same color.

2.2 Card-Based Protocols

We use two-color deck of cards: blacks ♣ and reds ♥ whose backs are all identical ?. Using these cards, we encode a Boolean value as well as specific symbols as follows:

- Boolean values: A bit $x \in \{0, 1\}$ is represented by the order of two cards: $0 =$ ♣♥ and $1 =$ ♥♣.

[1] While the rule shown in the Nikoli website uses the term "shortest path," the word "shortest" can be omitted, as implied in Section 1.1.

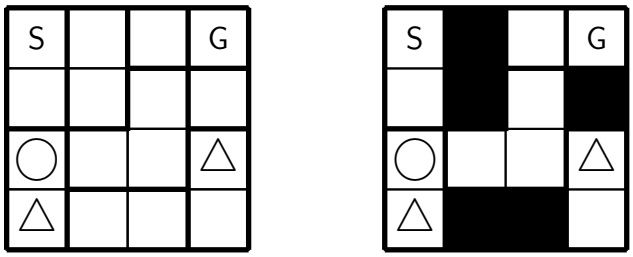


Figure 1 Example of a Nurimeizu puzzle (left) and its solution (right) taken from the Nikoli website: `https://www.nikoli.co.jp/en/puzzles/nurimeizu/`, with S (start), G (goal), ◯-cells (circle), and △-cells (triangle).

- Edge $\star$: To represent the active end of a path or cycle, we use ♣♣.
- Sentinel $\perp$: To represent the boundary of the grid, we use ♥♥ as a sentinel.

We say two face-down cards ?? as a *commitment* to x if the two cards represent $x \in \{0, 1, \star, \perp\}$ according to the above encoding.

Pile-Scramble Shuffle. To introduce randomness, we use a *pile-scramble shuffle* (pile-scramble for short) [10]. This shuffling differs from conventional shuffling in that it permutes "piles" of cards as single units rather than individual cards. Specifically, given a sequence of piles $(p_1, p_2, \ldots, p_k)$, where each p_i consists of the same number of face-down cards, a pile-scramble shuffle applies a random permutation π drawn from the symmetric group of degree k, denoted by $[\cdot|\ldots|\cdot]$:

$$[\overset{p_1}{??\cdots?} \mid \overset{p_2}{??\cdots?} \mid \cdots \mid \overset{p_k}{??\cdots?}] \;\to\; \overset{p_{\pi(1)}}{??\cdots?}\ \overset{p_{\pi(2)}}{??\cdots?}\ \cdots\ \overset{p_{\pi(k)}}{??\cdots?}.$$

A pile-scramble is called a random bisection cut in [24]. If $k = 2$, and secure implementations of a random bisection cut is discussed in [38].

Pile-Shifting Shuffle. We use another shuffling called a *pile-shifting shuffle* (pile-shifting for short) in [26]. Specifically, given a sequence of piles $(p_0, p_1, \ldots, p_{k-1})$ where each p_i consists of the same number of face-down cards, a pile-shifting shuffle applies a random cyclic shift $r \in \{0, 1, \ldots, k-1\}$, denoted by $\langle\cdot|\ldots|\cdot\rangle$:

$$\langle \overset{p_0}{??\cdots?} \mid \overset{p_1}{??\cdots?} \mid \cdots \mid \overset{p_{k-1}}{??\cdots?}\rangle \;\to\; \overset{p_r}{??\cdots?}\ \overset{p_{1+r}}{??\cdots?}\ \cdots\ \overset{p_{k-1+r}}{??\cdots?},$$

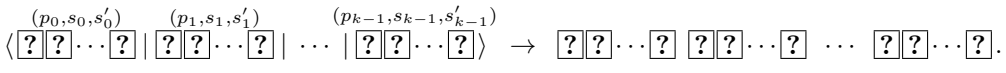
where the indices are taken modulo k. Here, the order of cards within each pile is unchanged. This shuffling is realized physically by arranging the k piles in a circle on a table and rotating the arrangement by a random amount.

