Hide a Liar:
Card-Based ZKP Protocol for Usowan

Léo Robert[1][0000−0002−9638−3143], Daiki Miyahara[2,4][0000−0002−5818−8937], Pascal Lafourcade[1][0000−0002−4459−511X], and Takaaki Mizuki[3,4][0000−0002−8698−1043]

[1] University Clermont Auvergne, LIMOS, CNRS UMR 6158, Aubière France
{leo.robert,pascal.lafourcade}@uca.fr
[2] The University of Electro-Communications, Tokyo, Japan
miyahara@uec.ac.jp
[3] Cyberscience Center, Tohoku University, Sendai, Japan
mizuki+lncs@tohoku.ac.jp
[4] National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

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. This is partly due to the fact that the solution should not be leaked during the protocol. In this work, we propose a card-based 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.

Keywords: Zero-knowledge Proof, Pencil Puzzle, Card-based Cryptography, Usowan

1 Introduction

Usowan [1] 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:

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.[1]

We depict in Fig. 1 an initial Usowan grid with its solution.

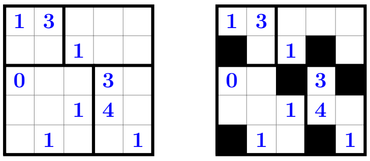


Fig. 1. Initial Usowan grid and its solution taken from [1].

Suppose that someone has found a solution for a given Usowan instance. Is it possible to design a protocol to convince anyone that he/she has the solution without revealing it? The answer 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.

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 (except the information recoverable without the help of the protocol). The 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.

Usually, ZKP protocols are executed by computers. We restrict ourselves by using only physical cards and envelopes. In this paper, we present a physical ZKP protocol for Usowan. 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

[1] A 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.

a NP-complete problem (otherwise the verifier could compute the secret in polynomial time). Hopefully, the NP-completeness of Usowan has been proved in [13]. This result ensures that there exists a ZKP protocol.

Contributions. We construct 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. For this, we use several sub-protocols to verify the rules and propose a completely novel ZKP protocol.

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 [21]; or in [8] where a physical secure auction protocol was proposed. Other implementations have been studied using cards in [4, 15], polarising plates [37], polygon cards [38], a standard deck of playing cards [18], using a PEZ dispenser [2, 3], using a dial lock [19], using a 15 puzzle [20], or using a tamper-evident seals [23–25]. ZKP's for several other puzzles have been studied such as Sudoku [30,36], Akari [5], Takuzu [5, 16], Kakuro [5, 17], KenKen [5], Makaro [6, 35], Norinori [9], Nonogram [7,29], Slitherlink [15], Suguru [27], Nurikabe [28], Ripple Effect [32], Numberlink [31], Bridges [33], and Cryptarithmetic [12].

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

Overview of our protocol. Before detailing our protocol and exisiting sub-protocols involved, we present an intuition of our construction (see Fig. 2). We represent a colored cell by placing colored cards on the cell. In the connectivity phase, P and V construct a connected figure according to P's solution without V knowing the exact shape. Thus, V is convinced that the resulting face-down cards satisfy the rule 2 (and the rule 1 can be easily verified by just revealing face-down cards on numbered cells).

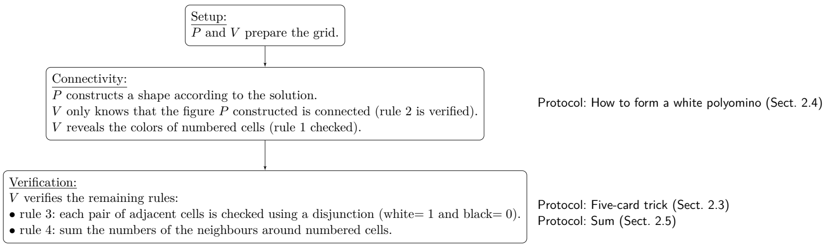

Setup:
P and V prepare the grid.
Connectivity:
P constructs a shape according to the solution.
V only knows that the figure P constructed is connected (rule 2 is verified).
V reveals the colors of numbered cells (rule 1 checked).
Protocol: How to form a white polyomino (Sect. 2.4)
Verification:
V verifies the remaining rules:
• rule 3: each pair of adjacent cells is checked using a disjunction (white= 1 and black= 0).
• rule 4: sum the numbers of the neighbours around numbered cells.
Protocol: Five-card trick (Sect. 2.3)
Protocol: Sum (Sect. 2.5)

