TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 03/20 pp1651–1666
DOI: 10.26599/TST.2023.9010153
Volume 29, Number 6, December 2024

ZKP Protocols for Usowan, Herugolf, and Five Cells

Daiki Miyahara*, Léo Robert, Pascal Lafourcade, and Takaaki Mizuki

Abstract: A Zero-Knowledge Proof (ZKP) protocol allows a participant to prove the knowledge of some secret without revealing any information about it. While such protocols are typically executed by computers, there exists a line of research proposing physical instances of ZKP protocols. Up to now, many card-based ZKP protocols for pen-and-pencil puzzles, like Sudoku, have been designed. Those games, mostly edited by Nikoli, have simple rules, yet designing them in card-based ZKP protocols is non-trivial. In this work, we propose a card-based ZKP protocol for Usowan, a Nikoli game. In Usowan, for each room of a puzzle instance, there is exactly one piece of false information. The goal of the game is to detect this wrong data amongst the correct data and also to satisfy the other rules. Designing a card-based ZKP protocol to deal with the property of detecting a liar has never been done. In some sense, we propose a physical ZKP for hiding of a liar. This work extends a previous paper appearing in Ref. [1]. In this extension, we propose two other protocols, for Herugolf and Five Cells. The puzzles are specifically chosen because each of those three puzzles shares a common constraint, connectivity. However, showing the connected configuration cannot be done with generic approach and brings new construction to the existing connectivity ZKP protocol. Indeed, in Herugolf, the connectivity is handled with a given length of cell which is decremental (i.e., the length of each connected cell decreases by one at each step). For Five Cells, there is an additional step in the setup allowing to encode all the information needed to ensure a valid ZKP protocol.

Key words: Zero-Knowledge Proof (ZKP) protocol; playing cards; card-based cryptography; physical assumptions; Usowan; Herugolf; Five Cells

1 Intoduction

Suppose that someone wishes to prove the knowledge of a secret without revealing it. For instance, solving a puzzle (e.g., Sudoku) and convincing a verifier that this is indeed the solution without directly revealing the solution is hard. Such construction already exists and can be found in the field of cryptography. Indeed, a Zero-Knowledge Proof (ZKP) is a process where one party can prove the knowledge of information without revealing it.

A simple application to ZKP can be related to password authentication for a website; only the person with this password can access to sensitive data but it is preferable to never reveal the password. A second example can be given in electronic voting. In this system, voters want to enforce the correctness of mixing ballots (without revealing how the mix is done). Finally, crypto-currencies, such as Bitcoin, Monero, or

• Daiki Miyahara is with Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo 1828585, Japan. E-mail: miyahara@uec.ac.jp.
• Léo Robert is with MIS Lab, University of Picardy Jules Verne, Amiens 80000, France. E-mail: leo.robert@u-picardie.fr.
• Pascal Lafourcade is with Laboratory of Computer Science, Modelling and Systems Optimisation, University of Clermont Auvergne, Clermont-Ferrand 63178, France. E-mail: pascal.lafourcade@uca.fr.
• Takaaki Mizuki is with Cyberscience Center, Tohoku University, Sendai 9808578, Japan. E-mail: mizuki@cc.tohoku.ac.jp.
* To whom correspondence should be addressed.

Manuscript received: 2023-02-12; revised: 2023-09-06; accepted: 2023-12-04

Zcash, are eager to include a mechanism to enforce knowledge of some secrets without revealing it (e.g., for anonymous transactions).

More formally, a ZKP protocol is between two parties:

- a prover P who knows a solution s to a problem and
- a verifier V who wants to be sure that P is indeed in possession of the solution.

However, no information about s should leak during the protocol. Notice that some information can be recovered by the verifier without participating in the protocol. The information that cannot be leaked is the one directly linked with the protocol. Note also that some protocols are non-interactive meaning that the prover does not interact with the verifier in order to prove the knowledge of a secret. However, we only consider here interactive protocols where both parties are interacting during the protocol.

A ZKP protocol must guarantee three security properties:

- **Completeness:** If P knows s, then V is convinced when the protocol ends.
- **Soundness:** If P does not have the solution, then V will detect it during the protocol.
- **Zero-knowledge:** V learns nothing about s.

Most of the practical applications for ZKP protocols are executed by computers. We restrict ourselves by using only physical cards and envelopes, hence providing a more understandable approach of how ZKP protocols are designed.

Usowan. In Ref. [1], we presented a physical ZKP protocol for Usowan[2], which is a pencil puzzle played with a rectangular grid composed of numbered cells and white cells delimited by regions (thick edges).

The goal is to fill (in black) some cells:

We depict in Fig. 1 an initial Usowan grid with its solution and the corresponding rules in Fig. 2. Notice that numbered cell whose number is four (or more) is automatically a liar. Indeed, if there are four black cells around a numbered cell, then the numbered cell cannot be connected to other white cells. This information is not considered as a leak from the protocol since it is deducible from the initial setup (and not from an interaction during the protocol).

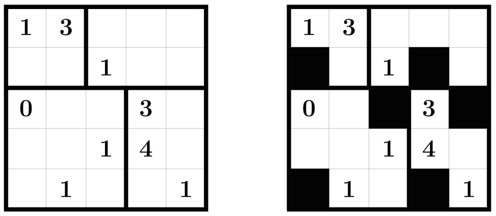


Fig. 1 Initial Usowan grid and its solution taken from Ref. [2].

Usowan rules:

(1) The numbered cells must remain white.

(2) The white cells form a connected shape.

(3) The black cells cannot connect vertically or horizontally.

(4) A numbered cell has the corresponding number of black cells around it (vertically or horizontally). However, each region has exactly one liar, i.e., the number of black cells is not equal to the numbered cell.

Fig. 2 Rules for Usowan[2].

While the hardness of the resolution for the underlying problem (here filling an Usowan grid) is not crucial for a physical protocol, a usual ZKP protocol needs to be based on a Non-deterministic Polynomial time (NP)-complete problem (otherwise the verifier could compute the secret in polynomial time). Fortunately, the NP-completeness of Usowan has been proved in Ref. [3]. This result ensures that there exists a ZKP protocol.

In this paper, we design two other protocols for two puzzles, Herugolf and Five Cells. Herugolf has been proven NP-complete in Ref. [4] and Five Cells in Ref. [5].

Five Cells. The goal of this puzzle is to divide the grid into blocks of cells, where the constraints are given in Fig. 3 and an example is illustrated in Fig. 4.

Five Cells rules:

(1) Each block is composed of exactly five cells.

(2) A number indicates how many lines there are around the cell (edges are also considered).

Fig. 3 Rules for Five Cells.

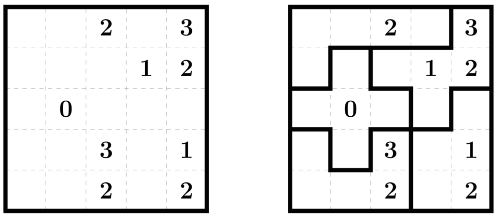


Fig. 4 Five Cells initial grid and its solution.

Herugolf. The goal of this puzzle is to draw arrows, in straightline, from center to center of cells. Each numbered cell must be connected with "H" (hole) cell. The constraints are given in Fig. 5 with an example in Fig. 6.

Contributions. We constructed in Ref. [1] a physical ZKP protocol for Usowan, giving the first application to detecting if a puzzle has flaws (i.e., the liar rule) while ensuring that the prover has the solution. It is the first physical ZKP protocol to prove that some information is incorrect among correct information. For this, we only use cards and envelopes. Moreover, we propose a trick that uses the rules of a Usowan grid in order to prove that exactly one piece of information is wrong in each room. We use several sub-protocols to verify the rules and propose a completely novel ZKP protocol.

In this paper, we propose two other protocols for two different puzzles, Five Cells and Herugolf. The link between the three puzzles presented here is the connectivity constraint, which enforces that each white cell must share at least one adjacent white cell. For Five Cells, the difficulty, in constructing a ZKP protocol, lies in changing the usual encoding for the connectivity problem. Indeed, our new protocol must solve the issue of having an encoding for connectivity and an encoding for delimiting region (i.e., encode lines between cells). For Herugolf, the main challenge is to design a ZKP protocol with a decrement for the connectivity length.

Herugolf rules:

(1) Show the movement of a circle by an arrow, with the tip of the arrow in the cell where it stops. The arrows can not cross other circles, H cells, or lines of other arrows.

(2) The first arrow from the numbered cell goes across that number of cells (indicated by the number). Other arrows are decreasing, in the number of cells, of 1.

(3) An arrow cannot leave the grid, or stop in the gray area (but an arrow can pass through it).

Fig. 5 Rules for Herugolf.