Chosen Pile Protocol. Our general protocol employs the chosen pile protocol proposed by Koch and Walzer [14]. Given a sequence of piles $(p_0, p_1, \ldots, p_{k-1})$, this protocol allows P to select a specific pile p_i without revealing the index i to V. Moreover, the original sequence is restored (after the operation on p_i is completed).

The protocol proceeds as follows.

1. P appends a face-down card ? to every pile p_j $(0 \le j < k)$. Let s_j be the card appended to p_j. Specifically, P sets $s_i =$ ♥ for the desired index i, and $s_j =$ ♣ for all $j \neq i$.
2. Additionally append $s'_0 =$ ♥ and $s'_j =$ ♣ for all $j \neq 0$ and turn them over.

3. Apply a pile-shifting to the sequence of piles, where each pile consists of (p_j, s_j, s'_j):



4. Reveal all s_j to identify the position of $\boxed{\heartsuit}$. Let the position of the revealed $\boxed{\heartsuit}$ be i'. The revealed cards are discarded.
5. P and V perform some operations on $p_{i'}$ (specified by another protocol). Note that $p_{i'} = p_i$ and the index i is kept secret thanks to the application of a pile-shifting in Step 3.
6. Apply a pile-shifting to the sequence of piles again.
7. Finally, reveal all s'_j to identify p_0. Restore the original sequence by rearranging p_j sequentially, starting from p_0.

3 General Protocol for Connected Component

We describe our main contribution, i.e., a general protocol for verifying various connected structures on a grid. After executing the protocol, V is convinced that P has colored a set of cells forming a connected component on a given grid, without revealing its specific shape.

Idea. The idea is an interactive protocol between P and V where P colors cells of a given grid sequentially. At each step, V confirms that the newly colored cell is adjacent to at least one of the previously colored cells. This ensures that the resulting shape forms a polyomino because there is no isolated cell colored by P. This idea was shown by Robert et al. [29] in 2022, and was extended to handle a simple cycle by Hand et al. [8] in 2024. Building on these results, we further generalize this idea to verify other geometric structures, such as a tree and path. The conditions for each newly colored cell confirmed by V are as follows:

Tree: It must be adjacent only to one of the previously colored cells. This ensures that the resulting shape is connected and contains no cycles.

Path: It must be adjacent only to the most recently colored cell. This ensures that the resulting shape has no cycles or branching.

Cycle: The condition is the same as a path until the last step. In the last step, the newly colored cell must be adjacent to exactly two cells, namely, the first and the most recently colored cells, to close the path into a single cycle.

However, the number of coloring steps inherently leaks the size of the resulting shape. To hide this information, we allow P to select whether to color a cell at each step using the chosen pile protocol. More precisely, let ℓ denote the number of colored cells in the solution. The protocol is executed for a fixed total of mn steps as follows:

1. P first colors an initial cell (in the solution).
2. P performs $mn - 2$ steps. In each step, P uses the chosen pile protocol to:
   - Select to color a new adjacent cell (repeated $\ell - 2$ times in total).
   - Select not to color a new cell (repeated $mn - \ell$ times in total).
3. Finally, P colors the last cell to close the path (for a cycle) or to complete the shape.

Here, the protocol always consists of mn steps and hence does not leak ℓ.

3.1 Description

The protocol takes as input a structure type $\sigma \in \{\mathsf{tree}, \mathsf{polyomino}, \mathsf{path}, \mathsf{cycle}\}$ and the size ℓ of the connected component, where ℓ is either a specific public value or unknown. After executing the protocol, V is convinced that P has placed a commitment on each cell of a given grid of size $m \times n$, where the set of the commitments form σ (of size ℓ).

The protocol proceeds as follows.

