



HAL
open science

ZKP protocols for Usowan, Herugolf and Five Cells

Daiki Miyahara, Léo Robert, Pascal Lafourcade, Takaaki Mizuki

► **To cite this version:**

Daiki Miyahara, Léo Robert, Pascal Lafourcade, Takaaki Mizuki. ZKP protocols for Usowan, Herugolf and Five Cells. Tsinghua Science and Technology, In press, 10.26599/TST.20xx.9010 . hal-04335703

HAL Id: hal-04335703

<https://u-picardie.hal.science/hal-04335703>

Submitted on 11 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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 [33]. 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: ZKP protocol; playing cards; card-based cryptography; physical assumptions; Usowan; Herugolf; Five Cells

1 Introduction

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 was done). Finally, crypto-currencies, such as Bitcoin, Monero, or 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

-
- 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, University of Picardy Jules Verne, France. E-mail: leo.robert@u-picardie.fr
 - Pascal Lafourcade is with LIMOS, University Clermont Auvergne, Clermont-Ferrand 63178, France
 - Takaaki Mizuki is with Cyberscience Center, Tohoku University, Sendai 9808578, Japan

* To whom correspondence should be addressed.

Manuscript received: 06-Sep-2023; accepted: 04-Dec-2023

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 are 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.

In [33], we presented a physical ZKP protocol for *Usowan* [1] 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).

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 NP-complete problem (otherwise

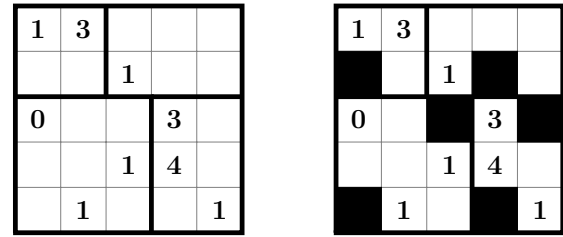


Fig. 1 Initial *Usowan* grid and its solution taken from [1].

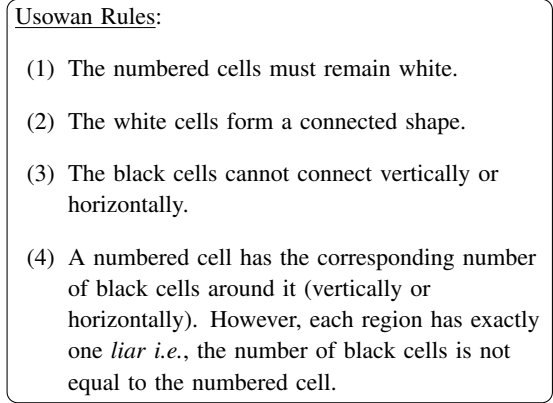


Fig. 2 Rules for *Usowan* [1].

the verifier could compute the secret in polynomial time). Fortunately, the NP-completeness of *Usowan* has been proved in [13]. 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 [14] and *Five Cells* in [15].

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.

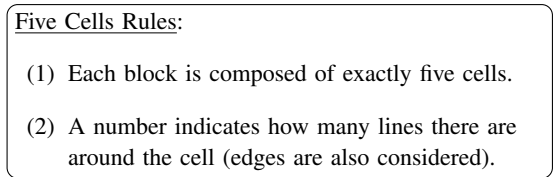


Fig. 3 Rules for *Five Cells*.

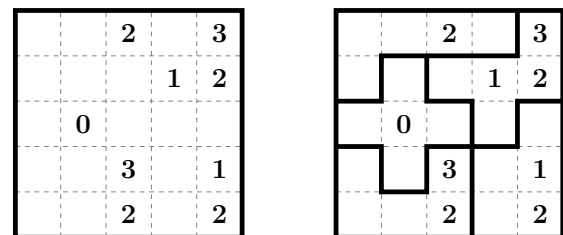


Fig. 4 *Five Cells* initial grid and its solution

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.

- 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.

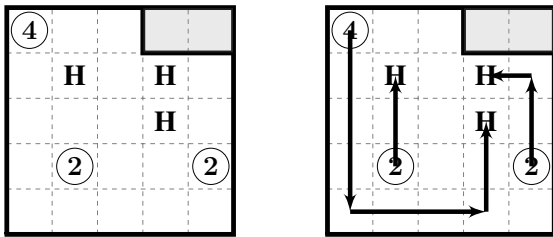


Fig. 6 Herugolf initial grid and its solution

Contributions. We constructed in [33] 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 used cards and envelopes. Moreover, we proposed 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 used several sub-protocols to verify the rules and proposed 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.

Note that this work is an extension of a previously accepted paper [33] where only the Usowan protocol was proposed.

Related Work. Goldwasser *et al.* [10] 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 [23]; or in [8] where a physical secure auction protocol was proposed. Other implementations have been studied using cards in [7, 17], polarising plates [43], polygon cards [44], a standard deck of playing cards [20], using a PEZ dispenser [2, 3], using a dial lock [21], using a 15 puzzle [22], or using a tamper-evident seals [25–27]. ZKP’s for several other puzzles have been studied such as Sudoku [35,42], Akari [4], Takuzu [4,18], Kakuro [4, 19], KenKen [4], Makaro [5, 41], Norinori [9], Nonogram [6, 34], Nurimisaki [32], Slitherlink [17], Suguru [29, 30], Nurikabe [31], Ripple Effect [37], Numberlink [36], Bridges [38], Shikaku [40] and Cryptarithmic [12].