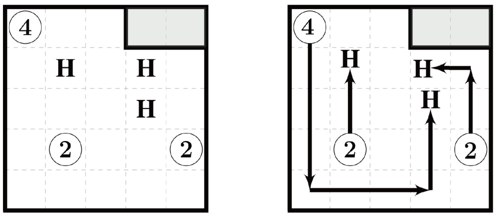


Fig. 6 Herugolf initial grid and its solution.

Note that this work is an extension of a previously accepted paper[1] where only the Usowan protocol is proposed.

Related work. Goldwasser et al.[6] proved that any NP-complete problem has its corresponding interactive ZKP protocol. Yet the generic approach has tremendous overhead leading to an impractical result. Works on implementing cryptographic protocols using physical objects are numerous, such as in Ref. [7]; or in Ref. [8], where a physical secure auction protocol is proposed. Other implementations have been studied using cards in Refs. [9, 10], polarising plates in Ref. [11], polygon cards in Ref. [12], a standard deck of playing cards in Ref. [13], using a PEZ dispenser[14, 15], using a dial lock[16], using a 15 puzzle[17], or using a tamper-evident seals[18–20]. Several ZKP protocols for other puzzles have been studied, such as Sudoku[21, 22], Akari[23], Takuzu[23, 24], Kakuro[23, 25], KenKen[23], Makaro[26, 27], Norinori[28], Nonogram[29, 30], Nurimisaki[31], Slitherlink[10], Suguru[32, 33], Nurikabe[34], Ripple Effect[35], Numberlink[36], Bridges[37], Shikaku[38], and Cryptarithmetic[39].

Very recently, Ruangwises[40] proposed a ZKP protocol for Five Cells, which is essentially different to our proposed one. This is concurrent and independent work.

Outline. In Section 2, we explain how to encode a grid with some cards in order to be able to construct our ZKP protocols. We also recall the existing card-based simple protocols of the literature that we use in our constructions. In Section 3, we briefly present our ZKP protocol for Usowan.

Next, Section 4 give the description of our ZKP protocol for Five Cells and its security proof.

Before concluding in Section 6, we present our ZKP protocol for Herugolf and its security proof in Section 5.

2 Preliminary

We explain the notations and sub-protocols used in our construction. We first introduce the general framework of card-based protocols, then present the existing sub-protocols used in our constructions.

Cards and encoding. The cards consist of clubs $\boxed{\clubsuit}$ and hearts $\boxed{\heartsuit}$ whose backs are identical $\boxed{?}$. We encode three colors {black, white, red} with the order

$$\begin{aligned}\boxed{\clubsuit}\boxed{\heartsuit} &\rightarrow \text{black},\\ \boxed{\heartsuit}\boxed{\clubsuit} &\rightarrow \text{white},\\ \boxed{\heartsuit}\boxed{\heartsuit} &\rightarrow \text{red}\end{aligned} \tag{1}$$

of two cards as follows:

We call a commitment a pair of face-down cards $\boxed{?}\boxed{?}$ corresponding to a color according to the above encoding rule. We also use the terms, a black commitment, a white commitment, and a red commitment. We sometimes regard black and white commitments as bit values, based on the following encoding:

$$\boxed{\clubsuit}\boxed{\heartsuit} \rightarrow 0, \quad \boxed{\heartsuit}\boxed{\clubsuit} \rightarrow 1 \tag{2}$$

For a bit $x \in \{0,1\}$, if a pair of face-down cards satisfies the encoding of Formula (2), we say that it is a commitment to x, denoted by $\underbrace{\boxed{?}\boxed{?}}_{x}$.

We also define two other encodings[41] as follows:

- $\clubsuit$-scheme: For $x \in \mathbb{Z}/p\mathbb{Z}$, there are p cards composed of $(p-1)$ $\heartsuit$ s and one $\clubsuit$ at position $(x+1)$ from the left. For example, 2 is represented as $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}$ in $\mathbb{Z}/4\mathbb{Z}$.
- $\heartsuit$-scheme: Same encoding is as above but the $\heartsuit$ and $\clubsuit$ are reversed. For instance, 2 is represented as $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}$ in $\mathbb{Z}/4\mathbb{Z}$.

2.1 Pile-shifting shuffle[12, 42]

This shuffling action means to shuffle piles of cards cyclically. More formally, given m piles, each of which consists of the same number of face-down cards, denoted by $(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$, applying a pile-shifting shuffle (denoted by $\langle \cdot \| \cdots \| \cdot \rangle$) results in $(\boldsymbol{p}_{s+1}, \boldsymbol{p}_{s+2}, \ldots, \boldsymbol{p}_{s+m})$,

$$\left\langle \underbrace{\boxed{?}}_{\boldsymbol{p}_1} \middle\| \underbrace{\boxed{?}}_{\boldsymbol{p}_2} \middle\| \cdots \middle\| \underbrace{\boxed{?}}_{\boldsymbol{p}_m} \right\rangle \rightarrow \underbrace{\boxed{?}}_{\boldsymbol{p}_{s+1}} \quad \cdots \quad \underbrace{\boxed{?}}_{\boldsymbol{p}_{s+m}},$$

where s is uniformly and randomly chosen from $\mathbb{Z}/m\mathbb{Z}$. We can simply implement this shuffling action using physical cases that can store a pile of cards, such as boxes and envelopes. A player (or players) cyclically shuffles them manually until everyone (i.e., P and V) loses track of the offset. Note that this shuffle can be "input-preserving" by writing ordered numbers at the back of envelopes. When all operations are done, players can put back all the commitments to their initial positions using those numbers. We implicitly use this when commitments need to be placed back to their initial positions after a shuffle.

2.2 Mizuki-Sone copy protocol[43]

We use Mizuki-Sone copy protocol to copy commitments, ensuring to V that this is indeed a correct copy of a given commitment (i.e., P cannot cheat with arbitrary value). Note that a red commitment is not considered in this protocol.

This description is a compact version of the original one[43]. Here, we use a pile-shifting shuffle in the following Step 2 instead of using a random bisection cut invented in Ref. [43].

The protocol proceeds as follows:

Step 1: Turn over all face-up cards and put the commitment to a above the four additional cards as follows:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit} \rightarrow \begin{array}{c}\underbrace{\boxed{?}\boxed{?}}_{a}\\ \underbrace{\boxed{?}\boxed{?}}_{0}\underbrace{\boxed{?}\boxed{?}}_{1}\end{array}.$$

Note that black-to-red represents 0, and red-to-black represents 1 according to Formula (2).

Step 2: Apply a pile-shifting shuffle as follows:

$$\left\langle \begin{array}{c}\boxed{?}\\ \boxed{?}\boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\ \boxed{?}\boxed{?}\end{array} \right\rangle \rightarrow \begin{array}{cc}\boxed{?} & \boxed{?}\\ \boxed{?}\boxed{?} & \boxed{?}\boxed{?}\end{array}.$$

Step 3: Reveal the two above cards and obtain two commitments to a as follows (note that negating a commitment is easy):

(a) If they are $\boxed{\clubsuit}\boxed{\heartsuit}$, then the four bottom cards are $\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{\overline{a}}$.

(b) If they are $\boxed{\heartsuit}\boxed{\clubsuit}$, then the four bottom cards are $\underbrace{\boxed{?}\boxed{?}}_{\overline{a}}\underbrace{\boxed{?}\boxed{?}}_{a}$.

2.3 Input-preserving Five-card trick[24]

This sub-protocol allows to compute an OR operation while being able to replace commitments back to their original configuration.

The sub-protocol proceeds as follows:

Step 1: Add helping cards and swap the two cards of the commitment to a, so that we have the negation $\overline{b}$ as follows:

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

Step 2: Rearrange the sequence of cards and turn over the face-up cards as follows:

$$\boxed{?}\boxed{?}\boxed{\heartsuit}\boxed{?}\boxed{?}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}$$
$$\to \begin{array}{c}\boxed{?}\boxed{?}\boxed{\heartsuit}\boxed{?}\boxed{?}\\ \boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\end{array} \to \begin{array}{c}\boxed{?}\boxed{?}\boxed{\heartsuit}\boxed{?}\boxed{?}\\ \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\end{array}.$$

Step 3: Regarding cards in the same column as a pile, apply a pile-shifting shuffle to the sequence as-follows:

$$\left\langle \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \right\rangle$$
$$\to \begin{array}{c}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\\ \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\end{array}.$$

Step 4: Reveal all the cards in the above row,

(a) If the resulting sequence is $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$ (up to cyclic shifts), then $a \vee b = 0$.

(b) If sit is $\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}$ (up to cyclic shifts), then $a \vee b = 1$.

Step 5: After turning over all the face-up cards, apply a pile-shifting shuffle.

Step 6: Reveal all the cards in the bottom row; then, the revealed cards should include exactly one $\boxed{\heartsuit}$.