Fig. 2. Overview of our protocol

Then, in the verification phase, V checks the two remaining rules. The rule 3 forces two adjacent cells to be composed of at least one white cell; this rule is verified by computing a disjunction of each possible pair. For verifying the rule 4, P and V compute the number of blacks around a given numbered cell. The result is represented as a sequence of face-down cards where the value is given by a position in the sequence. By revealing the card of position equal to the number written on the cell, V checks if the sum is equal or not (without knowing the exact value if different) to the numbered cell.

2 Preliminaries

We explain the notations and sub-protocols used in our construction; some of them are detailed in appendix while the general idea is given below. We first introduce the general framework of card-based protocols.

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

$$\boxed{\clubsuit}\boxed{\heartsuit} \rightarrow \text{ black}, \qquad \boxed{\heartsuit}\boxed{\clubsuit} \rightarrow \text{ white}, \qquad \boxed{\heartsuit}\boxed{\heartsuit} \rightarrow \text{ red}. \tag{1}$$

We call a pair of face-down cards $\boxed{?}\boxed{?}$ 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:

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

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

We also define two other encodings [34]:

- $\clubsuit$-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 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 [26,38]

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

$$\left\langle \underbrace{\boxed{?}}_{\mathbf{p}_1} \middle\| \underbrace{\boxed{?}}_{\mathbf{p}_2} \middle\| \cdots \middle\| \underbrace{\boxed{?}}_{\mathbf{p}_m} \right\rangle \rightarrow \underbrace{\boxed{?}}_{\mathbf{p}_{s+1}} \underbrace{\boxed{?}}_{\mathbf{p}_{s+2}} \cdots \underbrace{\boxed{?}}_{\mathbf{p}_{s+m}},$$

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.

2.2 Mizuki–Sone Copy Protocol [22]

The following protocol is used for copying commitments. We need it as a commitment can be used for several destructive[2] computations (here an addition). Given a commitment to $a \in \{0,1\}$ along with four cards $\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}$, the Mizuki–Sone copy protocol [22] outputs two commitments to a. We specifically describe the protocol in Appendix A.

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

2.3 Input-preserving five-card trick [16]

This sub-protocol is used during the verification phase (see Sect. 3.3) for the lonely black rule (rule 3). Given two commitments to $a, b \in \{0,1\}$

[2] This means that commitments used in the computation cannot be placed back with its initial value. A non-destructive protocol is called input-preserving (see Sect. 2.3).

based on the encoding rule (2), this sub-protocol [4,16] reveals only the value of $a \vee b$ as well as restores commitments to a and b:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\ \underbrace{\boxed{?}\boxed{?}}_{b} \rightarrow\ a \vee b\ \&\ \underbrace{\boxed{?}\boxed{?}}_{a}\ \underbrace{\boxed{?}\boxed{?}}_{b}\ .$$

The original sub-protocol [4,16] was designed for AND ($a \wedge b$), but we adjust it to compute OR ($a \vee b$). We give the detailed description in Appendix B.

2.4 How to Form a White Polyomino [28]

We introduce the idea of the generic method of [28] to perform the connectivity of colored cells without revealing any information about the resulting cells. We leave in Appendix C the details of the protocol.

First, all commitments on a grid of size $p \times q$ are black, and P chooses a commitment to turn it white (without V knowing which cell); we use the chosen-pile described in Appendix C.1 for this. Then P chooses a commitment next to the previous commitment to either turn it white or leave it black; V is ensured that both commitments are neighbours (*i.e.*, two adjacent cells) using a sub-protocol described in Appendix C.2. This step is repeated $pq - 1$ times to ensure that V does not know the number of white cells at the end of the protocol. Finally, each time a white commitment is created, V only knows that it is adjacent to another white commitment; thus V is convinced that the figure composed of white commitments is connected without knowing the number of cells.