Outline. In Sect. 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 Sect. 3, we briefly present our ZKP protocol for Usowan.

Next, Sect. 4 and Sect. 4.3 give the description of our ZKP protocol for Five Cells and its security proof, respectively.

Before concluding in Sect. 6, we present our ZKP protocol for Herugolf in 5 and its security proof in Sect. 5.2.

2 Preliminaries

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 ♣ and hearts ♥ whose backs are identical [?]. We encode three colors {black, white, red} with the order of two

cards as follows:

$$\begin{aligned} \begin{array}{|c|c|} \hline \clubsuit & \heartsuit \\ \hline \end{array} &\rightarrow \text{black,} \\ \begin{array}{|c|c|} \hline \heartsuit & \clubsuit \\ \hline \end{array} &\rightarrow \text{white,} \\ \begin{array}{|c|c|} \hline \heartsuit & \heartsuit \\ \hline \end{array} &\rightarrow \text{red.} \end{aligned} \quad (1)$$

We call a pair of face-down cards $\begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array}$ corresponding to a color according to the above encoding rule a *commitment* to the respective color. 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:

$$\begin{array}{|c|c|} \hline \clubsuit & \heartsuit \\ \hline \end{array} \rightarrow 0, \quad \begin{array}{|c|c|} \hline \heartsuit & \clubsuit \\ \hline \end{array} \rightarrow 1. \quad (2)$$

For a bit $x \in \{0, 1\}$, if a pair of face-down cards satisfies the encoding (2), we say that it is a commitment to x , denoted by $\underbrace{\begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array}}_x$.

We also define two other encodings [39]:

- \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 $\begin{array}{|c|c|c|c|} \hline \heartsuit & \heartsuit & \clubsuit & \heartsuit \\ \hline \end{array}$ in $\mathbb{Z}/4\mathbb{Z}$.
- \heartsuit -scheme: same encoding as above but the \heartsuit and \clubsuit are reversed. For instance, 2 is represented as $\begin{array}{|c|c|c|c|} \hline \clubsuit & \clubsuit & \heartsuit & \clubsuit \\ \hline \end{array}$ in $\mathbb{Z}/4\mathbb{Z}$.

2.1 Pile-shifting shuffle [28, 44]

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 $(\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m)$, applying a *pile-shifting shuffle* (denoted by $\langle \cdot \| \dots \| \cdot \rangle$) results in $(\mathbf{p}_{s+1}, \mathbf{p}_{s+2}, \dots, \mathbf{p}_{s+m})$:

$$\left\langle \begin{array}{|c|} \hline ? \\ \hline \end{array} \parallel \begin{array}{|c|} \hline ? \\ \hline \end{array} \parallel \dots \parallel \begin{array}{|c|} \hline ? \\ \hline \end{array} \right\rangle \rightarrow \begin{array}{|c|} \hline ? \\ \hline \end{array} \dots \begin{array}{|c|} \hline ? \\ \hline \end{array},$$

$\mathbf{p}_1 \quad \mathbf{p}_2 \quad \dots \quad \mathbf{p}_m \quad \mathbf{p}_{s+1} \quad \mathbf{p}_{s+m}$

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 to their initial positions all the commitments 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 [24]

We use it 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 [24]. Here, we use a pile-shifting shuffle in step 2 instead of using a random bisection cut invented in [24].

The protocol proceeds as follows.

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

$$\underbrace{\begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array}}_a \begin{array}{|c|c|c|c|} \hline \clubsuit & \heartsuit & \heartsuit & \clubsuit \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array} \begin{array}{|c|c|c|c|} \hline ? & ? & ? & ? \\ \hline \end{array} \begin{array}{c} a \\ 0 \quad 1 \end{array}.$$

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

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

$$\left\langle \begin{array}{|c|} \hline ? \\ \hline \end{array} \parallel \begin{array}{|c|} \hline ? \\ \hline \end{array} \right\rangle \rightarrow \begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array} \begin{array}{|c|c|c|c|} \hline ? & ? & ? & ? \\ \hline \end{array}.$$

- (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 $\begin{array}{|c|c|} \hline \clubsuit & \heartsuit \\ \hline \end{array}$, then the four bottom cards are $\underbrace{\begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array}}_a \underbrace{\begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array}}_{\bar{a}}$.
- (b) If they are $\begin{array}{|c|c|} \hline \heartsuit & \clubsuit \\ \hline \end{array}$, then the four bottom cards are $\underbrace{\begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array}}_{\bar{a}} \underbrace{\begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array}}_a$.

