Physical Zero-Knowledge Proof for Suguru Puzzle

Léo Robert, Daiki Miyahara, Pascal Lafourcade, Takaaki Mizuki

Physical Zero-Knowledge Proof for Suguru Puzzle

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

[1] University Clermont Auvergne, LIMOS, CNRS UMR 6158, Aubière France
{leo.robert,pascal.lafourcade}@uca.fr
[2] Graduate School of Information Sciences, Tohoku University, Sendai, Japan
daiki.miyahara.q4@dc.tohoku.ac.jp
[3] National Institute of Advanced Industrial Science and Technology, Tokyo, Japan
[4] Cyberscience Center, Tohoku University, Sendai, Japan
mizuki+lncs@tohoku.ac.jp

Abstract. Suguru is a paper and pencil puzzle invented by Naoki Inaba. The goal of the game is to fulfil a grid with numbers between 1 and 5 and to respect three simple constraints. In this paper we design a physical Zero-Knowledge Proof (ZKP) protocol for Suguru. A ZKP protocol allows a prover (P) to prove that he knows a solution of a Suguru grid to a verifier (V) without leaking any information on the solution. For constructing such a physical ZKP protocol, we only rely on a small number of physical cards and an adapted encoding. For a grid of Suguru with n cells, we only use $5n + 5$ cards. Moreover, we prove the three classical security properties of a ZKP: completeness, extractability, and zero-knowledge.

Keywords: Physical zero-knowledge proof, Suguru, Security, Completeness, Extractability, and Zero-knowledge.

1 Introduction

Zero-Knowledge Proofs (ZKP) were introduced in 1985 by Goldwasser et al. [8]. Two parties are involved in such a ZKP protocol: a prover P and a verifier V. At the end of the protocol, the verifier V is convinced that P knows the solution s to the instance $\mathcal{I}$ of a problem $\mathcal{P}$, without revealing any information about s. A zero-knowledge proof prevents the verifier from gaining any knowledge about the solution other than its correctness. In fact, when both randomization and interaction are allowed, the proofs that can be verified in polynomial time are exactly those proofs that can be generated within polynomial space [19].

Formally, for a solution s to any instance $\mathcal{I}$ of a problem P, a convincing interactive zero-knowledge protocol between P and V must then satisfy the three following properties[⋆]:
Completeness: If P knows s, then he is able to convince V.
Extractability[‡]**:** If P does not know s, then he is not able to convince V except with some *small* probability. More precisely, we want a negligible probability, *i.e.*, the probability should be a function f of a security parameter λ (for example the number of repetitions of the protocol) such that f is negligible, that is for every polynomial Q, there exists $n_0 > 0$ such that:

$$\forall\ x > n_0, f(x) < \frac{1}{Q(x)}.$$

Zero-knowledge: V learns *nothing* about s except $\mathcal{I}$, *i.e.* there exists a probabilistic polynomial time algorithm $\texttt{Sim}(\mathcal{I})$ (called the simulator) such that outputs of the real protocol and outputs of $\texttt{Sim}(\mathcal{I})$ follow the same probability distribution.

There exist two kinds of ZKP: *interactive* and *non-interactive*. In an interactive ZKP the prover can exchange messages with the verifier in order to convince him, while in the non-interactive case the prover can just create the proof in order to convince the verifier.

ZKPs are usually executed by computers. They are often used in electronic voting to prove that some parties correctly mix some ballots without cheating, or in multi-party computation [3,4,16]. Moreover, there exist generic cryptographic zero-knowledge proofs for all problems in NP [6], via a reduction to an NP-complete problem with a known zero-knowledge proof.