2.5 Sum in $\mathbb{Z}$ [34]

We give an overview of the protocol described in [34] 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 $i \in \{1, \ldots, n\}$ along with one $\boxed{\clubsuit}$ and one $\boxed{\heartsuit}$, 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 $x_1 + x_2$:

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

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

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

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 $\mathbb{Z}/3\mathbb{Z}$: $\underbrace{\boxed{?}\boxed{?}}_{x_2}\,\underset{\heartsuit}{\boxed{?}} \rightarrow \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_2}$.
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:

$$\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_1}\,\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_2} \rightarrow \begin{array}{c}\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_1}\\[4pt] \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{2-x_2}\end{array}$$

   (b) Apply a pile shifting shuffle as follows:

$$\left\langle \begin{array}{c}\boxed{?}\\\boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\\boxed{?}\end{array} \middle\| \begin{array}{c}\boxed{?}\\\boxed{?}\end{array} \right\rangle \rightarrow \begin{array}{c}\underbrace{\boxed{?}\boxed{?}\boxed{?}}_{x_1+r}\\[4pt] \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{2-x_2+r}\end{array}$$

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

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[3]. Recall that the result is encoded as the ♡-scheme (see Sect. 2); thus, the result of the sum has a ♡ in its corresponding position (and all other cards are ♣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).
   Example:
   $$\begin{array}{ccc} & \boxed{b} & \\ \boxed{a} & \boxed{3} & \boxed{c} \\ & \boxed{d} & \end{array} \longrightarrow a+b+c+d = \begin{array}{ccccc} \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} \\ 0 & 1 & 2 & 3 & 4 \\ & & & \uparrow & \end{array}$$
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.

4 Conclusion

We propose a physical ZKP protocol for Usowan, which has an interesting rule: some information on the initial grid are incorrect. For verifying such constraints without revealing knowledge about the solution, we construct a protocol based on computing the sum [34]. With this trick we are able to prove that we can hide exactly one liar in each room. The next step will be to see how we can propose a cryptographic ZKP protocol to prove that someone is lying. This is clearly not easy and might require complex and modern cryptographic primitives while we are able to do it only with cards and envelopes in real life.

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

Acknowledgements. We thank the anonymous referees, whose comments have helped us to improve the presentation of the paper. This work was supported in part by JSPS KAKENHI Grant Numbers JP21K11881 and JP18H05289. This study was partially supported by the French ANR project ANR-18-CE39-0019 (MobiS5). Other programs also fund to write this paper, namely the French government research program "Investissements d´Avenir" through the IDEX-ISITE initiative 16-IDEX-0001 (CAP 20-25) and the IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01). Finally, the French ANR project DECRYPT (ANR-18-CE39-0007) and SEVERITAS (ANR-20-CE39-0009) also subsidize this work.