Step 7: Shift the sequence of piles so that the leftmost card is the revealed $\boxed{\heartsuit}$, and swap the two cards of the commitment to $\bar{b}$ to restore commitments to a and b.

2.4 How to form a white polyomino

Before explaining the protocol, we need to describe two crucial sub-protocols first, namely the chosen pile protocol and the 4-neighbour protocol.

2.4.1 Chosen pile protocol[28]

This protocol allows P to choose a pile of cards without V knowing which one it is. Some operations can be done on this pile while all the commitments are replaced in their initial order.

This protocol is an extended version of the "chosen pile cut" proposed in Ref. [44]. Given m piles $(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$ with $2m$ additional cards, the chosen pile protocol enables a prover P to choose the i-th pile $\boldsymbol{p}_i$ (without revealing the index i) and revert the sequence of m piles to their original order after applying other operations to p_i, the detailed process is as follows:

Step 1: Using $(m-1)$ $\boxed{\clubsuit}$s and one $\boxed{\heartsuit}$, P places m face-down cards (denoted by Row 2) below the given piles, such that only the i-th card is $\boxed{\heartsuit}$. V further places m cards (denoted by Row 3) below the cards, such that only the first card is $\boxed{\heartsuit}$, as depicted in Fig. 7.

Step 2: Considering the cards in the same column as a pile, apply a pile-shifting shuffle to the sequence of piles.

Step 3: Reveal all the cards in Row 2 . Then, exactly one $\boxed{\heartsuit}$ appears, and the pile above the revealed $\boxed{\heartsuit}$ is the i-th pile (thus P can obtain $\boldsymbol{p}_i$). After this step is invoked, other operations are applied to the chosen pile. Then, the chosen pile is placed back to the i-th position in the sequence.

Step 4: Remove the revealed cards, i.e., the cards in Row 2. (Note, therefore, that we do not use the card $\boxed{\heartsuit}$ revealed in Step 3.) Then, apply a pile-shifting shuffle.

Step 5: Reveal all the cards in Row 3. Then, one $\boxed{\heartsuit}$ appears, and the pile above the revealed $\boxed{\heartsuit}$ is $\boldsymbol{p}_1$. Therefore, by shifting the sequence of piles (such that $\boldsymbol{p}_1$ becomes the leftmost pile in the sequence), we can obtain a sequence of piles whose order is the same as the original one without revealing any information about the order of the input sequence.

2.4.2 Sub-protocol: 4-neighbour protocol[34]

Given pq commitments placed on a $p \times q$ grid, a prover P has a commitment in mind, which we call a "target" commitment. The prover P wants to reveal the target commitment and another one that lies next to the target commitment (without revealing their exact positions). Here, a verifier V should be convinced that the second commitment is a neighbour of the first one (without knowing which one), as well as V should be able to confirm the colours of both the commitments. To handle the case where the target commitment is at the edge of the grid, we place commitments to red (as "dummy" commitments) in the left of the first column and below of the last row of the given grid (see Fig. 8 for example) to prevent P from choosing a commitment that is not a neighbour. Thus, the size of the expanded grid is $(p+1) \times (q+1)$.

Note that we do not place dummy commitments in the row above the first one and in the column right to the last one, because in the expanded grid of size

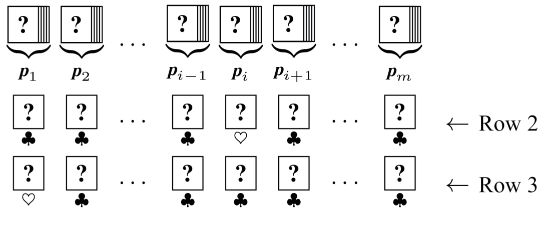


Fig. 7 Configuration at Step 1 in the chosen pile protocol.

$(p+1)(q+1)$ the row above the first one can be regarded as the last row, i.e., dummy commitments. Thus, we do not need dummy commitments placed in the row above the first one, which also holds for the column right to the last one.

The sub-protocol proceeds as follows:

Step 1: P and V pick the $(p+1)(q+1)$ commitments on the grid from left-to-right and top-to-bottom to make a sequence of commitments as follows:

$$\boxed{?}\boxed{?}\ \boxed{?}\boxed{?}\ \boxed{?}\boxed{?}\ \boxed{?}\boxed{?}\ \cdots\ \boxed{?}\boxed{?}.$$

Step 2: P uses the chosen pile protocol to reveal the target commitment.

Step 3: P and V pick all the four neighbours of the target commitment. Since a pile-shifting shuffle is a cyclic reordering, the distance between commitments are kept (up to a given modulo). That is, for a target commitment (not at the edge), the possible four neighbours are at distance one for the left or right one, and $p+1$ for the bottom or top one. Therefore, P and V can determine the positions of all the four neighbours.

Step 4: Among these four neighbours, P chooses one commitment using the chosen pile protocol and reveals it.

Step 5: P and V end the second and first chosen pile protocols.

2.4.3 Full protocol

Assume that there is a grid having $p \times q$ cells. Without loss of generality, P wants to arrange white commitments on the grid, such that they form a white-polyomino while V is convinced that the placement of commitments is surely a white-polyomino. The method is as follows.

Step 1: P and V place a black commitment (i.e., $\boxed{\clubsuit}\boxed{\heartsuit}$) on every cell and red commitments as mentioned in Section 2.4.2 so that they have $(p+1)(q+1)$ commitments on the board.

Step 2: V selects a black commitment on any cell that should be colored white by rules (e.g., numbered cells in a Usowan puzzle), and swaps the two cards constituting the commitment, so that it becomes a white commitment (recall the encoding as Formula (1)).

Step 3: P and V repeat the following steps exactly $pq-1$ times.

(a) P chooses one white commitment as a target and one black commitment among its neighbours using the 4-neighbour protocol; the neighbour is chosen, such that P wants to make it white.

(b) V reveals the target commitment. If it corresponds to white, then V continues; otherwise V aborts.

(c) V reveals the neighbour commitment (chosen P). If it corresponds to black, then P makes the neighbour white or keep it black (depending on P's choice) by executing the following steps; otherwise V aborts.[*]

i. If P wants to change the commitment, P places face-down club-to-heart pair below it; otherwise, P places a heart-to-club pair,

$$\boxed{?}\boxed{?} \rightarrow \begin{array}{cc}\boxed{?}&\boxed{?}\\ \boxed{?}&\boxed{?}\\ \clubsuit&\heartsuit\end{array} \quad \text{or} \quad \begin{array}{cc}\boxed{?}&\boxed{?}\\ \boxed{?}&\boxed{?}\\ \heartsuit&\clubsuit\end{array}.$$

ii. Regarding cards in the same column as a pile, V applies a pile-shifting shuffle to the sequence of piles,

$$\left\langle \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \right\rangle \rightarrow \begin{array}{cc}\boxed{?}&\boxed{?}\\ \boxed{?}&\boxed{?}\end{array}.$$

iii. V reveals the two cards in the second row. If the revealed right card is $\boxed{\heartsuit}$, then V swaps the two cards in the first row; otherwise V does nothing.

(d) P and V end the 4-neighbour protocol.

Step 4: P and V remove all the red commitments (i.e., dummy commitments), so that we have pq commitments on the board.

After this process, V is convinced that all the white commitments represent a white-polyomino. Therefore, this method allows a prover P to make a solution that only P has in mind, guaranteed to satisfy the connectivity constraint.

If the number of white cells in the final polyomino, say k, is public to a verifier V, it is sufficient that in Step 3, P and V repeat $(k-1)$ times and in Step 3c, and hence, V simply swaps the two cards constituting the neighbour commitment to make it white (without P's choice).

2.5 Sum in $\mathbb{Z}$[41]

We give an overview of the protocol described in Ref. [41] for adding elements in $\mathbb{Z}/2\mathbb{Z}$ with result in $\mathbb{Z}$. This protocol is needed for the liar rule 4[†].

Given commitments to $x_i \in \mathbb{Z}/2\mathbb{Z}$ for $i \in \{1,2,\ldots,n\}$ along with one $\boxed{\clubsuit}$ and one $\boxed{\heartsuit}$, the protocol produces

[*] One might think that this step can be simply achieved by letting P privately change the neighbour commitment. However, it might violate the soundness property because P can freely change it (e.g., into a red commitment), and hence, we have to additionally verify that P correctly changes the commitment.

their sum $S = \sum_{i=1}^{n} x_i$ in $\mathbb{Z}/(n+1)\mathbb{Z}$ encoded in the $\heartsuit$-scheme without revealing x_i. The computation is performed inductively; when starting by the two first commitments to x_1 and x_2, they are transformed into $x_1 - r$ and $x_2 + r$ encoded in the $\heartsuit$-scheme and $\clubsuit$-scheme, respectively, for uniformly random value $r \in \mathbb{Z}/3\mathbb{Z}$. Then $x_2 + r$ is revealed (no information about x_2 is revealed because r is random), and $x_1 - r$ is shifted by $x_2 + r$ positions to encode $(x_1 - r) + (x_2 + r) = x_1 + x_2$. Note that this result is in $\mathbb{Z}/(p+1)\mathbb{Z}$ (or simply $\mathbb{Z}$ because the result is less than or equal to p) for elements x_1 and x_2 in $\mathbb{Z}/p\mathbb{Z}$.

