IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

VOL. E102-A NO. 9
SEPTEMBER 2019

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

IEICE TRANS. FUNDAMENTALS, VOL.E102–A, NO.9 SEPTEMBER 2019

PAPER *Special Section on Discrete Mathematics and Its Applications*

Card-Based Physical Zero-Knowledge Proof for Kakuro

Daiki MIYAHARA[†,††a)], **Tatsuya SASAKI**[†], **Takaaki MIZUKI**[†††], *Members*, *and* **Hideaki SONE**[†††], *Fellow*

SUMMARY Kakuro is a popular logic puzzle, in which a player fills in all empty squares with digits from 1 to 9 so that the sum of digits in each (horizontal or vertical) line is equal to a given number, called a clue, and digits in each line are all different. In 2016, Bultel, Dreier, Dumas, and Lafourcade proposed a physical zero-knowledge proof protocol for Kakuro using a deck of cards; their proposed protocol enables a prover to convince a verifier that the prover knows the solution of a Kakuro puzzle without revealing any information about the solution. One possible drawback of their protocol would be that the protocol is not perfectly extractable, implying that a prover who does not know the solution can convince a verifier with a small probability; therefore, one has to repeat the protocol to make such an error become negligible. In this paper, to overcome this, we design zero-knowledge proof protocols for Kakuro having perfect extractability property. Our improvement relies on the ideas behind the copy protocols in the field of card-based cryptography. By executing our protocols with a real deck of physical playing cards, humans can practically perform an efficient zero-knowledge proof of knowledge for Kakuro.

key words: *cryptography, card-based protocols, real-life hands-on cryptography, Kakuro, physical zero-knowledge proof*

1. Introduction

Kakuro, also known as *Cross Sums*, is a popular logical puzzle, which is played with numbers. A puzzle instance of Kakuro consists of *empty squares* and numbers called *clues*, as illustrated in Fig. 1(a). Each clue (which is a positive number placed on a triangle) is associated with a *line* consisting of consecutive squares; for example, the clue "6" in Fig. 1(a) is associated with the (horizontal) line consisting of three consecutive squares on the first row, and the clue "11" is associated with the (vertical) line on the second column. The goal of Kakuro is to fill in all empty squares with digits obeying the following rules.

1. Fill in all empty squares with digits from 1 to 9.
2. For each (horizontal or vertical) line, the clue (which is an integer) associated with the line must be equal to the sum of all the digits on the line.
3. The digits on each line must be all different.

Figure 1(b) shows a solution to the puzzle in Fig. 1(a); one can easily confirm that the solution satisfies the above rules.

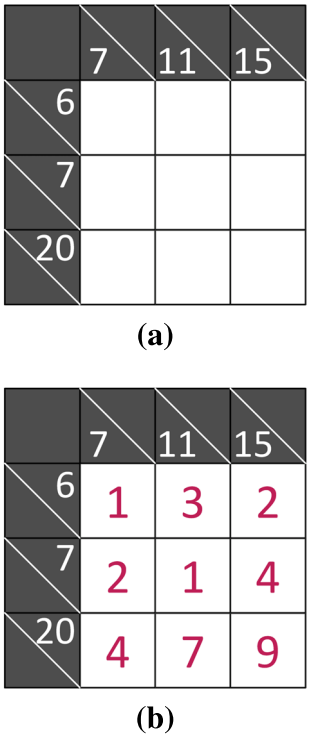


Fig. 1 Example of a Kakuro puzzle and its solution.

The example puzzle shown in Fig. 1 is pretty small; usually, Kakuro puzzles are played in larger grids, as illustrated in Fig. 2. As in this puzzle, multiple lines may exist on the same row or column thanks to *black cells* and/or triangles (on which clues are placed). This paper solicits *zero-knowledge proof* [1] schemes for Kakuro.

Because any NP problem has a zero-knowledge proof [2] and Kakuro is known to be NP-complete [3]–[5], we can construct a zero-knowledge proof protocol for Kakuro (e.g. [6]), which enables a prover P to convince a verifier V that the prover P knows the solution of a Kakuro puzzle without revealing any information about the solution. In contrast, an "unconventional" zero-knowledge proof protocol for Kakuro was proposed by Bultel, Dreier, Dumas, and Lafourcade [7] in 2016. Their protocol, which we call the *BDDL protocol*, uses a two-colored deck of cards, such as red cards $\boxed{\heartsuit}$ and black cards $\boxed{\clubsuit}$ having the same backs $\boxed{?}$. The BDDL protocol falls in the category of *physical zero-knowledge proofs* (e.g. [8]–[10]), which are supposed

Manuscript received September 25, 2018.
Manuscript revised January 25, 2019.