2.3 Input-preserving Five-card Trick [18]

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

- (1) Add helping cards and swap the two cards of the commitment to a so that we have the negation \bar{b} , as follows:

$$\underbrace{\begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array}}_a \underbrace{\begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array}}_b \rightarrow \underbrace{\begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array}}_a \underbrace{\begin{array}{|c|c|} \hline ? & ? \\ \hline \end{array}}_{\bar{b}} \begin{array}{|c|c|c|c|c|} \hline \heartsuit & \clubsuit & \clubsuit & \clubsuit & \clubsuit \\ \hline \end{array}.$$

- (2) Rearrange the sequence of cards and turn over the face-up cards as:

$$\begin{array}{|c|c|c|c|c|} \hline ? & ? & \heartsuit & ? & ? \\ \hline \end{array} \begin{array}{|c|c|c|c|c|} \hline \heartsuit & \clubsuit & \clubsuit & \clubsuit & \clubsuit \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|c|} \hline ? & ? & \heartsuit & ? & ? \\ \hline \end{array} \begin{array}{|c|c|c|c|c|} \hline ? & ? & ? & ? & ? \\ \hline \end{array}.$$

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

$$\left\langle \begin{array}{|c|} \hline ? \\ \hline ? \\ \hline \end{array} \parallel \begin{array}{|c|} \hline ? \\ \hline ? \\ \hline \end{array} \parallel \begin{array}{|c|} \hline ? \\ \hline ? \\ \hline \end{array} \parallel \begin{array}{|c|} \hline ? \\ \hline ? \\ \hline \end{array} \parallel \begin{array}{|c|} \hline ? \\ \hline ? \\ \hline \end{array} \right\rangle \rightarrow \begin{array}{|c|c|c|c|c|} \hline ? & ? & ? & ? & ? \\ \hline ? & ? & ? & ? & ? \\ \hline \end{array} .$$

- (4) Reveal all the cards in the above row.
- (a) If the resulting sequence is $\clubsuit\clubsuit\heartsuit\heartsuit\heartsuit$ (up to cyclic shifts), then $a \vee b = 0$.
- (b) If it is $\heartsuit\clubsuit\heartsuit\clubsuit\heartsuit$ (up to cyclic shifts), then $a \vee b = 1$.
- (5) After turning over all the face-up cards, apply a pile-shifting shuffle.
- (6) Reveal all the cards in the bottom row; then, the revealed cards should include exactly one \heartsuit .
- (7) Shift the sequence of piles so that the leftmost card is the revealed \heartsuit and swap the two cards of the commitment to 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 [9]

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 [16]. Given m piles $(\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m)$ with $2m$ additional cards, the *chosen pile protocol* enables a prover P to choose the i -th pile \mathbf{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 .

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

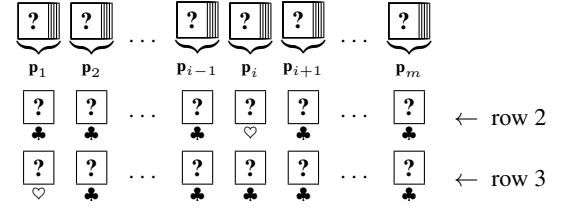


Fig. 7 Configuration at step 1 in the chosen pile protocol

- (2) Considering the cards in the same column as a pile, apply a pile-shifting shuffle to the sequence of piles.
- (3) Reveal all the cards in *row 2*. Then, exactly one \heartsuit appears, and the pile above the revealed \heartsuit is the i -th pile (thus P can obtain \mathbf{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.
- (4) Remove the revealed cards, *i.e.*, the cards in *row 2*. (Note, therefore, that we do not use the card \heartsuit revealed in Step 3.) Then, apply a pile-shifting shuffle.
- (5) Reveal all the cards in *row 3*. Then, one \heartsuit appears, and the pile above the revealed \heartsuit is \mathbf{p}_1 . Therefore, by shifting the sequence of piles (such that \mathbf{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 [31]

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 the below of the last row 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 $(p + 1)(q + 1)$ 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.

- (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:

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

- (2) P uses the chosen pile protocol (Sect. 2) to reveal the target commitment.
- (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.
- (4) Among these four neighbours, P chooses one commitment using the chosen pile protocol and reveals it.
- (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.

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

- (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 by 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:

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

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

$$\left\langle \begin{array}{c} \boxed{?} \\ \boxed{?} \end{array} \parallel \begin{array}{c} \boxed{?} \\ \boxed{?} \end{array} \right\rangle \rightarrow \begin{array}{c} \boxed{?} \boxed{?} \\ \boxed{?} \boxed{?} \end{array}.$$

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

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

- (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.

*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.

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} [39]

We give an overview of the protocol described in [39] 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 $i \in \{1, \dots, n\}$ along with one \clubsuit and one \heartsuit , the protocol produces 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, 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 Eqs. (1) and (2)). Two commitments to x_1 and x_2 (either 0 or 1) will be changed to $x_1 + x_2$:

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

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

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

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

- (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:

$$\underbrace{\boxed{?} \boxed{?} \boxed{?}}_{x_1} \underbrace{\boxed{?} \boxed{?} \boxed{?}}_{x_2} \rightarrow \underbrace{\boxed{?} \boxed{?} \boxed{?}}_{x_1} \underbrace{\boxed{?} \boxed{?} \boxed{?}}_{2-x_2}.$$

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

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

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 $x_2 - r$: $\underbrace{\boxed{?} \boxed{?} \boxed{?}}_{x_1+r} \underbrace{\boxed{?} \boxed{?} \boxed{?}}_{x_2-r}$.