References

1. `https://www.nikoli.co.jp/en/puzzles/usowan.html`, Nikoli, Usowan.
2. Abe, Y., Iwamoto, M., Ohta, K.: Efficient private PEZ protocols for symmetric functions. In: Hofheinz, D., Rosen, A. (eds.) Theory of Cryptography. LNCS, vol. 11891, pp. 372–392. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36030-6_15
3. Balogh, J., Csirik, J.A., Ishai, Y., Kushilevitz, E.: Private computation using a PEZ dispenser. Theor. Comput. Sci. **306**(1-3), 69–84 (2003). https://doi.org/10.1016/S0304-3975(03)00210-X
4. den Boer, B.: More efficient match-making and satisfiability: The five card trick. In: Quisquater, J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 208–217. Springer, Berlin, Heidelberg (1989). https://doi.org/10.1007/3-540-46885-4_23
5. Bultel, X., Dreier, J., Dumas, J., Lafourcade, P.: Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen. In: Demaine, E.D., Grandoni, F. (eds.) Fun with Algorithms. LIPIcs, vol. 49, pp. 8:1–8:20 (2016). https://doi.org/10.4230/LIPIcs.FUN.2016.8
6. Bultel, X., Dreier, J., Dumas, J., Lafourcade, P., Miyahara, D., Mizuki, T., Nagao, A., Sasaki, T., Shinagawa, K., Sone, H.: Physical zero-knowledge proof for Makaro. In: Izumi, T., Kuznetsov, P. (eds.) SSS 2018. LNCS, vol. 11201, pp. 111–125. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03232-6_8
7. Chien, Y.F., Hon, W.K.: Cryptographic and physical zero-knowledge proof: From Sudoku to Nonogram. In: Boldi, P., Gargano, L. (eds.) Fun with Algorithms. LNCS, vol. 6099, pp. 102–112. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13122-6_12
8. Dreier, J., Jonker, H., Lafourcade, P.: Secure auctions without cryptography. In: Ferro, A., Luccio, F., Widmayer, P. (eds.) Fun with Algorithms. LNCS, vol. 8496, pp. 158–170. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07890-8_-14
9. Dumas, J.G., Lafourcade, P., Miyahara, D., Mizuki, T., Sasaki, T., Sone, H.: Interactive physical zero-knowledge proof for Norinori. In: Du, D.Z., Duan, Z., Tian, C. (eds.) Computing and Combinatorics. LNCS, vol. 11653, pp. 166–177. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26176-4_14
10. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: STOC 1985. pp. 291–304. ACM, New York (1985). https://doi.org/10.1145/22145.22178

11. Gradwohl, R., Naor, M., Pinkas, B., Rothblum, G.N.: Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. Theory Comput. Syst. **44**(2), 245–268 (2009). https://doi.org/10.1007/s00224-008-9119-9
12. Isuzugawa, R., Miyahara, D., Mizuki, T.: Zero-knowledge proof protocol for Cryptarithmetic using dihedral cards. In: Kostitsyna, I., Orponen, P. (eds.) UCNC 2021. LNCS, vol. 12984, pp. 51–67. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87993-8_4
13. Iwamoto, C., Haruishi, M.: Computational complexity of Usowan puzzles. IEICE Trans. Fundamentals **E101.A**, 1537–1540 (2018). https://doi.org/10.1587/transfun.E101.A.1537
14. Koch, A., Walzer, S.: Foundations for actively secure card-based cryptography. In: Farach-Colton, M., Prencipe, G., Uehara, R. (eds.) Fun with Algorithms. LIPIcs, vol. 157, pp. 17:1–17:23. Schloss Dagstuhl, Dagstuhl (2021). https://doi.org/10.4230/LIPIcs.FUN.2021.17
15. Lafourcade, P., Miyahara, D., Mizuki, T., Robert, L., Sasaki, T., Sone, H.: How to construct physical zero-knowledge proofs for puzzles with a "single loop" condition. Theor. Comput. Sci. **888**, 41–55 (2021). https://doi.org/10.1016/j.tcs.2021.07.019
16. Miyahara, D., Robert, L., Lafourcade, P., Takeshige, S., Mizuki, T., Shinagawa, K., Nagao, A., Sone, H.: Card-based ZKP protocols for Takuzu and Juosan. In: Farach-Colton, M., Prencipe, G., Uehara, R. (eds.) Fun with Algorithms. LIPIcs, vol. 157, pp. 20:1–20:21. Schloss Dagstuhl, Dagstuhl (2021). https://doi.org/10.4230/LIPIcs.FUN.2021.20
17. Miyahara, D., Sasaki, T., Mizuki, T., Sone, H.: Card-based physical zero-knowledge proof for Kakuro. IEICE Trans. Fundamentals **102-A**(9), 1072–1078 (2019). https://doi.org/10.1587/transfun.E102.A.1072
18. Mizuki, T.: Efficient and secure multiparty computations using a standard deck of playing cards. In: Foresti, S., Persiano, G. (eds.) Cryptology and Network Security. LNCS, vol. 10052, pp. 484–499 (2016). https://doi.org/10.1007/978-3-319-48965-0_29
19. Mizuki, T., Kugimoto, Y., Sone, H.: Secure multiparty computations using a dial lock. In: Cai, J., Cooper, S.B., Zhu, H. (eds.) Theory and Applications of Models of Computation. LNCS, vol. 4484, pp. 499–510. Springer (2007). https://doi.org/10.1007/978-3-540-72504-6_45
20. Mizuki, T., Kugimoto, Y., Sone, H.: Secure multiparty computations using the 15 puzzle. In: Dress, A.W.M., Xu, Y., Zhu, B. (eds.) Combinatorial Optimization and Applications. LNCS, vol. 4616, pp. 255–266. Springer (2007). https://doi.org/10.1007/978-3-540-73556-4_28
21. Mizuki, T., Shizuya, H.: Practical card-based cryptography. In: Fun with Algorithms. LNCS, vol. 8496, pp. 313–324. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07890-8_27
22. Mizuki, T., Sone, H.: Six-card secure AND and four-card secure XOR. In: Deng, X., Hopcroft, J.E., Xue, J. (eds.) FAW 2009. LNCS, vol. 5598, pp. 358–369. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02270-8_36
23. Moran, T., Naor, M.: Polling with physical envelopes: A rigorous analysis of a human-centric protocol. In: Vaudenay, S. (ed.) Advances in Cryptology—EUROCRYPT 2006. LNCS, vol. 4004, pp. 88–108. Springer (2006). https://doi.org/10.1007/11761679_7
24. Moran, T., Naor, M.: Split-ballot voting: everlasting privacy with distributed trust. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) ACM Conference on Computer and Communications Security. pp. 246–255. ACM (2007). https://doi.org/10.1145/1315245.1315277