In [15], the authors explained simply this concept to some children using a circular cave. This was the first proposition of a physical ZKP. Later, Gradwohl et al. [9] proposed a ZKP for the famous Nikoli's puzzle called Sudoku[§]. They just used some physical cards to construct a ZKP protocol. It was one of the first interactive physical ZKP protocols for

[⋆] Moreover, if $\mathcal{P}$ is NP-complete, then the ZKP should be run in a polynomial time [7]. Otherwise it might be easier to find a solution than proving that a solution is a correct solution, making the proof pointless.
[‡] This implies the standard soundness property, which ensures that if there exists no solution of the puzzle, then the prover is not able to convince the verifier regardless of the prover's behavior.
[§] https://www.nikoli.co.jp/en/puzzles/sudoku.html

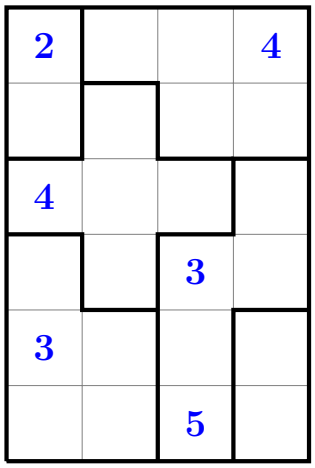


Fig. 1. Initial Suguru grid

such puzzles. Our aim is to design a ZKP protocol for Suguru puzzles in the same spirit as the one done for Sudoku.

Suguru: It was designed by Naoki Inaba, the original name of the game was "*Nanba Burokku*' but it is also known as *Tectonics* or *Number Blocks*. Suguru is a paper and pencil puzzle in which a grid is divided into outlined blocks called *region*. Each region containing up to five cells. Every cell of the grid must contain a number from 1 to 5 (according to the number of cells in the region). Each cell should be filled such that no two identical numbers touch — not even diagonally.

Suguru's rule: This puzzle is formed by a rectangular grid where blocks divide the overall area. Those blocks called *region* contain up to five cells. The goal is to fill all the cells with integers under the following constraints:

- **Number region rule:** A region composed of k cells must be filled with integers $1, \ldots, k$.
- **Neighbour rule:** For every cell, all of its eight neighbours must have different values from the cell's value.

In Figure 1, we give an example of an initial Suguru grid and in Figure 2 we give its unique solution.

Contributions: We propose a simple ZKP protocol for Suguru using a small number of cards. Our construction is simple and can be used as a pedagogical example to explain the role of ZKP protocols. We propose an encoding of the number using simple cards. Using this encoding, the

2	5	2	**4**
1	3	1	3
4	2	5	2
5	1	**3**	1
3	2	4	2
4	1	**5**	1

Fig. 2. Solution of the Suguru grid of Figure 1

prover places some cards on the grid according to its solution. We use these cards to prove that the two rules of Suguru are satisfied. We start with the first number of cell in a region, after having verified the validity of all regions and replacing the cards placed by the prover, we reuse them to prove the second rule of Suguru about the eight neighbours of each cell of the grid. Here is the difficulty of Suguru, since we need to prove that all values of the eight neighbours of each cell are different without revealing any information to the verifier. Here, we use a trick of our encoding of the values of the cards in order not to leak any information. Our encoding requires five cards per cell; therefore if a Suguru grid has n cells to guess, our protocol only requires $5n+5$ cards. Finally, we prove the three security properties of our construction *i.e.,* completeness, extractability, and zero-knowledge.

Related Work: In [18] the authors proposed an improved ZKP protocol for Sudoku that follows the pioneer work of [9]. In [5], the authors proposed a ZKP protocol for the Nikoli's puzzle Norinori.

In [12], a method to take into account one feature of several puzzles that consists to construct a single loop, has been invented. This technique used a topological approach with successive interactive transformations.

Recently several ZKP proofs have been proposed for different Nikoli's puzzles. In [13], card-based ZKP Protocols for Takuzu and Juosan have been proposed. In [17] a physical ZKP proof for Numberlink has been designed. In [14], a card-based physical ZKP for Kakuro have been given

that improves the first version proposed by Bultel et al. in [1] with the ZKP protocols for three other Nikoli's games: Akari, Takuzu, and Kenken.

All these works clearly demonstrate that designing physical ZKP is clearly an interesting topic of research. Each game has its own particular rules and requires an adapted construction.