- (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 described 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 the first step, we add a single \clubsuit to the first commitment and a single \heartsuit to the second; thus for a sum that could be equal to $q - 1$, we add $q - 2$ \clubsuit s to the first commitment and $q - 2$ \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.

3.2 Connectivity phase

We apply the sub-protocol introduced in Sect. 2.4 to form a white connected figure. After this phase, V is convinced that the white commitments are connected

(rule 2). Moreover, V reveals the commitments corresponding to numbered cells to check that they are indeed white (rule 1). Notice that revealing 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 (rule 3) and each numbered cell has the corresponding number of black cells around it except for one *liar* in each region (rule 4).

Lonely black. For each pair of adjacent commitments, V applies the five-card trick introduced in Sect. 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 (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 of 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 Sect. 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 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 Sect. 2.2. We can now formally describe the liar verification. For every region, apply the following steps:

- (1) For each cell that shares $k > 1$ numbered cells, apply the copy protocol (introduced in Sect. 2.2) $k-1$ times.
- (2) For each numbered cell, compute the addition of its four neighbors[†]. Recall that the result is encoded

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

as the \heartsuit -scheme (see Sect. 2); thus, the result of the sum has a \heartsuit in its corresponding position (and all other cards are \clubsuit s).

- (3) For each sequence obtained in the previous step, 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:

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

- (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) Since P knows the solution, the white cells are connected and hence P can always select a black commitment in step 3a to swap it to white.

(2) 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 loss 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 \clubsuit otherwise the numbered card is not a liar. Thus when revealing the cards at the last step, there is always a \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 choose a neighbor commitment that P wants to change at step 3c. 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 [11], 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 Sect. 2.3 with $\heartsuit \clubsuit \heartsuit \clubsuit \heartsuit$. 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 and 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:

\clubsuit	\heartsuit	→	black,
\heartsuit	\clubsuit	→	white,
\heartsuit	\heartsuit	→	red,
\clubsuit	\clubsuit	→	gray.

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 ($\frac{p \times q}{5}$), 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 Sect. 2.4.

However, we change the grid to add commitments representing the lines; so we cannot apply directly the sub-protocol of Sect. 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 cells. 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.

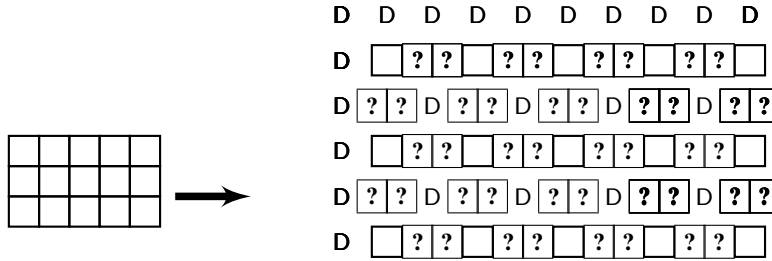


Fig. 8 Commitments setup for a Five Cells grid of 5×3 . Commitments represent line between regions. The notation D refers to dummy commitments.

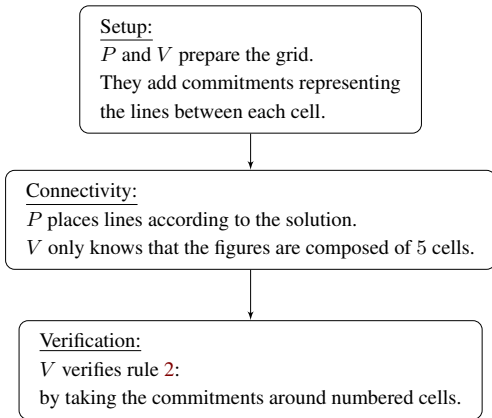


Fig. 9 Overview of our Five Cells protocol.

Finally, we must also add dummy commitments above the first row and on the left of the first column. This comes from the fact that commitments at the edge of the initial grid have no neighbour, but to avoid leaking information, we need to add dummy commitments. In the original protocol (see Sect. 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 (commitments between each cell and the outer part giving $(2p + 1) \times (2q + 1)$ grid and then two rows/columns on the left and bottom part of the grid).

	classic	variant
right/left	1	2
up/bot	$p + 1$	$2p + 4$

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).

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 is set to white by V .

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

- (1) V picks the four closest commitments (corresponding to the presence or absence of edges) and shuffles them.
- (2) V reveals all the commitments and checks if the number of black commitments is equal to i . If so, V continues otherwise aborts.
- (3) V puts back in their initial position the four commitments.

Checking pentominoes (rule 1). We have now all the material to verify the pentominoes. Repeat the following steps $\frac{p \times q}{5}$ times:

- (1) P chooses a white cell to begin its pentomino using the chosen-pile protocol.
- (2) V reveals the commitment to check if it is white; if so, V turns it to gray and continues, otherwise aborts.
- (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.
- (4) V makes the second commitment to gray and reveal the first commitment; if it is red, then continues, otherwise aborts.
- (5) P and V repeat steps 3 and 4 until a pentomino is constructed.