Let us describe the protocol. First, notice that black cells are assumed to be equal to 1 and white cells are equal to 0 (see Formulas (1) and (2)). Two commitments to x_1 and x_2 (either 0 or 1) will be changed to $x_1 + x_2$,

$$\underbrace{\boxed{?}\boxed{?}}_{x_1}\ \underbrace{\boxed{?}\boxed{?}}_{x_2}\ \boxed{\clubsuit}\boxed{\heartsuit} \rightarrow \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_1+x_2}.$$

Step 1: Swap the two cards of the commitment to x_1 and add a $\boxed{\clubsuit}$ face down to the right. Those three cards represent x_1 in the $\heartsuit$-scheme in $\mathbb{Z}/3\mathbb{Z}$,

$$\underbrace{\overset{\rightleftarrows}{\boxed{?}\boxed{?}}}_{x_1}\underset{\clubsuit}{\boxed{?}} \rightarrow \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_1}.$$

Step 2: Add a $\boxed{\heartsuit}$ on the right of the commitment to x_2. Those three cards represent x_2 in the $\clubsuit$-scheme in $\mathbb{Z}/3\mathbb{Z}$: $\underbrace{\boxed{?}\boxed{?}}_{x_2}\ \underset{\heartsuit}{\boxed{?}} \rightarrow \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_2}$.

Step 3: Obtain three cards representing $x_1 + r$ and those representing $x_2 - r$ for a uniformly random value $r \in \mathbb{Z}/3\mathbb{Z}$ as follows:

(a) Place in reverse order the three cards obtained in Step 2 below the three cards obtained in Step 1,

$$\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_1}\ \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_2} \rightarrow \begin{array}{c}\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_1}\\ \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{2-x_2}\end{array}.$$

(b) Apply a pile shifting shuffle as follows:

$$\left\langle \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\ \boxed{?}\end{array} \right\rangle \rightarrow \begin{array}{c}\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_1+r}\\ \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{2-x_2+r}\end{array}.$$

For a uniformly random value $r \in \mathbb{Z}/3\mathbb{Z}$, we obtain three cards representing $x_1 + r$ and those representing $2 - x_2 + r$.

(c) Reverse the order of the three cards representing $2 - x_2 + r$ to obtain those representing $x_2 - r$: $\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_1+r}\ \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_2-r}$.

Step 4: Reveal the three cards representing $x_2 - r$, and shift to the right the three cards representing $x_1 + r$ to obtain those representing $x_1 + x_2$ in the $\heartsuit$-scheme; apply the same routine for the remaining elements to compute the final sum.

Notice that we describe the protocol for a result in $\mathbb{Z}/3\mathbb{Z}$, but it is easily adaptable for a result in, let say, $\mathbb{Z}/q\mathbb{Z}$. Indeed, during Step 1, we add a single $\boxed{\clubsuit}$ to the first commitment and a single $\boxed{\heartsuit}$ to the second; thus for a sum that could be equal to $q - 1$, we add $q - 2$ $\boxed{\clubsuit}$ s to the first commitment and $q - 2$ $\boxed{\heartsuit}$ s to the second.

3 ZKP Protocol for Usowan

We present a card-based ZKP protocol for Usowan. Consider an Usowan instance composed as a rectangular grid of size $p \times q$.

3.1 Setup phase

The verifier V and prover P place black commitments on each cell of the $p \times q$ grid (also on the numbered cells) and place red commitments ("dummy" commitments) on the left of the first column and below the last row, so that we have $(p+1)(q+1)$ commitments, as depicted in Fig. 8.

3.2 Connectivity phase

We apply the sub-protocol introduced in Section 2.4 to form a white connected figure. After this phase, V is convinced that the white commitments are connected (i.e., Rule 2). Moreover, V reveals the commitments corresponding to numbered cells to check that they are indeed white (i.e., Rule 1). Notice that revealing

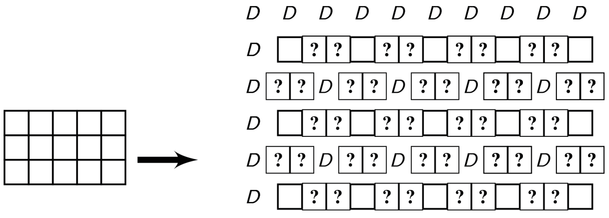


Fig. 8 Commitments setup for a Five Cells grid of 5×3. Commitments represent line between regions. The notation *D* refers to dummy commitments.

[†] A numbered cell has the corresponding number of black cells around it. However, each region has exactly one cell where the number of black cells is not equal to the numbered cell.

directly those commitments does reveal information about the solution (i.e., V learns that those cells are white), but this information is already known independently of the protocol.

3.3 Verification phases

There are two rules to check: black commitments cannot touch horizontally nor vertically (i.e., Rule 3) and each numbered cell has the corresponding number of black cells around it except for one liar in each region (i.e., Rule 4).

Lonely black. For each pair of adjacent commitments, V applies the five-card trick introduced in Section 2.3 to the two commitments to compute their disjunction. We consider here that a white commitment is equal to 1 while a black commitment is equal to 0 (see the encoding of Formula (2)). Hence, if the output is 1, then it means that at least one commitment is white, so V continues, otherwise V aborts (because the only case having output 0 is when there are two black commitments).

Liar. V needs to check that each numbered cell has the corresponding number of black cells around it except for exactly one liar in each region. We cannot simply check the number of black cells because it leaks information. Instead, we compute the sum of black cells in $\mathbb{Z}/5\mathbb{Z}$ introduced in Section 2.5 for all numbered cells in a region. However, we do not directly reveal the result but just the $(x-1)$-st card of the output sequence. This ensures that the sum is equal to or not to x instead of giving the actual sum.

It remains one sub-protocol to use because the addition is destructive; thus, we need to copy commitments sharing a numbered cell. The copy protocol is described in Section 2.2. We can now formally describe the liar verification. For every region, apply the following steps:

Step 1: For each cell that shares $k>1$ numbered cells, apply the copy protocol (introduced in Section 2.2) $k-1$ times.

Step 2: For each numbered cell, compute the addition of its four neighbors [‡]. Recall that the result is encoded as the ♡-scheme (see Section 2); thus, the result of the sum has a ♡ in its corresponding position (and all other cards are ♣s).

Step 3: For each sequence obtained in Step 1, pick the card in the position that corresponds to the number written on the numbered cell. The result must be kept secret (i.e., keep the cards face-down). For example, if the number is three, then the color of the fourth card from the left represents the sum as follows:

$$\begin{array}{ccc} & \boxed{b} & \\ \boxed{a} & \boxed{3} & \boxed{c} \\ & \boxed{d} & \end{array} \longrightarrow a+b+c+d = \underset{0}{\boxed{?}}\underset{1}{\boxed{?}}\underset{2}{\boxed{?}}\underset{\underset{\uparrow}{3}}{\boxed{?}}\underset{4}{\boxed{?}}.$$

Step 4: Shuffle and reveal all the cards previously chosen. If exactly one club is revealed, then continue (i.e., there is exactly one liar); otherwise aborts.

3.4 Security proofs

Our protocol needs to verify three security properties given as theorems. Note that the sub-protocols used from the literature have been proven secure, i.e., they are correct, complete, sound, and zero-knowledge.

Theorem 1 (Completeness) If P knows the solution of an Usowan grid, then P can convince V.

Proof P convinces V in the sense that the protocol does not abort, which means that all the rules are satisfied. The protocol can be split into two phases: (1) the connectivity phase and (2) the verification phase.

(1) Connectively phase: Since P knows the solution, the white cells are connected, and hence P can always select a black commitment in Step 3a of the protocol in Section 2.4.3 (dedicated to form a white polyomino) to swap it to white.

(2) Verification phase: For the lonely black verification, there is no configuration of two black cells that are touching horizontally nor vertically, hence for every pair of adjacent cells, there is always at least one white cell.

For the liar verification, there is exactly (in each region) one numbered cell surrounded by a different number of black cells. Suppose, without lost of generality, that the liar cell is equal to i in a given region (the same result could be applied for each other region). When the sum of the four neighbours is done, the card at position (from left) $i+1$ is $\boxed{\clubsuit}$, otherwise the numbered card is not a liar. Thus when revealing the cards at the last step of the protocal (Step 4), there is always a $\boxed{\clubsuit}$ card. ■

Theorem 2 (Soundness) If P does not provide a solution of the $p \times q$ Usowan grid, P is not able to convince V.