1. V places a commitment to $0 =$ ♣♥ on every cell of the grid. Let $c_{p,q}$ be the commitment placed on the cell (p, q), where (1,1) is the top-left cell.
2. V further places a commitment to $\perp =$ ♥♥ on every dummy[2] cell $(p, 0)$ for all $0 \le p \le m$ and $(0, q)$ for all $1 \le q \le n$.
3. P and V execute the chosen pile protocol introduced in Section 2.2 to select one commitment. Here, the input is all commitments $c_{p,q}$ arranged into a single sequence by relabeling them with a single index $j = q(n+1) + p$ in lexicographical order:

   c_0 ?? c_1 ?? $\cdots$ c_n ?? c_{n+1} ?? c_{n+2} ?? $\cdots$ c_{2n+1} ?? $\cdots$ $c_{(m+1)(n+1)-1}$??.

   Refer to Figure 2, taking Figure 1 as an example. Let c_i be the commitment selected by P. V performs the following operations on c_i.
   - **a.** Reveal c_i to confirm that its value is 0. Otherwise, V rejects the proof.
   - **b.** Change the value of c_i according to σ as follows.
     - If $\sigma \in \{\mathsf{tree}, \mathsf{polyomino}\}$, then change it from 0 to 1.
     - If $\sigma \in \{\mathsf{path}, \mathsf{cycle}\}$, then change it from 0 to $\star$.
4. P and V repeat the following procedure $\ell - 2$ times (if ℓ is given) or $mn - 2$ times (if ℓ is unknown): they execute the chosen pile protocol to select c_i as described in Step 3, and V performs the following operations on c_i.
   - **a.** Reveal c_i to confirm that its value is 0. Otherwise, V rejects the proof.
   - **b.** Change the value of c_i as in Step 3(b). If ℓ is unknown, then P and V prepare a commitment $c_{i'}$ to 1 (if $\sigma \in \{\mathsf{tree}, \mathsf{polyomino}\}$) or $\star$ (if $\sigma \in \{\mathsf{path}, \mathsf{cycle}\}$) and execute the chosen pile protocol to select either c_i or $c_{i'}$ and update it as c_i.
   - **c.** Identify the four commitments adjacent to c_i:

     c_{i-1} ?? c_{i+1} ?? $c_{i-(n+1)}$?? $c_{i+(n+1)}$??,

     where all indices are calculated modulo $(m+1)(n+1)$, respectively[3]. Then P and V execute the chosen pile protocol to select one commitment, denoted by c_j, among the four commitments and perform the following operation on c_j according to σ.
     - If $\sigma \in \{\mathsf{tree}\}$, then reveal c_j to confirm that the value of c_j is 1. In addition, for the other three commitments, reveal only their right cards to confirm that they are all ♥ (i.e., their values are not 1).
     - If $\sigma \in \{\mathsf{polyomino}\}$, then reveal c_j to confirm that the value of c_j is 1.
     - If $\sigma \in \{\mathsf{path}, \mathsf{cycle}\}$, then reveal c_j to confirm that the value of c_j is $\star$ and update it from $\star$ to 1. If and only if $\sigma \in \{\mathsf{cycle}\}$ and this is the first iteration, then this update is skipped. For the other three commitments, confirm the same thing as for $\sigma \in \{\mathsf{tree}\}$.

---

[2] Cells outside the grid.

[3] For example, in Figure 2, if c_{21} is selected, then four commitments of $(c_{20}, c_{22}, c_{16}, c_1)$ are adjacent to c_0.

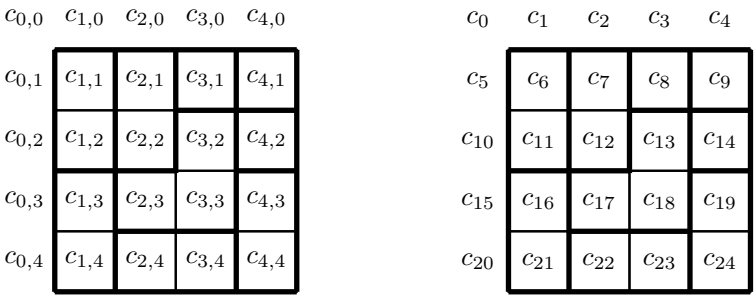


Figure 2 Example of the cell labeling in our general protocol: 2D (left) and 1D (right)

If any of the above conditions are not satisfied, then V rejects the proof.