Although the existence of all those previous works, one cannot reuse or adapt directly them for the Suguru game. Indeed, the main reason is the "strong" neighbour rule where no cell can have its eight neighbours with the same value. Other puzzles have a similar rule but with relaxed restrictions. For instance, Makaro has a neighbour rule but only for adjacent cells (and not in diagonal). Thus a naive adaptation would imply a loss in terms of efficiency and of zero-knowledge (no information about the solution can be leaked). Furthermore, it is worth noting that the encoding for the proof of NP-completness of Makaro cannot be applied for Suguru. Thus ZKP for Suguru cannot be directly adapted from the ZKP of Makaro. Further discussion is given in Section 5.

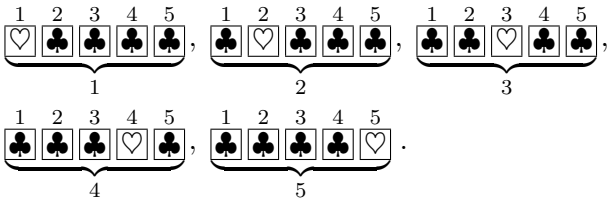
Outline: In Section 2, we present our notations, and all subprotocols needed to construct our ZKP. In Section 3, we design our ZKP protocol for Suguru. In Section 4, we prove the security of our protocol. In Section 5, we discuss a complexity of Suguru and of Makaro. In the last section, we conclude the paper.

2 Preliminaries

We introduce some notations of cards and shuffles used in our construction.

2.1 Notations

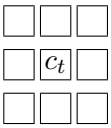
Card: A deck of cards used in our protocol consists of blacks [♣] and reds [♡] whose back sides are identical [?]. Each integer $i \in \{1, \ldots, 5\}$ is encoded as:



We call such face-down five cards $\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}$ corresponding to an integer according to the above encoding rule a *commitment* to the respective integer.

We also use *numbered* cards such as $\boxed{1}\boxed{2}\boxed{3}\boxed{4}\boxed{5}$ whose backs are identical $\boxed{?}$.

Neighbour cell: Consider a target cell denoted c_t on a grid. A cell is a *neighbour* of c_t if it is next to c_t. It can be on the left, the right, the top, or the bottom of c_t, and also on its diagonal:



Thus, a cell can have at most eight neighbours.

Pile-scramble shuffle: A shuffle used in our protocol is a *pile-scramble shuffle*, which was first used by Ishikawa et al. [10] and was used in other physical ZKP protocols for puzzles (e.g., Sudoku [18]). Consider that we have a sequence of ℓ piles of cards, each of which consists of the same number of face-down cards, denoted by $(p_1, p_2, \ldots, p_\ell)$ for some positive integer ℓ. Applying a pile-scramble shuffle to the sequence results in $(p_{r^{-1}(1)}, p_{r^{-1}(2)}, \ldots, p_{r^{-1}(\ell)})$ where permutation r is uniformly and randomly chosen from the symmetric group of degree ℓ. That is, it randomly permutes a sequence of piles and nobody knows the order of the resulting sequence.

One can easily implement a pile-scramble shuffle by using physical tools that can fix each pile of cards such as rubber bands and envelopes; a player (or players) randomly shuffle them until nobody traces the order of the piles.

3 ZKP Protocol for Suguru

We propose a ZKP protocol for Suguru composed of two phases, the setup phase and the verification phase.

3.1 Setup phase

The verifier V and the prover P place commitments corresponding to the integers on the initial grid of a Suguru puzzle. In addition, when a region

of k cells is already filled with $k-1$ cells, then P and V agreed on the last cell to complete and place the commitment accordingly[¶].

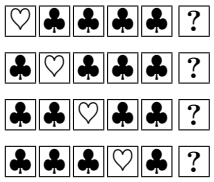
Then, P continues to place commitments on all the remaining cells by himself according to the solution of the puzzle.

3.2 Verification phase

There are two verifications to ensure the number region rule and the neighbour rule.

Number region rule: V wants to check that a region of k cells contains all the consecutive integers from 1 to k.