[†]The authors are with Sone–Mizuki Laboratory, Graduate School of Information Sciences, Tohoku University, Sendai-shi, 980-8579 Japan.
[††]The author is with National Institute of Advanced Industrial Science and Technology, Tokyo, 135-0064 Japan.
[†††]The authors are with Cyberscience Center, Tohoku University, Sendai-shi, 980-8578 Japan.
a) E-mail: daiki.miyahara.q4@dc.tohoku.ac.jp

DOI: 10.1587/transfun.E102.A.1072

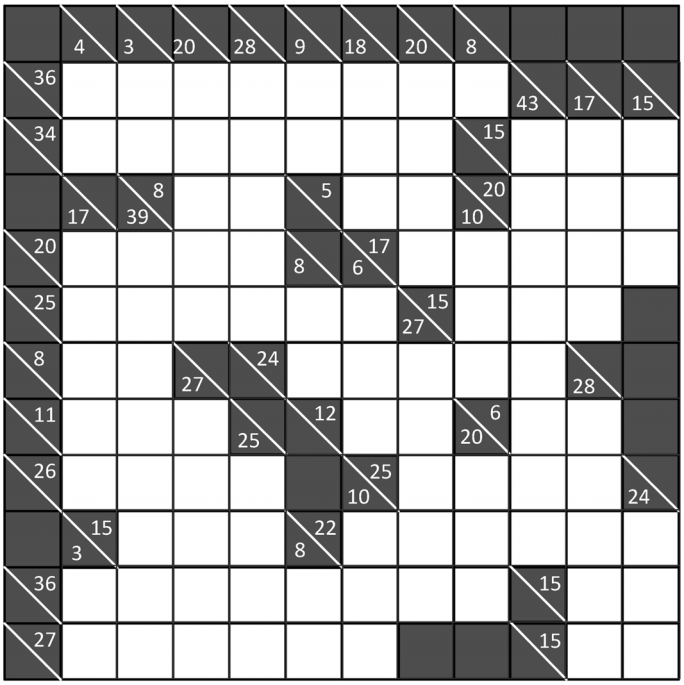


Fig. 2 Typical Kakuro puzzle.

to be performed by human hands along with everyday-life objects.

Although we will present the details of the BDDL protocol in Sect. 2.2, the protocol roughly proceeds as follows. We assume here that the prover P and the verifier V are in the same place.

1. The prover P puts on each empty square four identical piles (of cards), each of which encodes the digit on that square of the solution.
2. The verifier V confirms that the sum of all digits in each line is equal to the given clue and that digits in each line are all different by using shuffle actions to the consecutive piles along with envelopes and helping piles.

Let us look at the first phase 1: Following the solution that the prover P knows, P puts on each empty square four piles (of cards), all of which should be identical. However, P is technically able to put four piles that violate the protocol, that is, P may put four piles that are not necessarily the same. In this case, V can be convinced even if P does not know the solution. Therefore, the BDDL protocol is not *perfectly extractable*; that is, a probability that V becomes convinced despite for P not knowing the solution is non-zero. Thus, V and P have to repeat the BDDL protocol to make the probability of such an error become negligible. Because physical zero-knowledge proof protocols are supposed to be executed by humans, repeating a physical protocol many times would be a burden on humans.

In this work, we propose protocols that achieve perfect extractability; thus, the verifier V will never be convinced whenever the prover P does not know the solution. Our improvement comes from the ideas behind the copy protocols in the field of *card-based cryptography*[†]. Specifically, we first provide a copy protocol which enables V to duplicate a pile of cards put by P without revealing any information about the encoded number, and hence, we can attain perfect extractability because all piles on each square are guaranteed to be identical. Therefore, our improved protocol is more efficient since it needs no repetition. Table 1 shows a comparison of performances of the BDDL protocol and our protocol where we assume that a Kakuro puzzle contains n squares and ℓ clues. Both the protocols use nine envelopes when shuffling piles of cards. Moreover, we propose another protocol that uses cards numbered from 1 to 9 (e.g., $\boxed{1}, \boxed{2}, \ldots, \boxed{9}$).