5. Finally, as in Step 4, P and V execute the chosen pile protocol to perform the same operations on c_i and its neighbor c_j, with the following modifications:
   - **a.** This is the same as Step 4(a).
   - **b.** If $\sigma \in \{\mathsf{path}, \mathsf{cycle}\}$), then the value of c_i is changed to 1. Even if ℓ is unknown, this update is executed directly by V (as in Step 3(b)).
   - **c.** If $\sigma \in \{\mathsf{cycle}\}$), then for the other three commitments, P and V execute the chosen pile protocol again to select one commitment, denoted by c_k. Reveal c_k to confirm that its value is $\star$ and for the other two commitments, reveal only their right cards to confirm that they are all $\boxed{\heartsuit}$.

Efficiency. Our general protocol requires $O(mn)$ shuffles since it executes a constant number of operations repeated mn times. The number of cards required is $O(mn)$ since the protocol places a constant number of cards on each cell.

3.2 Proofs

We prove that the general protocol satisfies the desired properties.

▶ **Theorem 1.** *After executing our general protocol, a connected component σ of size ℓ is correctly represented by the placement of commitments on a given grid.*

Proof. Our protocol forces P to color a cell adjacent to previously colored cell sequentially, as indicated in Step 4(c). To do this, we confirm that at least one cell among the four adjacent cells has been colored. For $\sigma \in \{\mathsf{polyomino}\}$, such a check is trivially sufficient to guarantee the formation of a polyomino. For the other three shapes, we further confirm that exactly one cell among the four adjacent cells has been colored. This ensures that each newly colored cell connects to the existing component through only a single edge, which trivially prevents the formation of any cycles and thus guarantees an acyclic structure, and this completes the formation of a tree. For $\sigma \in \{\mathsf{path}, \mathsf{cycle}\}$, we restrict the candidate cell for the next coloring to be a neighbor of the most recently colored cell. This is done by marking the most recently colored cell with a special symbol, namely $\star$ as in Step 4(b). Since this prevents any branching from occurring, it trivially guarantees the formation of a path. Finally for $\sigma \in \{\mathsf{cycle}\}$, both the first cell and the most recently colored cell are marked with $\star$ as in Step 4(c). For the final cell, we confirm that it is adjacent to both $\star$ marks. This connects the two ends of a branchless path, trivially completing a simple cycle.

To handle boundary conditions, the grid is extended to $(m+1) \times (n+1)$ as in Step 2. These additional dummy cells act as a sentinel; when a selected cell lies on the boundary of

the original $m \times n$ grid, its adjacent cells that fall outside the boundary are represented by these sentinel cells. This prevents the figure from extending beyond the intended area. ◀

▶ **Theorem 2.** *The protocol reveals no information about the position, orientation, or specific shape of the connected component σ of size ℓ.*

Proof. First, when ℓ is unknown, the protocol repeats the steps $mn - 2$ times, which is publicly given. Using the chosen pile protocol, the protocol leaks no information about which cell P selected in each iteration. Moreover, both the confirmation of adjacent cells and the choice of whether to color a cell are performed using this protocol. Therefore, V cannot identify the location or the growth of the component. ◀

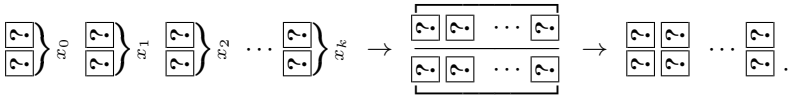
4 ZKP Protocol for Nurimeizu

Using our general protocol proposed in Section 3, we construct a card-based ZKP protocol for Nurimeizu.

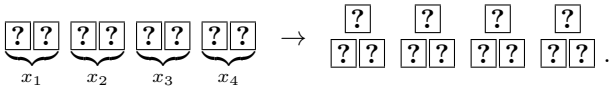
4.1 Description

We are ready to describe the protocol as follows.