- (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.
- (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 \heartsuit s and four \clubsuit s, then V continues; otherwise, aborts.
- (8) P and V repeat steps 6, 7, and 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 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 and (2) checking the shape. (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 such that the protocol never aborts. ■

Theorem 5 (Soundness) If P does not provide a solution of the $p \times q$ 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 [31] for the connectivity since our variant could be seen as their connectivity sub-protocol (described in Sect. 2.4) by adding commitments between each initial cell.

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 [31]), the region is 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 [11], 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 tiling 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. ■

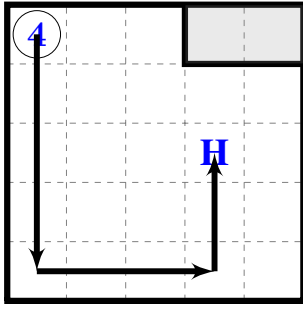


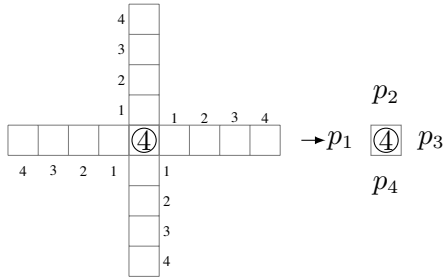
Fig. 10 Example of our protocol for Herugolf

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 \clubsuit 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 exemplify with the configuration of Fig. 10):

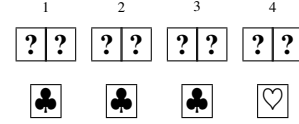
- (1) V takes the four (corresponding to $\textcircled{4}$) commitments, and the four cards in tip row, in each direction to form four (one for each direction) piles p_1, p_2, p_3, p_4 :



Additionally, V reveals the commitment of the circle cell (here $\textcircled{4}$) and aborts if it is black; otherwise continues.

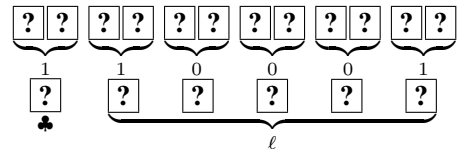
- (2) P and V apply the chosen pile protocol to p_1, p_2, p_3, p_4 so that P can choose in which direction the arrow is formed.
- (3) V reveals all the commitments of the chosen pile; V aborts if there is at least one black commitment (meaning that there is already another arrow). Then V swaps all commitments so that all the commitments are now black.

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



- (5) P and V replace back all the commitments to their initial positions by ending the chosen pile protocol.
- (6) V reveals the tip row corresponding to cells in the gray area. This ensures that rule 3 is respected.
- (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 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.

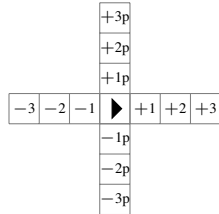
- (a) V prepares k commitments to 0, places a commitment to 1 on the right side of them, and places $k - 1$ commitments to 1 on the left side of them. Then V places $k + 2$ cards representing ℓ encoded in the \heartsuit -scheme and a face-down \clubsuit under them as follows.



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

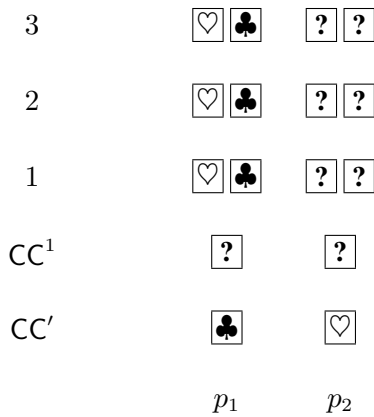
- (b) V applies a pile shifting shuffle to the two sequences of cards placed at the previous step, regarding cards in the same column as a pile.
- (c) V reveals all the cards of the bottom sequence. Then exactly one \heartsuit is revealed, and take the commitment above the revealed \heartsuit as well as the $k - 1$ commitments to the right (apart from cyclic rotation). We call these k commitments CC^1, CC^2, \dots, CC^k starting from the left.

- (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:



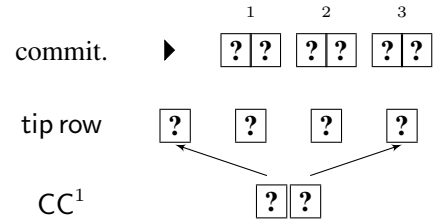
where the tip is denoted by the middle triangle. Note that V knows which cell corresponds to the tip by revealing a card in tip row.

- (9) P applies the chosen pile protocol to choose its direction to form the next arrow, and prepares two identical commitments $\clubsuit\heartsuit$ (if P wants to not draw the arrow anymore, he would have chosen $\heartsuit\clubsuit$). 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).
- (10) V places the three commitments chosen at the previous step, CC^1 , and additional cards, forming two piles p_1, p_2 :



where CC' denotes a commitment specifying the “real” arrow. V then turns over all the face-up cards, shuffles the two piles p_1, p_2 , and reveals CC^1 . Then either $\clubsuit\heartsuit$ or $\heartsuit\clubsuit$ is revealed, and V reveals the three commitments above the revealed \heartsuit ; V aborts if there is at least one black commitment. Finally, V swaps them so that they are now black.

- (11) V shuffles p_1 and p_2 again and reveals CC' . Then we derive the “real” arrow above the revealed \heartsuit .
- (12) The commitments are swapped with P 's solution and without V 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 \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 :



- (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.
- (14) P and V repeat steps 8 to 13 by decreasing the length of the arrow 1 and taking a commitment sequentially 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 \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 sub-protocols 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 .