Proof Suppose that P does not provide a solution. If the white cells are not connected, then P cannot

[‡] For a numbered cell in the edge of the board, compute the addition of its three or two neighbors.

choose a neighbor commitment that P wants to change at Step 3c of the protocol in Section 2.4.3 (dedicated to form a white polyomino). If there are two black commitments touching (or more), then the five-card trick will output 0; hence, V will abort. Finally, if there is not one liar exactly in a given region, then the last step of the verification will reveal either no ♣ or at least two ♣ s; hence, V will abort. ■

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

Proof We use the same proof technique as in Ref. [45], namely the description of an efficient simulator that simulates the interaction between an honest prover and a cheating verifier. The goal is to produce an indistinguishable interaction from the verifier's view (with the prover). Notice that the simulator does not have the solution, but it can swap cards during shuffles. Informally, the verifier cannot distinguish between the distributions of two protocols, one that is run with the actual solution and one with random commitments. The simulator acts as follows:

- The simulator constructs a random connected white polyomino.
- During the lonely black verification, the simulator replaces the cards in the five-card trick introduced in Section 2.3 with ♡♣♡♣♡. While the latter sequence is randomly shifted, this ensure that the protocol continues.
- During the liar verification, the simulator simply replaces, in the last step, the cards to have exactly one ♣ and the rest as ♡ s. This ensure that there is exactly one liar in a given region, meaning that the protocol does not abort.

The simulated and real proofs are indistinguishable, hence V learns nothing from the connectivity and verification phases. Finally, we conclude that the protocol is zero-knowledge. ■

4 ZKP Protocol for Five Cells

This puzzle is different from the two others presented here, as the player solving it must fill, not the cells themselves, but the edges between them. So the first step is to provide a specific setup to handle this difference. We thus need to add commitments between each cell to encode lines forming regions; this is done by adding $(q+1)\times p$ commitments in columns and $(p+1)\times q$ commitments in rows. We depict this setup in Fig. 8 in case of $(p,q)=(5,3)$, where D denotes a red commitment as a dummy.

For encoding, we introduce another color, gray, to distinguish cells from lines. The four colors are encoded as follows:

$$\begin{aligned} ♣♡ &\rightarrow \text{black},\\ ♡♣ &\rightarrow \text{white},\\ ♡♡ &\rightarrow \text{red},\\ ♣♣ &\rightarrow \text{gray}. \end{aligned}$$

In our ZKP protocol, either white or gray commitment is placed on each cell, and either black or red commitment is placed in-between, i.e., the color of its second card represents either a cell or a line.

We informally define our protocol for Five Cells, illustrated in Fig. 9, as follows:

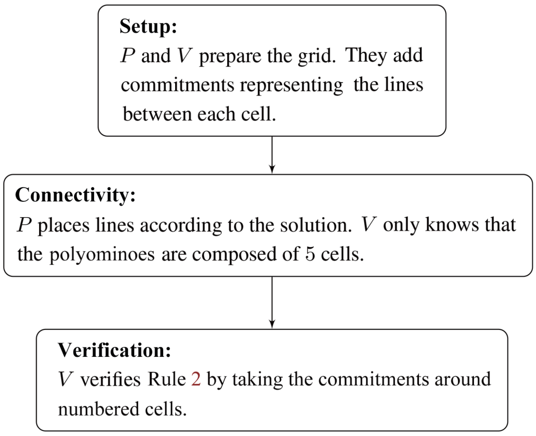
(1) P puts commitments between cells as described above and accordingly to its solution.

(2) V verifies the number rule by taking commitments around the cell; then shuffle them to reveal all the commitments. If the number of black commitments is the same as the number written in the cell, then V continues.

(3) The goal for P is to construct a pentomino without V knowing which shape it is. Since the total number of pentominoes is known ($pq/5$), the following constructive step is done for each pentomino: P chooses two adjacent commitments and V checks that there is no line in-between (then repeated 5 times to form a pentomino).

4.1 Checking the shape

Each delimited region must form a pentomino (i.e., composed of five connected cells). The shape is verified through the connectivity constraint using a variant of sub-protocol in Section 2.4.



Setup:
P and V prepare the grid. They add commitments representing the lines between each cell.

Connectivity:
P places lines according to the solution. V only knows that the polyominoes are composed of 5 cells.

Verification:
V verifies Rule 2 by taking the commitments around numbered cells.

Fig. 9 Overview of our Five Cells protocol.

However, we change the grid to add commitments representing the lines; so we cannot apply directly the sub-protocol of Section 2.4. Basically, the neighbours of a given cell have not the same distance when put in sequence but still follow the same evaluation. We give the corresponding distance in Table 1.

Notice that those distances are correct if the grid does not contain holes, i.e., there are commitments between each cell. This means that we need to add dummy commitments D to fill the grid. They are red commitments and only used to balance the grid to enforce the results of Table 1.

Finally, we must also add dummy commitments above the first row and on the left of the first column, as illustrated in Fig. 9. This comes from the fact that commitments at the edge of the intial grid have no neighbour, but to avoid leaking information, we need to add dummy commitments. In the original protocol (see Section 2.4), there is only one row/column of such dummies but here we need to add two rows/columns to keep the correct distances about neighbours. So in total, we go from a $p\times q$ grid to a $(2p+1+2)\times(2q+1+2)=(2p+3)\times(2q+3)$ grid. Indeed, commitments between each cell and the outer part give a $(2p+1)\times(2q+1)$ grid, and two rows (columns) on the bottom (left) part of the grid complete the final grid.

4.2 Our protocol for Five Cells

Setup. The initial grid is modified as explained in Section 4.1. Moreover, P applies its solution on the grid by putting black commitments to indicate edges and red commitments to indicate absence of edge, for each in-between cells. The commitments corresponding to actual cells of the grid are set to white by V.

Checking numbers (Rule 2 of Fig. 3). The verifier V checks the rule number by applying the following, on each numbered cell with number i:

Step 1: V picks the four closest commitments (corresponding to the presence or absence of edges) and shuffles them.

Step 2: V reveals all the commitments and checks if the number of black commitments is equal to i. If so, V continues, otherwise aborts.

Table 1 Distance (given for initial grid of a $p\times q$ size) from a cell when all commitments are put in sequence (as in Step 1 of protocol in Section 2.4.2).

Commitment	Classic	Variant
Right/Left	1	2
Up/Bottom	$p+1$	$2p+4$

Step 3: V puts back in their initial position the four commitments.

Checking pentominoes (Rule 1 of Fig. 3). We have now all the material to verify the pentominoes. Repeat the following steps $pq/5$ times:

Step 1: P chooses a white cell to begin its pentomino using the chosen-pile protocol.

Step 2: V reveals the commitment to check if it is white; if so, V turns it to gray and continues, otherwise aborts.

Step 3: P and V execute the 4-neighbour protocol and confirm that the target commitment is gray, but instead of taking one neighbour in each direction, they pick the two closest in each direction.

Step 4: V makes the second commitment to gray and reveal the first commitment; if it is red, then continues, otherwise aborts.

Step 5: P and V repeat Steps 3 and 4 until a pentomino is constructed.

Step 6: P and V execute the chosen-pile protocol and check that the chosen commitment (inside the pentomino) is gray; if so, V turns it to black and continues, otherwise aborts.

Step 7: V takes the first cards of every two closest commitments (of the previously chosen commitment) in each direction, shuffles the eight cards, and reveals them; if they are four s and four ♣s, then V continues; otherwise aborts.

Step 8: P and V repeat Steps 6–8 four times.

When all the pentominoes are constructed, V reveals the commitments corresponding to the cells of the grid (not the commitment corresponding to the lines). If all the cells are black, then V is convinced that Rule 1 (given in Fig. 3) is respected.

4.3 Security proofs for Five Cells

Our protocol needs to verify three security properties given as theorems. Note that the sub-protocols used from the literature have been proven secure, i.e., they are correct, complete, sound, and zero-knowledge.

Theorem 4 (Completeness) If P knows the solution of an Five Cells grid, then P can convince V.

Proof P convinces V in the sense that the protocol does not abort, which means that all the rules are satisfied. The protocol can be split into two phases: (1) verifying the number rule (Rule 2 in Fig. 3) and (2) checking the shape (Rule 1 in Fig. 3).

(1) Since P knows a solution, the number of black commitments (i.e., lines) around every numbered cell should be equal to that number. Thus, revealing the black commitments (after shuffling) as in the protocol, this rule is verified.