1. For every i, $1 \leq i \leq k$, V picks all cards of the i-th cell (in any ordering) to form a pile p_i. Then, V attaches a numbered card $\boxed{i}$ to p_i. Thus, there are $p_1, \ldots, p_k$ piles, each of which consists of six cards.
2. Apply the pile-scramble shuffle [10].
3. V reveals the cards of each pile except for the numbered card. The revealed output is of the form (up to a permutation in the rows), i.e., all the k (opened) commitments corresponding to 1 through k should appear. For example, if $k = 4$, the revealed output should be of the following form (up to a permutation in the rows):

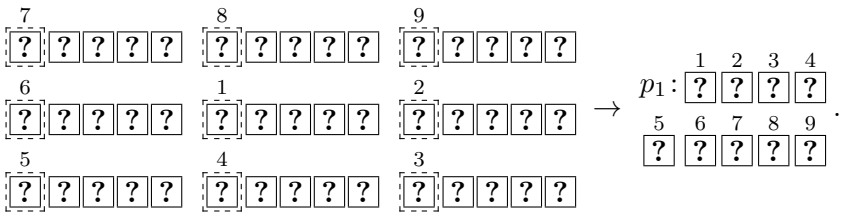


where the face-down cards on the right side are the numbered cards. If the revealed output is not of this form, V aborts.
4. Turn over the face-up cards and apply the pile-scramble shuffle again to the piles.
5. Reveal only the numbered cards of all piles. Because these revealed cards indicate the initial positions for each pile, V rearranges each pile back to their initial place. The revealed numbered cards can be reused for the remaining verifications.

[¶] For example, in Figure 1 the upper left region can be directly completed with a 1.

Neighbour rule: V wants to check if a given cell has no neighbour having the same integer as the cell.

1. V picks the first card of the target commitment and then picks each first card of the commitments on its neighbour cells (in any ordering) to form the pile p_1. The following is an example when there are eight neighbours:



2. V repeats the same operation until the pile p_5 is formed.
3. V attaches a numbered card $\boxed{i}$ to p_i. (If the target cell is the last one, V does not perform this step.)
4. If an integer is written on the target cell, then go to the next step. Otherwise, apply the pile-scramble shuffle to the piles.
5. V reveals the first card of each pile, which corresponds to the target commitment. Let p_t denotes the pile where a red card appears.
6. V reveals all the cards in the pile p_t except for the numbered cards. If there are two red cards in the pile then V aborts; otherwise, V goes to the next step.
7. As Steps 4 and 5 in the previous verification, V rearranges all the cards in the piles back to their initial places. (If the target cell is the last one, V does not perform this step.)

3.3 Evaluation

If the size of the grid is represented by n then the number of cards used in this protocol is equal to $5n+5$. Indeed, each cell must be encoded with 5 cards (4 blacks and 1 red)[‖] and 5 numbered cards are used for all target cell.

[‖] We could have encoded each cell with a total of ℓ cards where ℓ is the number of cells in the region (thus, a region with two cells has its cell encoded with only two cards, a red and a black). Yet, this would lead to inconstancy in the encoding rule which is required in the neighbour verification.

4 Security proofs

We give the theorems along with their proofs to provide that our protocol respects the security properties. A ZKP protocol is secure if the following three properties are satisfied:

Correctness: If the prover P commits its cards according to the actual solution, then all verifications will not abort. Hence, if P knows a solution, then it can always convince the verifier V. Correctness is proven in Theorem 1.

Extractability: If P's input is invalid, the protocol will point errors out to V. Therefore, if P does not know the solution, it cannot convince V. Extractability is proven in Theorem 2.

Zero-knowledge: V learns nothing about P's solution. Zero-knowledge is proven in Theorem 3.

Theorem 1 (Completeness). *If P knows a solution of a Suguru grid, then it can convince V.*

Proof. Suppose that the prover P knows the solution of the Suguru grid. It runs with the verifier V the Setup phase (Section 3.1). We show that P can perform both verification phases without aborting.