The remainder of this paper is organized as follows. In Sect. 2, we introduce the existing protocol. In Sect. 3, we present our improved protocol. In Sect. 4, we propose another protocol based on the numbered cards. Section 5 concludes this paper.

2. Preliminaries

In this section, we briefly review zero-knowledge proofs, and then explain the BDDL protocol [7], which is the existing physical zero-knowledge proof scheme for Kakuro.

2.1 Zero-Knowledge Proof

A zero-knowledge proof is an interactive proof between a prover P and a verifier V [1]. They are assumed to be

[†]Card-based cryptography enables us to perform secure multi-party computations by using a deck of cards (e.g., [11]–[15]).

Table 1 Card-based zero-knowledge proof protocols for Kakuro.

	# of Cards	# of Shuffles	Extractability Error
BDDL[7]	$81\ell + 18n$	2ℓ	at most 1/4
Ours (§3)	$81\ell+81$	$3\ell + 1$	0

given an instance y of a problem, and P knows a witness w of the solution while V does not know w. In addition, computational power of V is bounded so that V cannot obtain w from y. Under these assumptions, P wants to convince V that P knows w without revealing any information about w. A *zero-knowledge proof* should satisfy the following three properties.

Completeness If P knows w, V is convinced.
Extractability If P does not know w, V is convinced only with a small probability.
Zero-knowledgeness V does not obtain any information about w.

The probability that V is convinced even though P does not know w is called the *extractability error*. If we have a zero-knowledge proof protocol whose extractability error is $\delta > 0$, repeating the protocol t times allows V to detect that P does not know w with a probability of $1 - \delta^t$, which is overwhelming. Therefore, we can establish a zero-knowledge proof of knowledge practically by the repetition even if a protocol has extractability error, i.e., it is not perfectly extractable.

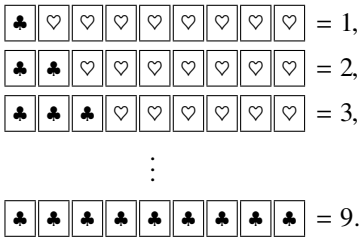
However, as for physical zero-knowledge proofs, a protocol is expected to be executed by human hands, and hence, it is difficult to repeat the protocol many times. Thus, constructing a zero-knowledge proof protocol with no extractability error, i.e., a perfectly extractable protocol, would be much desired.

2.2 The BDDL Protocol

Given a Kakuro puzzle, the BDDL protocol [7] enables the prover P to convince the verifier V that P knows the solution of the puzzle; as mentioned in the previous section, the BDDL protocol uses a two-colored deck of cards

whose backs are identical  .

Before going into the details of the protocol, we mention the encoding scheme to be used. A digit x, $1 \le x \le 9$, is encoded with a "pile" consisting of x black cards and $(9 - x)$ red cards, as follows:

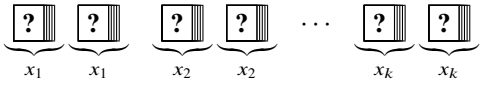


We denote a pile of nine face-down cards encoding a digit x according to the above scheme by

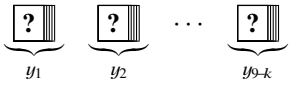
and we call it a *face-down pile* of x.

Given a Kakuro puzzle whose solution is known to the prover P, the BDDL protocol proceeds as follows.

1. The prover P does the following for each line of the puzzle; assume that the line consists of k squares, and let $x_1, x_2, \ldots, x_k$ be the solution digits on the squares in this order. Note that k satisfies $1 \le k \le 9$.
   - For every i-th square on the line, P puts two (identical) piles of x_i on the square. We now have

     on the line consisting the k squares.
   - Let $\{y_1, \ldots, y_{9-k}\} := \{1, 2, \ldots, 9\} - \{x_1, \ldots, x_k\}$. P puts on the associated triangle[†] a pile of y_j for every j, $1 \le j \le 9 - k$, that is, P puts $(9 - k)$ piles that do not appear as digits on the line. We now have

     on the triangle. Notice that because there are horizontal lines and vertical lines, four piles in total have been put on each square.
2. The verifier V does the following for each line.
   - V randomly picks a pile on every square of the line; V also picks all the piles on the associated triangle. Now, V has nine piles, and V puts each pile into an identical envelope. All the nine envelopes are shuffled and then all cards are taken out of every envelope to check whether the nine piles encode all distinct digits.
   - V randomly picks a pile on every square of the line, accumulate all the cards in the picked piles, and then shuffles all the cards. Then, V reveals the shuffled cards to check whether the number of black cards is equal to the clue.