1. **Tree**: P and V execute the general protocol, where $\sigma = \mathsf{tree}$ and ℓ is set as unknown. Here, we note that a Boolean value 0 represents black and 1 represents white. Then V reveals the commitments placed on all cells within the rooms including S-, G-, ◯-, and △-cells to confirm that their values are all 1.
2. **Room**: Let $x_i \in \{0, 1\}$ be the value of the commitment placed in the i-th cell of each room, where $i\,(> 0)$ is the cell index within the room (in any order). V confirms that $x_i = x_j$ for every i, j, as follows.
   a. Let k denote the size of the room. Place each commitment vertically as well as a commitment to $x_0 := 0$ and then apply a random bisection cut as follows:



   Let $r \in \{0, 1\}$ denote a random bit generated by the random bisection cut. Then each value x_i becomes $x_i \oplus r$.
   b. Reveal every commitment other than x_0 and confirm that $x_i \oplus r = x_j \oplus r$, i.e., $x_i = x_j$ for every i, j. Otherwise, V rejects the proof.
   c. Following the same procedures as in Steps 6 and 7 of the chosen pile protocol (Section 2.2), restore the original commitments to their original positions. Specifically, apply a random bisection cut again and reveal only the commitment to x_0 to retrieve the original state.
3. **Square of** 2×2: For each square of size 2×2, let $x_i \in \{0, 1\}$ be the value of the commitment placed in the i-th cell. V confirms that there exists at least one 0 and at least one 1 among $\{x_1, x_2, x_3, x_4\}$, as follows.
   a. The prover P appends a face-down card [?] to each commitment: [♥] to exactly one commitment to 0, [♥] to exactly one commitment to 1, and [♣] to the other two commitments.



b. Additionally append a commitment to x'_i to each commitment to x_i, where

$$(x'_1, x'_2, x'_3, x'_4) \coloneqq (0, 1, \star, \bot).$$

c. Apply a pile-scramble to the four piles of five cards.

d. Reveal the four cards that P appended in Step 3(a). If more than two $\boxed{\heartsuit}$s appear, then V rejects the proof.

e. Reveal the two commitments to which $\boxed{\heartsuit}$ was appended. If their values are not 0 and 1 (in any order), then V rejects the proof.

f. Restore the original commitments to their original positions using the commitments appended in Step 3(b).

4. **Shortest Path**: P and V execute the general protocol, where $\sigma = \mathsf{path}$ and ℓ is set as unknown. This is executed over the cells already colored in the previous "tree step," as exemplified in Figure 3. For this:
   - First, replace the commitment placed on every △-cell with a commitment to $\bot$ (as shown in Figures 3a and 3b). Skip Steps 1 and 2 of the general protocol.
   - In Step 3, instead of executing the chosen pile protocol, simply replace the commitment on S-cell with a commitment to $\star$ (Figure 3c).
   - For Step 4, we first confirm that the value of c_i is 1 in Step 4(a). Then in Steps 4(b) and 4(c), we allow P to select whether to replace c_i (the adjacent 1) and c_j (the current $\star$) with a commitment to $\star$ and a commitment to $\bot$, respectively, using the chosen pile protocol (Figures 3d and 3e). Moreover, we skip to confirm the other three commitments in Step 4(c).
   - Finally, in Step 5(b), we just change the value of c_i and c_j to $\star$ and $\bot$, respectively (Figure 3f).

   Finally, V reveals the commitments placed on G-cell and all ◯-cells to confirm that their values are $\star$ and $\bot$, respectively.

4.2 Proofs

We prove that our ZKP protocol for Nurimeizu satisfies the properties required for ZKPs.