25. Moran, T., Naor, M.: Basing cryptographic protocols on tamper-evident seals. Theor. Comput. Sci. **411**(10), 1283 – 1310 (2010), https://doi.org/10.1016/j.tcs.2009.10.023

26. Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: Pile-shifting scramble for card-based protocols. IEICE Trans. Fundamentals **101-A**(9), 1494–1502 (2018). https://doi.org/10.1587/transfun.E101.A.1494

27. Robert, L., Miyahara, D., Lafourcade, P., Libralesso, L., Mizuki, T.: Physical zero-knowledge proof and np-completeness proof of suguru puzzle. Inf. Comput. **285**(Part), 104858 (2022). https://doi.org/10.1016/j.ic.2021.104858, https://doi.org/10.1016/j.ic.2021.104858

28. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Card-based ZKP for connectivity: Applications to Nurikabe, Hitori, and Heyawake. New Gener. Comput. **40**, 149–171 (2022). https://doi.org/10.1007/s00354-022-00155-5

29. Ruangwises, S.: An improved physical ZKP for Nonogram. In: COCOA. LNCS, vol. 13135, pp. 262–272. Cham (2021). https://doi.org/10.1007/978-3-030-92681-6_22

30. Ruangwises, S.: Two standard decks of playing cards are sufficient for a ZKP for Sudoku. New Gener. Comput. **40**, 49–65 (2022). https://doi.org/10.1007/s00354-021-00146-y

31. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for Numberlink puzzle and k vertex-disjoint paths problem. New Gener. Comput. **39**(1), 3–17 (2021). https://doi.org/10.1007/s00354-020-00114-y

32. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for Ripple Effect. Theor. Comput. Sci. **895**, 115–123 (2021). https://doi.org/10.1016/j.tcs.2021.09.034

33. Ruangwises, S., Itoh, T.: Physical ZKP for connected spanning subgraph: Applications to Bridges puzzle and other problems. In: Kostitsyna, I., Orponen, P. (eds.) UCNC 2021. LNCS, vol. 12984, pp. 149–163. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87993-8_10

34. Ruangwises, S., Itoh, T.: Securely computing the n-variable equality function with 2n cards. Theor. Comput. Sci. **887**, 99–110 (2021). https://doi.org/10.1016/j.tcs.2021.07.007