One can verify that the BDDL protocol above satisfies

[†]For a line, the *associated triangle* means the triangle whose clue is associated with the line.

the three properties of a zero-knowledge proof, i.e., completeness, extractability, and zero-knowledgeness [7]. As for extractability, let us consider a situation where V is convinced with the highest probability despite illegal input by P. Such a situation was shown to occur when three piles of the same digit and another pile of a different digit were put by P on each square; in this case, a probability that V is convinced is at most 1/4 [7]. Thus, the BDDL protocol does not achieve perfect extractability.

Denote the number of squares by n and the number of lines by ℓ. Then, the number of cards put on the table after Step 1 of the BDDL protocol is $81\ell + 18n$. In order for P to be able to put such a number of cards, $81\ell + 81\ell$ cards suffice; if we adjust the numbers of black and red cards taking the values of all clues into account, we can reduce the number of required cards so that $81\ell + 18n$ cards are sufficient. Furthermore, the number of required shuffles is 2ℓ. See Table 1 again.

3. Our Improved Protocol

In this section, we improve upon the BDDL protocol introduced in the previous section so that we have a protocol achieving perfect extractability. To this end, we make use of the ideas behind the copy protocols in card-based cryptography.

We first provide a copy protocol for duplicating a sequence of nine distinct piles in Sect. 3.1. Using the copy protocol as a sub-protocol, we present our improved protocol in Sect. 3.2. Furthermore, we verify the validity of our improved protocol in Sect. 3.3.

3.1 Duplicating Commitments

Remember that the BDDL protocol presented in Sect. 2.2 uses a sequence of nine distinct piles in the first item of Step 2 to confirm whether the consecutive piles (put by P) in a line are all different. We call such a sequence of nine piles a *line commitment*, and consider how to duplicate a line commitment.

We hereinafter regard a line commitment

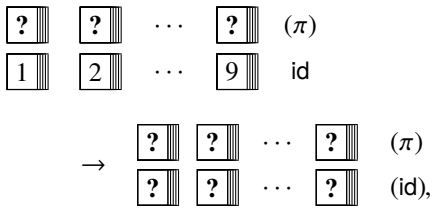
as a permutation

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \end{pmatrix},$$

which belongs to the symmetric group of degree nine, denoted by S_9.

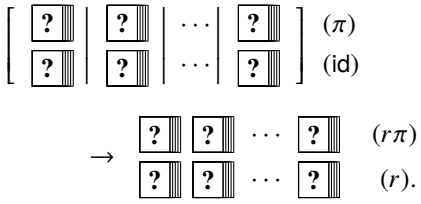
Given a line commitment, which corresponds to a permutation $\pi \in S_9$, along with additional cards, the following procedure enables V to make two copied line commitments.

1. Using nine additional face-up piles, V puts a line commitment corresponding to the identity permutation id, as follows:



where $\boxed{x}$ represents a (face-up) pile of x, $1 \le x \le 9$, and a permutation with parentheses indicates that cards are face-down.

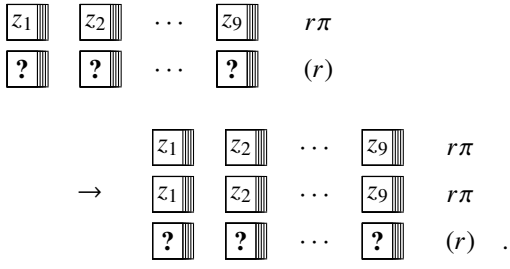
2. V accumulates two piles in each column and put them in an envelope without changing their order and then the nine envelopes are shuffled. After the shuffle, V takes the piles from each envelope. Now, for a uniformly distributed random permutation $r \in S_9$, we have



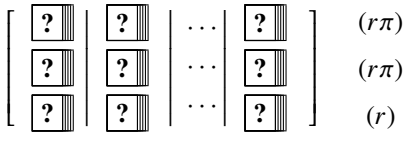
3. V reveals the line commitment in the first row to confirm that the piles $z_1, z_2, \cdots, z_9$ in the line commitment are all different, where

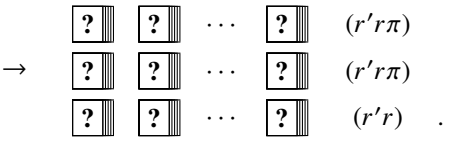
$$r\pi = \begin{pmatrix} 1 & 2 & \dots & 9 \\ z_1 & z_2 & \dots & z_9 \end{pmatrix}.$$