▶ **Theorem 3.** *Our ZKP protocol for Nurimeizu satisfies the properties required for ZKPs.*

Proof. The proofs for other than the shortest path is implied or already appeared in the literature, and we focus on the shortest path. Recall that we employ the general protocol, in which P does not color a new path but instead traverses the pre-colored tree. First, every △-cell is replaced with $\bot$. This prevents the traversal from passing through these cells, as it is impossible for P to select a commitment to $\bot$. In each step of the traversal, we only need to prove that the newly selected cell is 1 and is adjacent to the previous $\star$. Since the Tree step has already guaranteed the connectivity of the colored area, this adjacency check is sufficient to ensure a continuous path. By changing the previous $\star$ to $\bot$ during the traversal, the protocol maintains a single "moving head," preventing any branching. All operations, including cell selection and adjacency verification, are performed using the chosen pile protocol. This ensures that V learns nothing about the specific route taken through the grid. Finally, V confirms that G-cell is $\star$. This proves a valid (shortest) path was formed while preserving zero-knowledge. ◀

1	0	1	1
1	0	1	0
1	1	1	1
1	0	0	1

(a) Initial state

1	0	1	1
1	0	1	0
1	1	1	$\perp$
$\perp$	0	0	1

(b) $\triangle$-cells replaced with $\perp$

$\star$	0	1	1
1	0	1	0
1	1	1	$\perp$
$\perp$	0	0	1

(c) S-cell replaced with $\star$

$\perp$	0	1	1
$\star$	0	1	0
1	1	1	$\perp$
$\perp$	0	0	1

(d) $\star$ and 1 replaced with $\perp$ and $\star$, respectively

$\perp$	0	$\star$	1
$\perp$	0	$\perp$	0
$\perp$	$\perp$	$\perp$	$\perp$
$\perp$	0	0	1

(e) State just before end

$\perp$	0	$\perp$	$\star$
$\perp$	0	$\perp$	0
$\perp$	$\perp$	$\perp$	$\perp$
$\perp$	0	0	1

(f) G-cell reached

Figure 3 Example of the value transitions during the shortest path verification

5 Conclusion

We constructed a general protocol to verify that a prover knows a valid solution of a given grid that forms a connected component, including a tree, path, cycle, and polyomino. Using this protocol, we proposed a ZKP protocol for Nurimeizu, where a valid solution requires a set of colored cells to form a tree, in which the specific route from the start to the goal must follow a unique shortest path. Our future work includes extending the protocol to handle multiple components, such as polyominoes.

References

1 Y. Abe, M. Iwamoto, and K. Ohta. Efficient private PEZ protocols for symmetric functions. In D. Hofheinz and A. Rosen, editors, *Theory of Cryptography*, volume 11891 of *LNCS*, pages 372–392, Cham, 2019. Springer.

2 J. Balogh, J. A. Csirik, Y. Ishai, and E. Kushilevitz. Private computation using a PEZ dispenser. *Theor. Comput. Sci.*, 306(1):69–84, 2003.

3 X. Bultel. Physical ring signature. In A. Z. Broder and T. Tamir, editors, *Fun with Algorithms*, volume 291 of *LIPIcs*, pages 7:1–7:18. Schloss Dagstuhl, 2024.

4 X. Bultel, J. Dreier, J.-G. Dumas, and P. Lafourcade. Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen. In E. D. Demaine and F. Grandoni, editors, *Fun with Algorithms*, volume 49 of *LIPIcs*, pages 8:1–8:20, Dagstuhl, Germany, 2016. Schloss Dagstuhl.

5 Y.-F. Chien and W.-K. Hon. Cryptographic and physical zero-knowledge proof: From Sudoku to Nonogram. In P. Boldi and L. Gargano, editors, *Fun with Algorithms*, volume 6099 of *LNCS*, pages 102–112, Berlin, Heidelberg, 2010. Springer.

6 J. Dreier, H. Jonker, and P. Lafourcade. Secure auctions without cryptography. In A. Ferro, F. Luccio, and P. Widmayer, editors, *Fun with Algorithms*, volume 8496 of *LNCS*, pages 158–170. Springer, 2014.