(2) Even if any of four cells in a pentomino are colored with red, P can always find a white cell next to one of the red cells, such that there is no line between them because P knows a solution. This means that P can always choose two white commitments starting from a red commitment via the 4-neighbour protocol at Step 3 of protocol in Section 2.4.2, such that the protocol never aborts. ■

Theorem 5 (Soundness) If P does not provide a solution of a $p \times q$ Five Cells grid, P is not able to convince V.

Proof Suppose that P does not provide a solution. We directly apply the soundness proof of Ref. [34] for the connectivity, since our variant could be seen as their connectivity sub-protocol (described in Section 2.4) by adding commitments between each cell (of the initial grid). This means that their connectivity construction can be modeled as our encoding (i.e., with additional commitments) by considering that there is no line delimiting region, hence no region is formed.

Having checked the connectivity constraint, there is an additional property to check (which is out of scope for Ref. [34]), the region are formed of five cells and no more (or less). If a region is not formed of exactly five cells, then our protocol will detect it. Indeed, by adding a color to the encoding (i.e., gray), there is a verification about all cells in a region. Each neighbour of a cell (inside a region, and detected by the gray color) has each neighbour either gray with no line (i.e., both are inside the region) or white/red with a line between them (the cell is at the edge). When the pentomino is checked, then its color is turned to black, meaning that P cannot continue this pentomino to add cells. ■

Theorem 6 (Zero-knowledge) V learns nothing about P's solution of the given grid G.

Proof We use the same proof technique as in Ref. [45], namely the description of an efficient *simulator* that simulates the interaction between an honest prover and a cheating verifier. The goal is to produce an indistinguishable interaction from the verifier's view (with the prover). Notice that the simulator does not have the solution, but it can swap cards during shuffles. Informally, the verifier cannot distinguish between the distributions of two protocols, one that is run with the actual solution and one with random commitments. The simulator acts as follows.

- For the connectivity phase, the simulator simply draws random pentominoes to construct a tilling of the grid. Notice that this is not the solution with overwhelming probability, but V will not abort at this point.
- Now, for each numbered cell, the simulator swaps card to the corresponding number being equal to the number of black commitments (which is possible since there is a shuffle). Thus V will not abort.

The simulated and real proofs are indistinguishable, and hence V learns nothing from our protocol, so we conclude that the protocol for Five Cells is zero-knowledge. ■

5 ZKP Protocol for Herugolf

The setup is straightforward since our protocol is constructive (P will construct its solution throughout the protocol). We emphasize that all cells are considered as white commitments. Additionally, we place a black card ♣ under each commitment; this will be used later to mark the tip of the arrow. Let us call this row the tip row.

For clarity, suppose we need to construct the arrows depicted in Fig. 10. The following steps are done for each numbered cell (but examplify with the configuration of Fig. 10):

Step 1: V takes the four (corresponding to ④) commitments, and the four cards in tip row, in each direction to form four (one for each direction) piles p_1, p_2, p_3, and p_4, shown in Fig. 11: Additionally, V reveals the commitment of the circle cell (here ④) and aborts if it is black; otherwise continues.

Step 2: P and V apply the chosen pile protocol to p_1, p_2, p_3, and p_4, so that P can choose in which

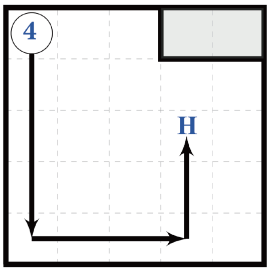


Fig. 10 Example of our protocol for Herugolf.

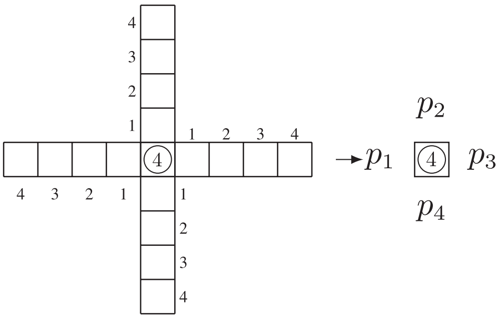
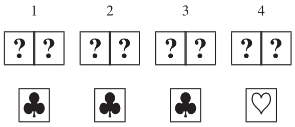


Fig. 11 Forming four piles for each directions.

direction the arrow is formed.

Step 3: V reveals all the commitments of the chosen pile; V aborts if there is at least one black commitment (meaning that there is already another arrow). Then V swaps all commitments so that all the commitments are now black.

Step 4: Before replacing back the piles, V replaces a red card under the last commitment (the number 4 of the pile) in tip row:

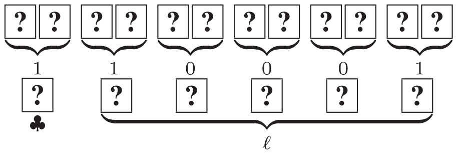


Step 5: P and V replace back all the commitments to their initial positions by ending the chosen pile protocol.

Step 6: V reveals the tip row corresponding to cells in the gray area. This ensures that Rule 3 is respected.

Step 7: Seeing Fig. 10, P wants to continue the path by constructing arrows of sizes 3 and 2, but an arrow of size 1 is not needed. Let k denote the length of the next arrow we consider (i.e., $k=3$ for this example). Let ℓ denote the length of the last arrow that P wants to construct (i.e., $\ell=2$). At this step, we create k commitments, each of which will be used at Step 9, as follows.

(a) V places k commitments to 0, places a commitment to 1 on the right side of them, and places $(k-1)$ commitments to 1 on the left side of them. Then V places $(k+2)$ cards of value ℓ encoded in the ♡-scheme and a face-down ♣ under them as follows:

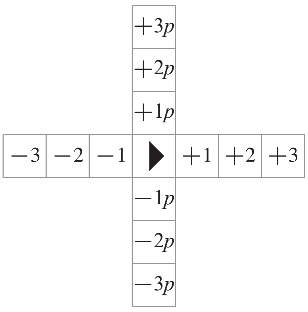


V confirms that $\ell \neq 0$ by revealing the leftmost card of $(k+2)$ cards representing ℓ; if it is a , then V aborts; otherwise, V continues.

(b) V applies a pile shifting shuffle to the two sequences of cards placed at the previous step, regarding cards in the same column as a pile.

(c) V reveals all the cards of the bottom sequence. Then exactly one ♡ is revealed, and take the commitment above the revealed ♡ as well as the $(k-1)$ commitments to the right (apart from cyclic rotation). We call these k commitments $\mathrm{CC}^1, \mathrm{CC}^2, \ldots, \mathrm{CC}^{\mathrm{k}}$ starting from the left.

Step 8: P wants to continue the path by constructing an arrow of size 3. The 4-neighbour protocol described in Section 2.4.2 is used except that 3 commitments are taken (except of just 1). The technique for Five Cells is used to know which commitment to take in the large sequence:

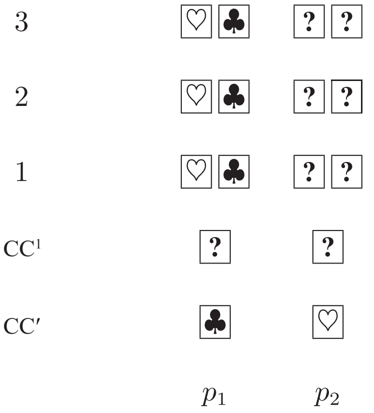


where the tip is denoted by the middle triangle, and p represents the number of line from a $p \times q$ grid. Note that V knows which cell corresponds to the tip by revealing a card in tip row.

Step 9: P applies the chosen pile protocol to choose its direction to form the next arrow, and prepares two identical commitments ♣ ♡ (if P wants not to draw the arrow anymore, P would have chosen ♡ ♣). For this, P takes CC^1 created at Step 7, and V copies it with the copy protocol of Section 2.2 (this ensures that the same commitments are used).

Step 10: V places the three commitments chosen at the previous step, CC^1, and additional cards, forming two piles p_1 and p_2 as follows:

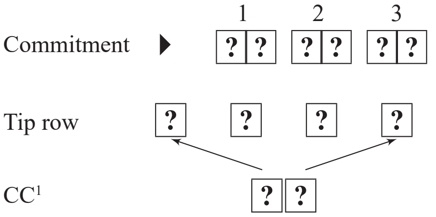
where CC' denotes a commitment specifying the "real" arrow. V then turns over all the face-up cards, shuffles the two piles p_1 and p_2, and reveals CC^1. Then either ♣ ♡ or ♡ ♣ is revealed, and V reveals the three commitments above the revealed ♡; V aborts if there is at least one black commitment. Finally, V



swaps them so that they are now black.

Step 11: V shuffles p_1 and p_2 again, and reveals CC′. Then we derive the "real" arrow above the revealed $\boxed{\heartsuit}$.