Number region verification: In this phase, the goal of P is to show that each region of size k contains consecutive integers from 1 to k (note that the lower bound of k is 1 and its upper bound 5). Since P places the cards accordingly with the solution, each region of size k contains the numbers 1 to k. Without loss of generality, suppose that the pile p_i corresponds to the number i with $i = 1 \dots k$. The pile p_1 is composed of the sequence (in this order):

 .

The pile p_2 is composed of the sequence (in this order):

 .

More generally, the pile p_i is a sequence of black cards where the red card is placed on position i.

Since the pile-scramble shuffle applied on Step 2 does not modify the order of the sequence, the red card of pile p_i is on position i. As $i = 1 \dots k$, all numbers from 1 to k are represented. Thus, V is convinced that the number region rule is verified by revealing the piles in Step 3.

Neighbour verification: The goal of P is to convince V that no cell has the same number of its neighbours (there are eight neighbours as defined in Section 2). Let c_t be the target cell placed on the center of the 3×3 square. Since P placed the commitments according to the solution, there is no cell with the same value of c_t in this square. Let i be the position of the red card of c_t (with $i = 1 \dots 5$). Since no neighbour cell has the same value of c_t, there is no other red card with position i. Since each pile is composed of card with the same indices, the pile p_i (before the shuffle) contains exactly one red card. Hence, V is convinced that the neighbour rule is verified.

Finally, since all verifications are checked, we proved that if P has the solution then the verifications will always succeed. □

Theorem 2 (Extractability). *If P does not provide a solution of the Suguru puzzle, it is not able to convince V.*

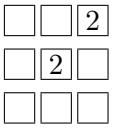
Proof. Suppose that P does not know a solution for the puzzle. We want to show that V will always detect it.

Since P cannot provide the solution, at least one of the two rules is not verified (if both can be verified, this is the solution). We can distinguish two cases corresponding to each verification:

- The number region rule is not respected. That is, suppose w.l.o.g. that a region of size k with $k > 1$ does not contain the number 1. Hence, the sequence corresponding to this number is missing, meaning that V cannot reveal at Step 3 the sequence:

  Thus, V will abort the protocol and detect that P cannot provide the solution.
- The neighbour rule is not respected. Suppose that we have the following configuration:

  with blank cards of unimportant value but different from 2.
  We encode number 2 as:

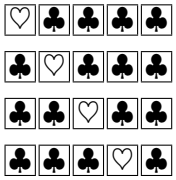
Thus, the pile p_2 corresponding to all the cards with indices 2 (before the shuffle) will contain exactly two red cards. Thus, V will abort the protocol.

We proved that if P does not have the solution then the verifications will abort in both cases meaning that P cannot convince V. □

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 [9]: zero-knowledge is caused by a description of an efficient *simulator* which simulates interaction between a cheating verifier and a real prover. However, the simulator does not have a solution but it can swap cards for different ones during shuffles. The simulator acts as follows:

- During the number region verification at Step 2, the simulator swaps the piles to replace them by the sequences (up to a permutation in the rows):



- During the neighbour verification, when revealing the cards at Step 6, the simulator swaps the pile with a pile containing $k-1$ black cards and 1 red card.

The simulated proofs and the real proofs are indistinguishable; thus, V learns nothing about P's solution. □

5 Discussion

Among related work introduced in Section 1, the closest work is [2], where a ZKP for Makaro has been proposed. This game is close to the Nikoli's game called Makaro[**] where regions are called rooms and some extra black cells indicate thanks to an arrow the position of the biggest number among the (up to) four cells around (over, under, left, right) the cell with the arrow (black) is in the cell the arrow points at.

[**] http://nikoli.co.jp/en/puzzles/makaro.html

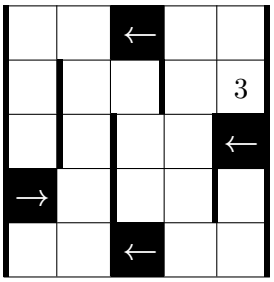


Fig. 3. Example of a Makaro grid

1	2	←	1	4
2	3	1	2	3
1	2	3	4	←
→	3	1	2	3
1	2	←	1	2

Fig. 4. Solution of a Makaro grid of Figure 3

In [11], the authors proved that the two games Herugolf and Makaro are NP-complete. Solving Makaro was shown to be NP-complete via a reduction from 3-SAT.