7 R. Gradwohl, M. Naor, B. Pinkas, and G. N. Rothblum. Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. *Theory of Computing Systems*, 44(2):245–268, 2009.

8 S. Hand, A. Koch, P. Lafourcade, D. Miyahara, and L. Robert. Efficient card-based ZKP for single loop condition and its application to Moon-or-Sun. *New Gener. Comput.*, 42:449–477, 2024.

9 K. Hatsugai, S. Ruangwises, K. Asano, and Y. Abe. NP-completeness and physical zero-knowledge proofs for Sumplete, a puzzle generated by ChatGPT. *New Gener. Comput.*, 42:429–448, 2024.

10 R. Ishikawa, E. Chida, and T. Mizuki. Efficient card-based protocols for generating a hidden random permutation without fixed points. In C. S. Calude and M. J. Dinneen, editors, *Unconventional Computation and Natural Computation*, volume 9252 of *LNCS*, pages 215–226, Cham, 2015. Springer.

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

12 S. Kaneko, P. Lafourcade, L. Mallordy, D. Miyahara, M. Puys, and K. Sakiyama. Secure voting protocol using balance scale. In K. Adi, S. Bourdeau, C. Durand, V. V. T. Tong, A. Dulipovici, Y. Kermarrec, and J. García-Alfaro, editors, *Foundations and Practice of Security*, volume 15532 of *LNCS*, pages 365–376, Cham, 2025. Springer.

13 S. Kaneko, P. Lafourcade, L.-B. Mallordy, D. Miyahara, M. Puys, and K. Sakiyama. Balance-based ZKP protocols for pencil-and-paper puzzles. In N. Mouha and N. Nikiforakis, editors, *Information Security*, volume 15257 of *LNCS*, pages 211–231, Cham, 2025. Springer.

14 A. Koch and S. Walzer. Foundations for actively secure card-based cryptography. In M. Farach-Colton, G. Prencipe, and R. Uehara, editors, *Fun with Algorithms*, volume 157 of *LIPIcs*, pages 17:1–17:23, Dagstuhl, Germany, 2020. Schloss Dagstuhl.

15 Y. Komano and T. Mizuki. Coin-based secure computations. *Int. J. Inf. Secur.*, 21:833–846, 2022.

16 Y. Komano and T. Mizuki. Physical zero-knowledge proof protocols for Topswops and Botdrops. *New Gener. Comput.*, 42:399–428, 2024.

17 Y. Komano and T. Mizuki. Card-based zero-knowledge proof protocols for pancake sorting. *IEICE Trans. Fundam.*, E109.A(3):371–382, 2026.

18 P. Lafourcade, D. Miyahara, T. Mizuki, L. Robert, T. Sasaki, and H. Sone. How to construct physical zero-knowledge proofs for puzzles with a "single loop" condition. *Theor. Comput. Sci.*, 888:41–55, 2021.

19 Y. Minamikawa and K. Shinagawa. Coin-based cryptographic protocols without hand operations. *IEICE Trans. Fundamentals*, E107.A(8):1178–1185, 2024.

20 D. Miyahara, H. Haneda, and T. Mizuki. Card-based zero-knowledge proof protocols for graph problems and their computational model. In Q. Huang and Y. Yu, editors, *Provable and Practical Security*, volume 13059 of *LNCS*, pages 136–152, Cham, 2021. Springer.

21 D. Miyahara, Y. Komano, T. Mizuki, and H. Sone. Cooking cryptographers: Secure multiparty computation based on balls and bags. In *Computer Security Foundations Symposium*, pages 389–404, NY, 2021. IEEE.

22 D. Miyahara, L. Robert, P. Lafourcade, and S. Kaneko. When locality implies globality: Card-based ZKP protocol for Shakashaka puzzle. In J. Iacono, editor, *Fun with Algorithms*, volume 366 of *LIPIcs*, pages 22:1–22:15, Dagstuhl, Germany, 2026. Schloss Dagstuhl. To Appear.