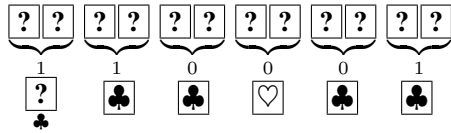
Proof P convinces V in the sense that the protocol does not abort, which means that all the rules are satisfied. In particular, each numbered cell is connected with a *hole* cell without crossing branches.

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. 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, CC^1, CC^2, \dots, CC^k are derived in step 7. In the case of Fig. 10, each of them denotes the following commitment:

$$CC^1 = \underbrace{[?][?]}_0, \quad CC^2 = \underbrace{[?][?]}_0, \quad CC^3 = \underbrace{[?][?]}_1,$$

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 \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 $2 \leq i \leq k$.

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 [31] for the connectivity construction, *i.e.*, arrows P depicts 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. ■

Theorem 9 (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 and 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 design 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

We thank the anonymous referees, whose comments have helped us improve the presentation of the paper. The first 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 (ANR-10-LABX-16-01). Finally, the French ANR project DECRYPT (ANR-18-CE39-0007) and SEVERITAS (ANR-20-CE39-0009) also subsidize this work. The second author was supported in part by Kayamori Foundation of Informational Science

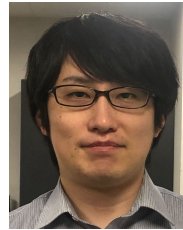
Advancement and JSPS KAKENHI Grant Number JP23H00479. The fourth author was supported in part by JSPS KAKENHI Grant Numbers JP21K11881 and JP23H00479.

References

- [1] Nikoli, Usowan.
- [2] Yoshiki Abe, Mitsugu Iwamoto, and Kazuo Ohta. Efficient private PEZ protocols for symmetric functions. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography*, volume 11891 of *LNCS*, pages 372–392, Cham, 2019. Springer.
- [3] József Balogh, János A. Csirik, Yuval Ishai, and Eyal Kushilevitz. Private computation using a PEZ dispenser. *Theor. Comput. Sci.*, 306(1-3):69–84, 2003.
- [4] Xavier Bultel, Jannik Dreier, Jean-Guillaume Dumas, and Pascal Lafourcade. Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen. In Erik D. Demaine and Fabrizio Grandoni, editors, *Fun with Algorithms*, volume 49 of *LIPICs*, pages 8:1–8:20, 2016.
- [5] Xavier Bultel, Jannik Dreier, Jean-Guillaume Dumas, Pascal Lafourcade, Daiki Miyahara, Takaaki Mizuki, Atsuki Nagao, Tatsuya Sasaki, Kazumasa Shinagawa, and Hideaki Sone. Physical zero-knowledge proof for Makaro. In Taisuke Izumi and Petr Kuznetsov, editors, *SSS 2018*, volume 11201 of *LNCS*, pages 111–125, Cham, 2018. Springer.
- [6] Yu-Feng Chien and Wing-Kai Hon. Cryptographic and physical zero-knowledge proof: From Sudoku to Nonogram. In Paolo Boldi and Luisa Gargano, editors, *Fun with Algorithms*, volume 6099 of *LNCS*, pages 102–112, Berlin, Heidelberg, 2010. Springer.
- [7] Bert den Boer. More efficient match-making and satisfiability: The five card trick. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *EUROCRYPT 1989*, volume 434 of *LNCS*, pages 208–217, Berlin, Heidelberg, 1989. Springer.
- [8] Jannik Dreier, Hugo Jonker, and Pascal Lafourcade. Secure auctions without cryptography. In Alfredo Ferro, Fabrizio Luccio, and Peter Widmayer, editors, *Fun with Algorithms*, volume 8496 of *LNCS*, pages 158–170, Cham, 2014. Springer.
- [9] Jean-Guillaume Dumas, Pascal Lafourcade, Daiki Miyahara, Takaaki Mizuki, Tatsuya Sasaki, and Hideaki Sone. Interactive physical zero-knowledge proof for Norinori. In Ding-Zhu Du, Zhenhua Duan, and Cong Tian, editors, *Computing and Combinatorics*, volume 11653 of *LNCS*, pages 166–177, Cham, 2019. Springer.
- [10] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *STOC 1985*, pages 291–304, New York, 1985. ACM.
- [11] Ronen Gradwohl, Moni Naor, Benny Pinkas, and Guy N. Rothblum. Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. *Theory Comput. Syst.*, 44(2):245–268, 2009.
- [12] Raimu Isuzugawa, Daiki Miyahara, and Takaaki Mizuki. Zero-knowledge proof protocol for Cryptarithmic using dihedral cards. In Irina Kostitsyna and Pekka Orponen, editors, *UCNC 2021*, volume 12984 of *LNCS*, pages 51–67, Cham, 2021. Springer.
- [13] Chuzo Iwamoto and Masato Haruishi. Computational complexity of Usowan puzzles. *IEICE Trans. Fundamentals*, E101.A:1537–1540, 2018.
- [14] Chuzo Iwamoto, Masato Haruishi, and Tatsuaki Ibusuki. Herugolf and Makaro are np-complete. In Hiro Ito, Stefano Leonardi, Linda Pagli, and Giuseppe Prencipe, editors, *Fun with Algorithms 2018*, volume 100 of *LIPICs*, pages 24:1–24:11. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.
- [15] Chuzo Iwamoto and Tatsuya Ide. Five cells and tilepaint are np-complete. *IEICE Trans. Inf. Syst.*, 105-D(3):508–516, 2022.
- [16] Alexander Koch and Stefan Walzer. Foundations for actively secure card-based cryptography. In Martin Farach-Colton, Giuseppe Prencipe, and Ryuhei Uehara, editors, *Fun with Algorithms*, volume 157 of *LIPICs*, pages 17:1–17:23, Dagstuhl, 2021. Schloss Dagstuhl.
- [17] Pascal Lafourcade, Daiki Miyahara, Takaaki Mizuki, Leo Robert, Tatsuya Sasaki, and Hideaki Sone. How to construct physical zero-knowledge proofs for puzzles with a “single loop” condition. *Theor. Comput. Sci.*, 888:41–55, 2021.
- [18] Daiki Miyahara, Léo Robert, Pascal Lafourcade, So Takeshige, Takaaki Mizuki, Kazumasa