Makaro is a game close to Suguru, with different constraints as follows:

1. *Room condition*: Each room contains all the numbers from 1 up to the number of cells in the room.
2. *Neighbour condition*: A number cannot be next (adjacent) to the same number in another room.
3. *Arrow condition*: Every black arrow cell must point at the largest number among the numbers in the adjacent cells of the black cell (possibly the fours cells: right, left, above, and bottom).

In Figure 3, we give a simple example of a Makaro game, where all black cells are arrow cells and all white cells are empty cells except for one filled cell with three. It is easy to verify that the three constraints are satisfied in the solution given in Figure 4. We remark that in a solution all white cells are filled with numbers between 1 and k, where k is the maximum size of all the rooms of the grid.

The room condition is the same, however the neighbour condition is different in Suguru. In Suguru for each cell, we consider the eight neigh-

bours while in Makaro we consider only the four neighbours. Moreover in Suguru the size of the regions is limited to five while in Makaro there is no limit on the size of the rooms.

These differences avoid us to reuse the 3-SAT encoding used in the proof of NP-completeness of Makaro [11]. If we remove the limit of the maximum of five cells in each region, we can use the same gadgets as the ones used for Makaro to prove in a similar way that Suguru is also NP-complete. But with the limit of five cells per regions and the change of four neighbours into eight neighbours, it is not clear if we can prove the NP-completeness of Suguru. These extra constraints seem to remove some difficulty in finding some solutions for Suguru. At the moment we are not able to prove NP-completeness and we conjecture that solving a Suguru grid should be in P.

Even if solving Suguru is in P, our physical ZKP is an interesting approach since it requires only $5n + 5$ cards as we show when we present our ZKP protocol in Section 3. (In addition, it could always happen that one cannot solve a Suguru puzzle.) This is clearly a real ZKP protocol that can be used by Suguru players in practice.

6 Conclusion

In this paper we propose a simple card-based physical ZKP for Suguru. Our solution is simple and efficient since it relies on only $5n + 5$ cards. One open question left for the future is to demonstrate the conjecture states in the introduction *i.e.*, proving that solving a Suguru grid is in P. We clearly cannot adapt the proof of NP-completeness of Makaro with the rules of Suguru, it is why we conjecture that Suguru is in P.

Moreover, our long term research direction is to design physical ZKP protocols for all Nikoli's games. However some rules of some games like Shakashaka[††] that requires to draw rectangles which is not easy to model without leaking any information. Another example of challenging game is Shikaku[‡‡], where the rules are simple: 1) Divide the grid into rectangles with the numbers in the cells. 2) Each rectangle is to contain only one number showing the number of cells in the rectangle. However it remains a challenging open question to design a physical ZKP for this game without revealing any information on the positions of the rectangles.

[††] http://www.nikoli.co.jp/en/puzzles/shakashaka.html
[‡‡] https://www.nikoli.co.jp/en/puzzles/shikaku.html

Acknowledgements. We thank the anonymous referees, whose comments have helped us to improve the presentation of the paper. This work was supported in part by JSPS KAKENHI Grant Number JP19J21153.