Next, V generates the same line commitment as the top row and places at the top so that we have

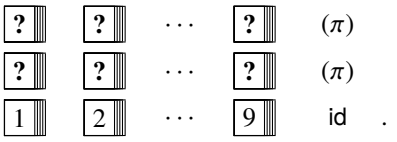


4. Turn over all the face-up cards so that we have two line commitments to $r\pi$. In a similar manner to Step 2, V accumulates three piles in each column and put them in an envelope without changing their order and then the nine envelopes are shuffled. After the shuffle, V takes the piles from each envelope. Now, for a uniformly distributed random permutation $r' \in S_9$, we have





5. Finally, V reveals the line commitment in the bottom row and then V sorts the columns by applying $(r'r)^{-1}$ to each row so that we obtain two copied line commitments to π:



It should be noted that the revealed cards can be reused in the next "copy" task.

Note that, in this copy protocol, V can not only duplicate given piles on the line but also can convince that the piles encode all distinct digits, as seen in Step 3. Thus, if we slightly change Step 3 so that only one line commitment to $r\pi$ is put, we can have a protocol for checking whether all piles are different. We will also use this in our main protocol in the next subsection.

The construction of our copy protocol partially borrows the idea given in [16] and [17], namely, indexing a pile with cards, regarding a sequence of piles as a permutation and how to generate the inverse of a permutation.

3.2 Description of Our Improved Protocol

Given a Kakuro puzzle, which has ℓ_h horizontal lines and ℓ_v vertical lines with $\ell_h \geq \ell_v$, our protocol proceeds as follows. (If $\ell_h < \ell_v$, interchange the words 'horizontal' and 'vertical' and interchange 'ℓ_h' and 'ℓ_v' in the protocol description below.)

1. The prover P holds $81(\ell_h + \ell_v)$ cards.
2. P puts on each empty square a pile of the digit of the solution. For every vertical line, P puts on the associated triangle all piles that do not appear in the line. After the placement, there are $81\ell_v$ cards put on the table.
3. Next, for every horizontal line, P puts on the associated triangle all piles that do not appear in the line. This operation is performed with remaining $81\ell_h$ cards; however, the cards which were not used must be secret from the verifier V.
4. P and V do the following.
   - For each horizontal line, take all piles on the line and the associated triangle to create a line commitment, and apply the (modified) copy protocol (without duplication) shown in Sect. 3.1 to confirm that the digits are all different.
   - P collects the cards which were not used in Step 3 and the cards which are on the associated triangles of the horizontal lines, and then, these cards are shuffled. These cards can be reused for the later applications of the copy protocol.
   - For each vertical line, take all piles on the line and the associated triangle to create a line commitment, and apply the copy protocol shown in Sect. 3.1 to make two copied line commitments (and confirm that the digits are all different).
5. For each line, V picks a pile on every square of the line, accumulate all the cards in the picked piles, and then all the cards are shuffled. Finally, V reveals the shuffled cards to check whether the number of black cards is equal to the associated clue.

As described in Step 1, the number of cards to execute this protocol is $81\ell+81$ (where $\ell = \ell_h+\ell_v$ is the total number of clues). The number "81" comes from the reusable cards noted in Step 5 in our copy protocol shown in Sect. 3.1. The number of required shuffles is $3\ell + 1$. Our protocol is implementable with fewer cards than the BDDL protocol (when $n \geq 5$) as shown in Table 1.

3.3 The Validity

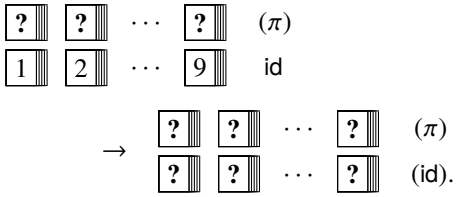
One can easily confirm the three properties of zero-knowledge proof of our improved protocol presented in Sect. 3.2, as follows.

Completeness If P puts a line commitment according to the solution, V is absolutely convinced that the piles on the lines are all different in Step 3 in Sect. 3.2. In addition, since the number of black cards in the piles put by P on each line is equal to the clue, by confirming that, V is absolutely convinced.