Step 12: The commitments are swapped with P's solution and without V's knowing if there is an arrow. Now, this step ensures that the tip of the arrow is marked on the newly created arrow (if so) or stays on the previous one. For this, V uses the second commitment of CC^1 (the first one has been used in the previous step) by replacing them with the cards of tip row. Concretely, V replaces the $\boxed{\heartsuit}$ card of tip row by the left card of CC^1, and the card under the third commitment by the right card of CC^1 as follows:



Step 13: Put back all the commitments in their respective position in the grid. V reveals the tip row corresponding to the gray area; this ensures that no tip is placed on those forbidden cells, thus ensuring Rule 3 given in Fig. 5.

Step 14: P and V repeat Steps 8 to 13 by decreasing the length of the arrow 1 and takeing a commitment sequentiall starting from CC^1 until reaching a length arrow equal to 1.

5.1 Verification phase

V simply reveals the tip row of the "H" cells to check that the tip is a $\boxed{\heartsuit}$ (meaning that each numbered cell is connected with a hole and that every arrow ends at a hole).

5.2 Security proofs for Herugolf

As before, our protocol needs to verify three security properties given as theorems. Note that the subprotocols used from the literature have been proven secure, i.e., they are correct, complete, sound, and zero-knowledge.

Theorem 7 (Completeness) If P knows the solution of an Herugolf grid, then P can convince V.

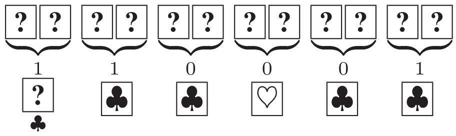
Proof P convinces V in the sense that the protocol does not abort, which means that all the rules are satisfied. In particular, each numbered cell is connected with a hole cell without crossing branches.

Firstly, given a numbered cell, an arrow is always depicted in a direction P chooses, and its tip is represented in tip row by executing Steps 1 to 6 in Section 5. The length of the arrow is always the same as the number written on the given numbered cell, because in Step 3, V changes the same number of white commitments into black ones.

Next, $\mathrm{CC}^1, \mathrm{CC}^2, \ldots, \mathrm{CC}^k$ are derived in Step 7. In the case of Fig. 10, each of them denotes the following commitment:

$$\mathrm{CC}^1 = \underbrace{\boxed{?}\boxed{?}}_{0},\quad \mathrm{CC}^2 = \underbrace{\boxed{?}\boxed{?}}_{0},\quad \mathrm{CC}^3 = \underbrace{\boxed{?}\boxed{?}}_{1},$$

because in Step 7(a), the sequence of cards is placed by V and P as follows:



and in Step 7(c), the commitment above the revealed $\boxed{\heartsuit}$ is taken as CC^1, i.e., a commitment to 0. CC^2 and CC^3 are the commitments to the right of CC^1, i.e., commitments to 0 and 1, respectively. For any case, CC^i is always a commitment to 1 if CC^{i-1} is a commitment to 1 for $2 \leqslant i \leqslant k$.

Finally, each of remaining arrows is depicted in the remaining Steps. Because the cell where P strats depicting an arrow is represented in tip row (and is updated in Step 12), P can always select such a cell using the 4-neighbour protocol in Step 8. If P does not want to depict an arrow anymore and take CC^i for some i in Step 10, then CC^i is always a commitment to 1 because P sets an appropriate value to ℓ in Step 7. Thus, once an arrow approaching a hole is depicted, no more arrows must be depicted, and V never aborts in Step 10 because the "dummy" arrow (i.e., commitments in p_1) is always revealed. In Step 12, tip row is updated by replacing the corresponding cards with CC^i for some i. Because CC^i is a commitment to 1 if and only if P does not want to depict an arrow, the cards in tip row remains unchanged even if executing Step 12. When CC^i is a commitment to 0, the cards in tip row are updated, so that the new tip is represented in an appropriate cell, and the old tip disappears. ■

Theorem 8 (Soundness) If P does not provide a solution of the $p \times q$ Herugolf grid, P is not able to convince V.

Proof We rely on the proof of Ref. [34] for the connectivity construction, i.e., arrows depicted by P are always connected for each numbered cell using the 4-neighbor protocol. Notice that Rule 3 is checked but revealing the tip row during the connectivity construction phase. ■

Theorem9 (Zero-knowledge) V learns nothing about P's solution of the given grid G.

Proof As in the previous proofs for the two other puzzles, we describe an efficient simulator. Informally, the verifier cannot distinguish between the distributions of two protocols, one that is run with the actual solution and one with random commitments. The simulator simply swaps cards to ensure that V will not abort. This is possible since each revealing step is preceded by a shuffle.

The simulated and real proofs are indistinguishable, hence V learns nothing from our protocol, so we conclude that the protocol for Herugolf is zero-knowledge. ■

6 Conclusion

We propose three ZKP protocols dedicated to convince a verifier that a prover has the solution without leaking any bit of information of the solution. Those protocols are designed for each of the following puzzles: Usowan, Five Cells, and Herugolf. Those three puzzles share a common connectivity constraint but with additional specific constraints.

The design of the ZKP protocol for Usowan uses mainly the sum sub-protocol, while Five Cells is designed through an hybrid encoding of the commitments (for the cells but also for the edge delimiting the region). The proposed ZKP protocol for Herugolf is somewhat in extension of the connectivity protocol which allows to construct connected figures of given length.

Acknowledgment

The second and third 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 (No. ANR-10-LABX-16-01), Finally, the French ANR project DECRYPT (No. ANR-18-CE39-0007) and SEVERITAS (No. ANR-20-CE39-0009) also subsidize this work. The first author was supported in part by Kayamori Foundation of Informational Science Advancement and JSPS KAKENHI (No. JP23H00479). The fourth author was supported in part by JSPS KAKENHI (Nos. JP21K11881 and JP23H00479).

References

[1] L. Robert, D. Miyahara, P. Lafourcade, and T. Mizuki, Hide a liar: Card-based ZKP protocol for usowan, in *Proc. 17th Int. Conf. Theory and Applications of Models of Computation*, Tianjin, China, 2022, pp. 201–217.

[2] Nikoli, Usowan, https://www.nikoli.co, 2023.

[3] C. Iwamoto and M. Haruishi, Computational complexity of Usowan puzzles, *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E101. A, no. 9, pp. 1537–1540, 2018.

[4] C. Iwamoto, M. Haruishi, and T. Ibusuki, Herugolf and Makaro are NP-complete, in *Proc. 9th Int. Conf. Fun with Algorithms*, La Maddalena, Italy, 2018, pp. 24: 1–24: 11.

[5] C. Iwamoto and T. Ide, Five cells and tilepaint are NP-complete, *IEICE Trans. Inf. Syst.*, vol. 105-D, no. 3, pp. 508–516, 2022.

[6] S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof-systems, in *Proc. Seventeenth Annu. ACM Symp. Theory of Computing*, Providence, RI, USA, 1985, pp. 291–304.

[7] T. Mizuki and H. Shizuya, Practical card-based cryptography, in *Proc. 7th Int. Conf. Fun with Algorithms*, Lipari Island, Italy, 2014, pp. 313–324.

[8] J. Dreier, H. Jonker, and P. Lafourcade, Secure auctions without cryptography, in *Proc. 7th Int. Conf. Fun with Algorithms*, Lipari Island, Italy, 2014, pp. 158–170.

[9] B. D. Boer, More efficient match-making and satisfiability the five card trick, in *Proc. Workshop on the Theory and Application of of Cryptographic Techniques*, Houthalen, Belgium, 1989, pp. 208–217.

[10] 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.*, vol. 888, pp. 41–55, 2021.

[11] K. Shinagawa, T. Mizuki, J. C. N. Schuldt, K. Nuida, N. Kanayama, T. Nishide, G. Hanaoka, and E. Okamoto, Secure computation protocols using polarizing cards,

IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol. E99. A, no. 6, pp. 1122–1131, 2016.

[12] K. Shinagawa, T. Mizuki, J. C. N. Schuldt, K. Nuida, N. Kanayama, T. Nishide, G. Hanaoka, and E. Okamoto, Cardbased protocols using regular polygon cards, *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E100. A, no. 9, pp. 1900–1909, 2017.

[13] T. Mizuki, Efficient and secure multiparty computations using a standard deck of playing cards, in *Proc. 15th Int. Conf. Cryptology and Network Security*, Milan, Italy, 2016, pp. 484–499.

[14] J. Balogh, J. A. Csirik, Y. Ishai, and E. Kushilevitz, Private computation using a PEZ dispenser, *Theor. Comput. Sci.*, vol. 306, nos. 1–3, pp. 69–84, 2003.

[15] Y. Abe, M. Iwamoto, and K. Ohta, Efficient private PEZ protocols for symmetric functions, in *Proc. 17th Theory of Cryptography Conf.*, Nuremberg, Germany, 2019, pp. 372–392.

[16] T. Mizuki, Y. Kugimoto, and H. Sone, Secure multiparty computations using a dial lock, in *Proc. 4th Int. Conf. Theory and Applications of Models of Computation*, Shanghai, China, 2007, pp. 499–510.