- Shinagawa, Atsuki Nagao, and Hideaki Sone. Card-based ZKP protocols for Takuzu and Juosan. In Martin Farach-Colton, Giuseppe Prencipe, and Ryuhei Uehara, editors, *Fun with Algorithms*, volume 157 of *LIPICs*, pages 20:1–20:21, Dagstuhl, 2021. Schloss Dagstuhl.
- [19] Daiki Miyahara, Tatsuya Sasaki, Takaaki Mizuki, and Hideaki Sone. Card-based physical zero-knowledge proof for Kakuro. *IEICE Trans. Fundamentals*, 102-A(9):1072–1078, 2019.
- [20] Takaaki Mizuki. Efficient and secure multiparty computations using a standard deck of playing cards. In Sara Foresti and Giuseppe Persiano, editors, *Cryptology and Network Security*, volume 10052 of *LNCS*, pages 484–499, 2016.
- [21] Takaaki Mizuki, Yoshinori Kugimoto, and Hideaki Sone. Secure multiparty computations using a dial lock. In Jin-yi Cai, S. Barry Cooper, and Hong Zhu, editors, *Theory and Applications of Models of Computation*, volume 4484 of *LNCS*, pages 499–510. Springer, 2007.
- [22] Takaaki Mizuki, Yoshinori Kugimoto, and Hideaki Sone. Secure multiparty computations using the 15 puzzle. In Andreas W. M. Dress, Yinfeng Xu, and Binhai Zhu, editors, *Combinatorial Optimization and Applications*, volume 4616 of *LNCS*, pages 255–266. Springer, 2007.
- [23] Takaaki Mizuki and Hiroki Shizuya. Practical card-based cryptography. In *Fun with Algorithms*, volume 8496 of *LNCS*, pages 313–324, Cham, 2014. Springer.
- [24] Takaaki Mizuki and Hideaki Sone. Six-card secure AND and four-card secure XOR. In Xiaotie Deng, John E. Hopcroft, and Jinyun Xue, editors, *FAW 2009*, volume 5598 of *LNCS*, pages 358–369, Berlin, Heidelberg, 2009. Springer.
- [25] Tal Moran and Moni Naor. Polling with physical envelopes: A rigorous analysis of a human-centric protocol. In Serge Vaudenay, editor, *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 88–108. Springer, 2006.
- [26] Tal Moran and Moni Naor. Split-ballot voting: everlasting privacy with distributed trust. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM Conference on Computer and Communications Security*, pages 246–255. ACM, 2007.
- [27] Tal Moran and Moni Naor. Basing cryptographic protocols on tamper-evident seals. *Theor. Comput. Sci.*, 411(10):1283 – 1310, 2010.
- [28] Akihiro Nishimura, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. Pile-shifting scramble for card-based protocols. *IEICE Trans. Fundamentals*, 101-A(9):1494–1502, 2018.
- [29] Léo Robert, Daiki Miyahara, Pascal Lafourcade, Luc Libralesso, and Takaaki Mizuki. Physical zero-knowledge proof and NP-completeness proof of Suguru puzzle. *Inf. Comput.*, 285(Part):104858, 2022.
- [30] Léo Robert, Daiki Miyahara, Pascal Lafourcade, and Takaaki Mizuki. Physical zero-knowledge proof for suguru puzzle. In Stéphane Devismes and Neeraj Mittal, editors, *Stabilization, Safety, and Security of Distributed Systems - 22nd International Symposium, SSS 2020, Austin, TX, USA, November 18-21, 2020, Proceedings*, volume 12514 of *Lecture Notes in Computer Science*, pages 235–247. Springer, 2020.
- [31] Léo Robert, Daiki Miyahara, Pascal Lafourcade, and Takaaki Mizuki. Card-based ZKP for connectivity: Applications to Nurikabe, Hitori, and Heyawake. *New Gener. Comput.*, 40:149–171, 2022.
- [32] Léo Robert, Daiki Miyahara, Pascal Lafourcade, and Takaaki Mizuki. Card-based ZKP protocol for Nurimisaki. In Stéphane Devismes, Franck Petit, Karine Altisen, Giuseppe Antonio Di Luna, and Antonio Fernández Anta, editors, *Stabilization, Safety, and Security of Distributed Systems - 24th International Symposium, SSS 2022, Clermont-Ferrand, France, November 15-17, 2022, Proceedings*, volume 13751 of *Lecture Notes in Computer Science*, pages 285–298. Springer, 2022.
- [33] Léo Robert, Daiki Miyahara, Pascal Lafourcade, and Takaaki Mizuki. Hide a liar: Card-based ZKP protocol for usowan. In Ding-Zhu Du, Donglei Du, Chenchen Wu, and Dachuan Xu, editors, *Theory and Applications of Models of Computation - 17th Annual Conference, TAMC 2022, Tianjin, China, September 16-18, 2022, Proceedings*, volume 13571 of *Lecture Notes in Computer Science*, pages 201–217. Springer, 2022.
- [34] Suthee Ruangwises. An improved physical ZKP for Nonogram. In *COCOA*, volume 13135 of *LNCS*, pages 262–272, Cham, 2021.