(Perfect) extractability Since V uses the duplicated piles for the verification, the piles (put by V) in a square are guaranteed to be identical. Therefore, if P does not know the solution, V always finds an illegal input.

Zero-knowledgeness We assume a simulator S which simulates the conversation with V. S does not know the solution of a Kakuro puzzle while S is permitted to exchange piles with other piles during the shuffle action. If we let S act as follows, the conversation of S and the one of P are indistinguishable from V (here, we only show zero-knowledgeness of our copy protocol because the other part of our improved protocol is quite similar to the BDDL protocol presented in Sect. 2.2).

- Using nine additional piles, V puts a line commitment to the identity permutation id as follows.



- V accumulates two piles in each column and put them in an envelope without changing its order and

then V shuffles the nine envelopes. At this time, S exchanges the envelopes with other envelopes containing any distinct nine piles. After the shuffle, V takes the piles from each envelope. From now on, V continues the same procedure of the protocol.

4. Our Numbered-Card Protocol

In this section, we propose another physical zero-knowledge proof protocol for Kakuro that has perfect extractability property. Unlike the previous sections, this protocol uses a deck of cards numbered from 1 to 9 such as $\boxed{1}\boxed{2}\cdots\boxed{9}$ with identical back $\boxed{?}$. Note that cards having the same number need to be identical.

Given a Kakuro puzzle whose solution is known to the prover P, our numbered-card protocol proceeds as follows, although we omit the details of the copy protocol of the numbered-card version because it is quite similar to the one in Sect. 3.1, i.e., just replacing a pile with a numbered card.

1. P and V do the following.
   - The prover P puts on each empty square a card having the number of the solution.
   - For every horizontal line, P puts on the associated triangle all cards that do not appear in the line.
   - For each horizontal line, take all piles on the line and the associated triangle to create a line commitment, and apply the copy protocol shown in Sect. 3.1.
2. P and V do the following for each line.
   - Assume that the line consists of k squares. P generates all possible combinations of k cards where the sum of numbers is equal to the corresponding clue except for the combination of the solution. P puts each k cards into an identical envelope and places them on the associated triangle. At this time, P needs to put them so that V does not know the order of each k cards.
   - V picks a card on every square of the line, shuffle the cards and puts them into an envelope; if there are some envelopes on the associated triangle, V also picks all the envelopes. All the envelopes are shuffled and then all the cards are taken out to check whether there are all combinations of k cards such that the sum of numbers is equal to the clue.

Although P has to calculate all combinations satisfying the condition in the first item of Step 2, the number of possible combinations is restricted because Kakuro deals with a number from 1 to 9. The case where P has to put the largest number of cards on an associated triangle is when P has to consider $c = 25$ and $k = 5$ (where c is the clue number). In this case, there are possible 12 piles of 5 cards, and hence, P has to put on the associated triangle a number $55\,(= 5 \times 12 - 5)$ of cards except the solution. This is the worst case; the number of required cards for this protocol depends much on a puzzle instance.

One can easily confirm the three properties of zero-knowledge proof of this protocol because it is quite similar to those of our improved protocol presented in Sect. 3.3 except a candidate list. As for zero-knowledgeness, it is obvious that S can easily exchange envelopes with the ones corresponding to the candidate list during the shuffle.

This protocol is designed for the standard Kakuro, which handles numbers 1 to 9, implying that the size of any candidate list is bounded by a constant.

5. Conclusion

In this paper, by applying the ideas behind the copy protocols in card-based cryptography, we proposed two card-based physical zero-knowledge proof protocols for Kakuro with perfectly extractability. Our protocols do not need to be repeated, and hence, they are efficient.

Acknowledgments

We thank the anonymous referees, whose comments helped us to improve the presentation of the paper. This work was supported by JSPS KAKENHI Grant Number 17K00001.