[17] T. Mizuki, Y. Kugimoto, and H. Sone, Secure multiparty computations using the 15 puzzle, in *Proc. First Int. Conf. Combinatorial Optimization and Applications*, Xi'an, China, 2007, pp. 255–266.

[18] T. Moran and M. Naor, Basing cryptographic protocols on tamper-evident seals, *Theor. Comput. Sci.*, vol. 411, no. 10, pp. 1283–1310, 2010.

[19] T. Moran and M. Naor, Polling with physical envelopes: A rigorous analysis of a human-centric protocol, in *Proc. 25th Annu. Int. Conf. the Theory and Applications of Cryptographic Techniques*, St. Petersburg, Russia, 2006, pp. 88–108.

[20] T. Moran and M. Naor, Split-ballot voting: Everlasting privacy with distributed trust, in *Proc. 14th ACM Conf. Computer and Communications Security*, Alexandria, VA, USA, 2007, pp. 246–255.

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

[22] S. Ruangwises, Two standard decks of playing cards are sufficient for a ZKP for Sudoku, *New Gener. Comput.*, vol. 40, no. 1, pp. 49–65, 2022.

[23] X. Bultel, J. Dreier, J. G. Dumas, and P. Lafourcade, Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen, in *Proc. 8th Int. Conf. Fun with Algorithms*, La Maddalena, Italy, 2016, pp. 8: 1–8: 20.

[24] D. Miyahara, L. Robert, P. Lafourcade, S. Takeshige, T. Mizuki, K. Shinagawa, A. Nagao, and H. Sone, Card-based ZKP protocols for Takuzu and Juosan, in *Proc. 10th Int. Conf. Fun with Algorithms*, Favignana Island, Italy, 2021, pp. 20: 1–20: 21.

[25] D. Miyahara, T. Sasaki, T. Mizuki, and H. Sone, Card-based physical zero-knowledge proof for Kakuro, *Fundam. Electron. Commun. Comput. Sci.*, vol. E102. A, no. 9, pp. 1072–1078, 2019.

[26] X. Bultel, J. Dreier, J. G. Dumas, P. Lafourcade, D. Miyahara, T. Mizuki, A. Nagao, T. Sasaki, K. Shinagawa, and H. Sone, Physical zero-knowledge proof for Makaro, in *Proc. 20th Int. Symp. on Stabilizing, Safety, and Security of Distributed Systems*, Tokyo, Japan, 2018, pp. 111–125.

[27] S. Ruangwises and T. Itoh, Physical ZKP for Makaro using a standard deck of cards, in *Proc. 17th Int. Conf. Theory and Applications of Models of Computation*, Tianjin, China, 2022, pp. 43–54.

[28] J. G. Dumas, P. Lafourcade, D. Miyahara, T. Mizuki, T. Sasaki, and H. Sone, Interactive physical zero-knowledge proof for Norinori, in *Proc. 25th Int. Computing and Combinatorics Conf.*, Xi'an, China, 2019, pp. 166–177.

[29] Y. F. Chien and W. K. Hon, Cryptographic and physical zero-knowledge proof: From Sudoku to Nonogram, in *Proc. 5th Int. Conf. Fun with Algorithms*, Ischia, Italy, 2010, pp. 102–112.

[30] S. Ruangwises, An improved physical ZKP for Nonogram, in *Proc. 15th Int. Conf. Combinatorial Optimization and Applications*, Tianjin, China, 2021, pp. 262–272.

[31] L. Robert, D. Miyahara, P. Lafourcade, and T. Mizuki, Card-based ZKP protocol for Nurimisaki, in *Proc. 24th Int. Symp. on Stabilizing, Safety, and Security of Distributed Systems*, Clermont-Ferrand, France, 2022, pp. 285–298.

[32] L. Robert, D. Miyahara, P. Lafourcade, and T. Mizuki, Physical zero-knowledge proof for suguru puzzle, in *Proc. 22nd Int. Symp. on Stabilizing, Safety, and Security of Distributed Systems*, Austin, TX, USA, 2020, pp. 235–247.

[33] L. Robert, D. Miyahara, P. Lafourcade, L. Libralesso, and T. Mizuki, Physical zero-knowledge proof and NP-completeness proof of Suguru puzzle, *Inf. Comput.*, vol. 285, p. 104858, 2022.

[34] L. Robert, D. Miyahara, P. Lafourcade, and T. Mizuki, Card-based ZKP for connectivity: Applications to Nurikabe, Hitori, and Heyawake, *New Gener. Comput.*, vol. 40, no. 1, pp. 149–171, 2022.

[35] S. Ruangwises and T. Itoh, Physical zero-knowledge proof for Ripple Effect, *Theor. Comput. Sci.*, vol. 895, pp. 115–123, 2021.

[36] S. Ruangwises and T. Itoh, Physical zero-knowledge proof for Numberlink puzzle and k vertex-disjoint paths problem, *New Gener. Comput.*, vol. 39, no. 1, pp. 3–17, 2021.

[37] S. Ruangwises and T. Itoh, Physical ZKP for connected spanning subgraph: Applications to bridges puzzle and other problems, in *Proc. 19th Int. Conf. Unconventional Computation and Natural Computation*, Espoo, Finland, 2021, pp. 149–163.

[38] S. Ruangwises and T. Itoh, How to physically verify a rectangle in a grid: A physical ZKP for shikaku, in *Proc. 11th Int. Conf. Fun with Algorithms*, Favignana Island, Italy, 2022, pp. 24: 1–24: 12.

[39] R. Isuzugawa, D. Miyahara, and T. Mizuki, Zero-knowledge proof protocol for Cryptarithmetic using dihedral cards, in *Proc. 19th Int. Conf. Unconventional Computation and Natural Computation*, Espoo, Finland, 2021, pp. 51–67.

[40] S. Ruangwises, Physical zero-knowledge proofs for five cells, in *Proc. 8th Int. Conf. Cryptology and Information Security in Latin America*, Quito, Ecuador, 2023, pp. 315–330.

[41] S. Ruangwises and T. Itoh, Securely computing the n-variable equality function with 2*n* cards, *Theor. Comput. Sci.*, vol. 887, pp. 99–110, 2021.

[42] A. Nishimura, Y. I. Hayashi, T. Mizuki, and H. Sone, Pile-shifting scramble for card-based protocols, *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E101. A, no. 9, pp. 1494–1502, 2018.

[43] T. Mizuki and H. Sone, Six-card secure AND and four-card secure XOR, in *Proc. Third Int. Workshop on Frontiers in Algorithmics*, Hefei, China, 2009, pp. 358–369.

[44] A. Koch and S. Walzer, Foundations for actively secure card-based cryptography, in *Proc. 10th Int. Conf. Fun with Algorithms*, Favignana Island, Italy, 2021, pp. 17: 1–17: 23.

[45] R. Gradwohl, M. Naor, B. Pinkas, and G. N. Rothblum, Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles, *Theory Comput. Syst.*, vol. 44, no. 2, pp. 245–268, 2009.

Léo Robert is an assistant professor at MIS Lab, University Picardy Jules Verne, France. He obtained the PhD degree in 2022 from University Clermont Auvergne (Clermont-Ferrand, France) and did a one-year postdoctoral research at XLIM, University of Limoges (France). His research interests mainly focus on provable security (protocols and primitives) and theoretical computer science.

Pascal Lafourcade is a full professor in computer security at University of Clermont Auvergne, and where he is a member of Laboratory of Computer Science, Modelling and Systems Optimisation. He received the PhD degree in 2006 from ENS Cachan (France) on the formal verification of cryptographic protocols. Then he did postdoctoral research at ETH Zurich (Switzerland) before getting an assistant professor position at Verimag in Grenoble (France) in 2007. He then held the industrial chair in digital trust at University of Auvergne (France) from 2013 to 2016. In 2016, he was recruited to the Computer Science Department, University of Clermont Auvergne (France). His research interests span a wide range of critical areas in computer security, including formal verification of cryptographic protocols and cryptography, electronic voting, blockchain, secure cloud computing, and symmetric cryptanalysis. He has authored over 140 papers with over 130 collaborators worldwide. He has also written several books on security. Additionally, his authored books likely serve as valuable resources for students and professionals in the field of security.

Daiki Miyahara received the BEng, MEng, and PhD degrees from Tohoku University, Japan in 2017, 2019, and 2021, respectively. He has been an assistant professor at The University of Electro-Communications since 2021, Japan. His research interests include theoretical computer science, information security, and cryptography.

Takaaki Mizuki received the PhD degree in information sciences from Tohoku University, Japan in 2000. He is currently a professor at Cyberscience Center, Tohoku University, Japan. His research interests include theoretical computer science and card-based cryptography.