23 D. Miyahara, L. Robert, P. Lafourcade, and T. Mizuki. ZKP protocols for Usowan, Herugolf, and Five Cells. *Tsinghua Science and Technology*, 29(6):1651–1666, 2024.

24 T. Mizuki and H. Sone. Six-card secure AND and four-card secure XOR. In X. Deng, J. E. Hopcroft, and J. Xue, editors, *Frontiers in Algorithmics*, volume 5598 of *LNCS*, pages 358–369, Berlin, Heidelberg, 2009. Springer.

25 S. Murata, D. Miyahara, T. Mizuki, and H. Sone. Public-PEZ cryptography. In W. Susilo, R. H. Deng, F. Guo, Y. Li, and R. Intan, editors, *Information Security*, volume 12472 of *LNCS*, pages 59–74, Cham, 2020. Springer.

26 A. Nishimura, Y. Hayashi, T. Mizuki, and H. Sone. Pile-shifting scramble for card-based protocols. *IEICE Trans. Fundam.*, 101(9):1494–1502, 2018.

27 K. Nuida. Card-based protocol counting connected components of graphs. *New Gener. Comput.*, 43:18, 2025.

28 T. Otsuji, P. Fulla, and T. Fukunaga. NP-Completeness and physical zero-knowledge proof of Hotaru Beam. In Y. Chen, X. Gao, X. Sun, and A. Zhang, editors, *Computing and Combinatorics*, pages 239–251, Singapore, 2024. Springer.

29 L. Robert, D. Miyahara, P. Lafourcade, and T. Mizuki. Card-based ZKP for connectivity: Applications to Nurikabe, Hitori, and Heyawake. *New Gener. Comput.*, 40:149–171, 2022.

30 L. Robert, D. Miyahara, P. Lafourcade, and T. Mizuki. Physical ZKP protocols for Nurimisaki and Kurodoko. *Theor. Comput. Sci.*, 972:114071, 2023.

31 S. Ruangwises. Physical zero-knowledge proof for ball sort puzzle. In G. D. Vedova, B. Dundua, S. Lempp, and F. Manea, editors, *Unity of Logic and Computation*, volume 13967 of *LNCS*, pages 246–257, Cham, 2023. Springer.

32 S. Ruangwises and T. Itoh. Physical zero-knowledge proof for Ripple Effect. *Theor. Comput. Sci.*, 895:115–123, 2021.

33 S. Ruangwises and T. Itoh. Physical ZKP for connected spanning subgraph: Applications to Bridges puzzle and other problems. In I. Kostitsyna and P. Orponen, editors, *Unconventional Computation and Natural Computation*, pages 149–163, Cham, 2021. Springer.

34 S. Ruangwises and T. Itoh. How to physically verify a rectangle in a grid: A physical ZKP for Shikaku. In P. Fraigniaud and Y. Uno, editors, *Fun with Algorithms*, volume 226 of *LIPIcs*, pages 24:1–24:12, Dagstuhl, 2022. Schloss Dagstuhl.

35 S. Sasaki and K. Shinagawa. Physical zero-knowledge proof for Sukoro. *New Gener. Comput.*, 42:381–398, 2024.

36 T. Sasaki, D. Miyahara, T. Mizuki, and H. Sone. Efficient card-based zero-knowledge proof for Sudoku. *Theor. Comput. Sci.*, 839:135–142, 2020.

37 Y. Tamura, A. Suzuki, and T. Mizuki. Card-based zero-knowledge proof protocols for the 15-puzzle and the token swapping problem. In *ACM ASIA Public-Key Cryptography Workshop*, pages 11–22, New York, 2024. ACM.

38 I. Ueda, D. Miyahara, A. Nishimura, Y. Hayashi, T. Mizuki, and H. Sone. Secure implementations of a random bisection cut. *Int. J. Inf. Secur.*, 19(4):445–452, 2020.

39 R. Uehara. Computational complexity of puzzles and related topics. *Interdisciplinary Information Sciences*, 29(2):119–140, 2023.