References

1. 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.) 8th International Conference on Fun with Algorithms, FUN 2016, June 8-10, 2016, La Maddalena, Italy. LIPIcs, vol. 49, pp. 8:1–8:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016). https://doi.org/10.4230/LIPIcs.FUN.2016.8, https://doi.org/10.4230/LIPIcs.FUN.2016.8
2. 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 (2018)
3. Cramer, R., Damgård, I., Nielsen, J.B.: Multiparty computation from threshold homomorphic encryption. In: Pfitzmann, B. (ed.) Advances in Cryptology — EUROCRYPT 2001. pp. 280–300. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
4. Damgård, I., Faust, S., Hazay, C.: Secure two-party computation with low communication. In: Cramer, R. (ed.) Theory of Cryptography. pp. 54–74. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
5. Dumas, J., Lafourcade, P., Miyahara, D., Mizuki, T., Sasaki, T., Sone, H.: Interactive physical zero-knowledge proof for Norinori. In: COCOON 2019. LNCS, vol. 11653, pp. 166–177. Springer (2019)
6. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In: 27th Annual Symposium on Foundations of Computer Science (sfcs 1986). pp. 174–187 (Oct 1986). https://doi.org/10.1109/SFCS.1986.47
7. Goldreich, O., Micali, S., Wigderson, A.: How to prove all NP-Statements in zero-knowledge, and a methodology of cryptographic protocol design. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 171–185. Springer (1987)
8. Goldwasser, S., Micali, S., Rackoff, C.: Knowledge complexity of interactive proof-systems. Conference Proceedings of the Annual ACM Symposium on Theory of Computing pp. 291–304 (1985). https://doi.org/10.1145/3335741.3335750
9. Gradwohl, R., Naor, M., Pinkas, B., Rothblum, G.N.: Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. In: Proceedings of the 4th International Conference on Fun with Algorithms. pp. 166–182. FUN'07, Springer-Verlag, Berlin, Heidelberg (2007)
10. Ishikawa, R., Chida, E., Mizuki, T.: Efficient card-based protocols for generating a hidden random permutation without fixed points. In: Calude, C.S., Dinneen, M.J. (eds.) UCNC 2015. LNCS, vol. 9252, pp. 215–226. Springer (2015)
11. Iwamoto, C., Haruishi, M., Ibusuki, T.: Herugolf and Makaro are NP-complete. In: Ito, H., Leonardi, S., Pagli, L., Prencipe, G. (eds.) Fun with Algorithms 2018. LIPIcs, vol. 100, pp. 24:1–24:11. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2018)
12. Lafourcade, P., Miyahara, D., Mizuki, T., Sasaki, T., Sone, H.: A physical ZKP for Slitherlink: How to perform physical topology-preserving computation. In: Heng,

S.H., Lopez, J. (eds.) Information Security Practice and Experience. pp. 135–151. Springer International Publishing, Cham (2019)
13. 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 2020. LIPIcs (2020)
14. Miyahara, D., Sasaki, T., Mizuki, T., Sone, H.: Card-based physical zero-knowledge proof for Kakuro. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E102.A**(9), 1072–1078 (2019). https://doi.org/10.1587/transfun.E102.A.1072
15. Quisquater, J., Quisquater, M., Quisquater, M., Quisquater, M., Guillou, L.C., Guillou, M.A., Guillou, G., Guillou, A., Guillou, G., Guillou, S., Berson, T.A.: How to explain zero-knowledge protocols to your children. In: Brassard, G. (ed.) Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings. Lecture Notes in Computer Science, vol. 435, pp. 628–631. Springer (1989). https://doi.org/10.1007/0-387-34805-0_60, https://doi.org/10.1007/0-387-34805-0_60
16. Romero-Tris, C., Castellà-Roca, J., Viejo, A.: Multi-party private web search with untrusted partners. In: Rajarajan, M., Piper, F., Wang, H., Kesidis, G. (eds.) Security and Privacy in Communication Networks. pp. 261–280. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
17. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for Numberlink. In: Farach-Colton, M., Prencipe, G., Uehara, R. (eds.) Fun with Algorithms 2020. LIPIcs (2020)
18. Sasaki, T., Miyahara, D., Mizuki, T., Sone, H.: Efficient card-based zero-knowledge proof for Sudoku. Theoretical Computer Science (2020). https://doi.org/https://doi.org/10.1016/j.tcs.2020.05.036, http://www.sciencedirect.com/science/article/pii/S0304397520303200
19. Shamir, A.: IP = PSPACE. J. ACM **39**(4), 869–877 (Oct 1992). https://doi.org/10.1145/146585.146609, http://doi.acm.org/10.1145/146585.146609