35. Ruangwises, S., Itoh, T.: Physical ZKP for Makaro using a standard deck of cards. In: Theory and Applications of Models of Computation. LNCS, Springer, Cham (2022), to appear

36. Sasaki, T., Miyahara, D., Mizuki, T., Sone, H.: Efficient card-based zero-knowledge proof for Sudoku. Theor. Comput. Sci. **839**, 135–142 (2020). https://doi.org/10.1016/j.tcs.2020.05.036

37. Shinagawa, K., Mizuki, T., Schuldt, J.C.N., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Secure computation protocols using polarizing cards. IEICE Trans. Fundamentals **E99.A**(6), 1122–1131 (2016). https://doi.org/10.1587/transfun.E99.A.1122

38. Shinagawa, K., Mizuki, T., Schuldt, J.C.N., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Card-based protocols using regular polygon cards. IEICE Trans. Fundamentals **100-A**(9), 1900–1909 (2017). https://doi.org/10.1587/transfun.E100.A.1900

A Mizuki–Sone Copy Protocol [22]

The protocol proceeds as follows.[4]

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 Eq. (2).

2. Apply a pile-shifting shuffle as follows:

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

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 $\underbrace{\boxed{?}\boxed{?}}_{\overline{a}}\ \underbrace{\boxed{?}\boxed{?}}_{a}$.

B Input-preserving Five-card Trick [16]

The sub-protocol proceeds as follows.

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

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

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

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

[4] This description is a compact version of the original one [22]. Here, we use a pile-shifting shuffle in step 2 instead of using a random bisection cut invented in [22].

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

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

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 it is $\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}$ (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 $\boxed{\heartsuit}$.
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.

C How to Form a White Polyomino

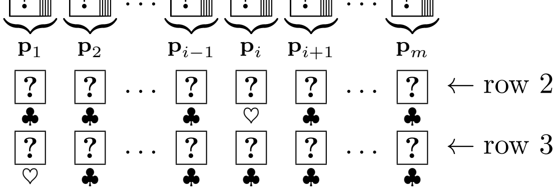
Before explaining the protocol, we need to describe two crucial subprotocols first, namely the chosen pile protocol and the 4-neighbour protocol.

C.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 [14]. Given m piles $(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m)$ 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 $m-1$ $\boxed{\clubsuit}$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 $\boxed{\heartsuit}$. We further put m cards (denoted by *row 3*) below the cards such that only the first card is $\boxed{\heartsuit}$:



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 $\boxed{\heartsuit}$ appears, and the pile above the revealed $\boxed{\heartsuit}$ 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 $\boxed{\heartsuit}$ revealed in Step 3.) Then, apply a pile-shifting shuffle.
5. Reveal all the cards in *row 3*. Then, one $\boxed{\heartsuit}$ appears, and the pile above the revealed $\boxed{\heartsuit}$ 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.

C.2 Sub-protocol: 4-Neighbour Protocol [28]

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 $(p+1) \times (q+1)$.[5]

This 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:
$$\boxed{?}\boxed{?}\ \boxed{?}\boxed{?}\ \boxed{?}\boxed{?}\ \boxed{?}\boxed{?}\ \cdots\ \boxed{?}\boxed{?}.$$
2. P uses the chosen pile protocol (Sect. 2) to reveal the target commitment.

[5] Here, 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.

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.

C.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 commitment to black (*i.e.*, $\boxed{\clubsuit}\boxed{\heartsuit}$) on every cell and commitments to red as mentioned in Sect. 2.4 so that they have $(p + 1)(q + 1)$ commitments on the board.
2. P uses the chosen pile protocol to choose one black commitment that P wants to change.
   (a) V swaps the two cards constituting the chosen commitment so that it becomes a white commitment (recall the encoding (1)).
   (b) P and V end the chosen pile protocol to return the commitments to their original positions.
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:

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

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

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

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.

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).

D 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 choose a black commitment at step 2 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 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 ♣ otherwise the numbered card is not a liar. Thus when revealing the cards at the last step, there is always a ♣ 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 ♡♣♡♣♡. 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.