- [35] Suthee Ruangwises. Two standard decks of playing cards are sufficient for a ZKP for Sudoku. *New Gener. Comput.*, 40:49–65, 2022.
- [36] Suthee Ruangwises and Toshiya Itoh. Physical zero-knowledge proof for Numberlink puzzle and k vertex-disjoint paths problem. *New Gener. Comput.*, 39(1):3–17, 2021.
- [37] Suthee Ruangwises and Toshiya Itoh. Physical zero-knowledge proof for Ripple Effect. *Theor. Comput. Sci.*, 895:115–123, 2021.
- [38] Suthee Ruangwises and Toshiya Itoh. Physical ZKP for connected spanning subgraph: Applications to Bridges puzzle and other problems. In Irina Kostitsyna and Pekka Orponen, editors, *UCNC 2021*, volume 12984 of *LNCS*, pages 149–163, Cham, 2021. Springer.
- [39] Suthee Ruangwises and Toshiya Itoh. Securely computing the n-variable equality function with 2n cards. *Theor. Comput. Sci.*, 887:99–110, 2021.
- [40] Suthee Ruangwises and Toshiya Itoh. How to physically verify a rectangle in a grid: A physical ZKP for shikaku. In Pierre Fraigniaud and Yushi Uno, editors, *11th International Conference on Fun with Algorithms, FUN 2022, May 30 to June 3, 2022, Island of Favignana, Sicily, Italy*, volume 226 of *LIPICs*, pages 24:1–24:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- [41] Suthee Ruangwises and Toshiya Itoh. Physical ZKP for Makaro using a standard deck of cards. In Ding-Zhu Du, Donglei Du, Chenchen Wu, and Dachuan Xu, editors, *Theory and Applications of Models of Computation*, volume 13571 of *LNCS*, pages 43–54, Cham, 2022. Springer.
- [42] Tatsuya Sasaki, Daiki Miyahara, Takaaki Mizuki, and Hideaki Sone. Efficient card-based zero-knowledge proof for Sudoku. *Theor. Comput. Sci.*, 839:135–142, 2020.
- [43] Kazumasa Shinagawa, Takaaki Mizuki, Jacob C. N. Schuldt, Koji Nuida, Naoki Kanayama, Takashi Nishide, Goichiro Hanaoka, and Eiji Okamoto. Secure computation protocols using polarizing cards. *IEICE Trans. Fundamentals*, E99.A(6):1122–1131, 2016.
- [44] Kazumasa Shinagawa, Takaaki Mizuki, Jacob C. N. Schuldt, Koji Nuida, Naoki Kanayama, Takashi Nishide, Goichiro Hanaoka, and Eiji Okamoto. Card-based protocols using regular polygon cards. *IEICE Trans. Fundamentals*, 100-A(9):1900–1909, 2017.



Daiki Miyahara Daiki Miyahara received the B.E., M.E., and Ph.D. degrees from Tohoku University in 2017, 2019, and 2021, respectively. He has been an assistant professor at The University of Electro-Communications since 2021. His research interests include theoretical computer science, information security,

and cryptography.



Léo Robert Léo Robert is an assistant professor at University Picardy Jules Verne at Amiens (France), in the MIS lab. He obtained his thesis in 2022 at University Clermont Auvergne (Clermont-Ferrand, France) and did a one year post-doc at XLIM, University of Limoges (France). His research interests mainly focus on

provable security (protocols, primitives), and also theoretical computer science.



Pascal Lafourcade Pascal Lafourcade is a full professor in computer security at the University of Clermont Auvergne and a member of LIMOS (Laboratory of Computer Science, Modelling and Systems Optimisation). He obtained his PhD in 2006 at ENS Cachan on the formal verification of cryptographic protocols.

Then he did a post-doc at ETH Zurich before getting an assistant professor position at Verimag in Grenoble in 2007. He then held the industrial chair in digital trust at the University of Auvergne from 2013 to 2016. In 2016 he was recruited to the Computer Science department of the IUT of the University of Clermont Auvergne. 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.



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