References

[1] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," SIAM J. Comput., vol.18, no.1, pp.186–208, 1989.
[2] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems," J. ACM, vol.38, no.3, pp.691–729, 1991.
[3] T. Seta, "The complexities of puzzles, cross sum, and their another solution problems (ASP)," Senior Thesis, Faculty of Science, The University of Tokyo, 2002.
[4] G. Kendall, A.J. Parkes, and K. Spoerer, "A survey of NP-complete puzzles," ICGA J., vol.31, no.1, pp.13–34, 2008.
[5] O. Ruepp and M. Holzer, "The computational complexity of the Kakuro puzzle, revisited," Fun with Algorithms, 5th International Conference, FUN 2010, Proceedings, P. Boldi and L. Gargano, eds., Lecture Notes in Computer Science, vol.6099, pp.319–330, Springer, Ischia, Italy, June 2010.
[6] R.G. Hunter, "Zero-knowledge proofs for puzzles," Senior Thesis, Department of Computer Science, Marrimack College, 2016.
[7] X. Bultel, J. Dreier, J. Dumas, and P. Lafourcade, "Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen," 8th International Conference on Fun with Algorithms, FUN 2016, E.D. Demaine and F. Grandoni, eds., LIPIcs, vol.49, pp.8:1–8:20, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, La Maddalena, Italy, June 2016.
[8] T. Sasaki, T. Mizuki, and H. Sone, "Card-based zero-knowledge proof for Sudoku," 9th International Conference on Fun with Algorithms, FUN 2018, H. Ito, S. Leonardi, L. Pagli, and G. Prencipe, eds., LIPIcs, vol.100, pp.29:1–29:10, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, La Maddalena, Italy, June 2018.
[9] R. Gradwohl, M. Naor, B. Pinkas, and G.N. Rothblum, "Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles," Theory Comput. Syst., vol.44, no.2, pp.245–268, 2009.
[10] Y. Chien and W. Hon, "Cryptographic and physical zero-knowledge

proof: From Sudoku to Nonogram," Fun with Algorithms, 5th International Conference, FUN 2010, Proceedings, P. Boldi and L. Gargano, eds., Lecture Notes in Computer Science, vol.6099, pp.102–112, Springer, Ischia, Italy, June 2010.

[11] B. den Boer, "More efficient match-making and satisfiability: *The five card trick*," Advances in Cryptology - EUROCRYPT'89, Workshop on the Theory and Application of of Cryptographic Techniques, Proceedings, J. Quisquater and J. Vandewalle, eds., Lecture Notes in Computer Science, vol.434, pp.208–217, Springer, Houthalen, Belgium, April 1989.

[12] C. Crépeau and J. Kilian, "Discreet solitary games," Advances in Cryptology - CRYPTO'93, 13th Annual International Cryptology Conference, Proceedings, D.R. Stinson, ed., Lecture Notes in Computer Science, vol.773, pp.319–330, Springer, Santa Barbara, California, USA, Aug. 1993.

[13] T. Mizuki, M. Kumamoto, and H. Sone, "The five-card trick can be done with four cards," Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings, X. Wang and K. Sako, eds., Lecture Notes in Computer Science, vol.7658, pp.598–606, Springer, Beijing, China, Dec. 2012.

[14] A. Koch, S. Walzer, and K. Härtel, "Card-based cryptographic protocols using a minimal number of cards," Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Proceedings, Part I, T. Iwata and J.H. Cheon, ed.. Lecture Notes in Computer Science, vol.9452, pp.783–807, Springer, Auckland, New Zealand, Nov.-Dec. 2015.

[15] J. Kastner, A. Koch, S. Walzer, D. Miyahara, Y. Hayashi, T. Mizuki, and H. Sone, "The minimum number of cards in practical card-based protocols," Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Proceedings, Part III, T. Takagi and T. Peyrin, eds., Lecture Notes in Computer Science, vol.10626, pp.126–155, Springer, Hong Kong, China, Dec. 2017.

[16] Y. Hashimoto, K. Shinagawa, K. Nuida, M. Inamura, and G. Hanaoka, "Secure grouping protocol using a deck of cards," Information Theoretic Security - 10th International Conference, ICITS 2017, Proceedings, J. Shikata, ed., Lecture Notes in Computer Science, vol.10681, pp.135–152, Springer, Hong Kong, China, Nov.-Dec. 2017.

[17] T. Ibaraki and Y. Manabe, "A more efficient card-based protocol for generating a random permutation without fixed points," 2016 Third International Conference on Mathematics and Computers in Sciences and in Industry (MCSI), pp.252–257, Aug. 2016.

Daiki Miyahara received his B.E. degree from Tohoku University, Japan, in 2017. He is currently a second grade postgraduate student of Tohoku University. His research interests include cryptology and information security. He is a member of IEICE and IACR.

Tatsuya Sasaki received his B.E. degree from Tohoku University, Japan, in 2017. He is currently a second grade postgraduate student of Tohoku University. His research interests include cryptology and information security. He is a member of IEICE.

Takaaki Mizuki received his B.E. degree in information engineering and his M.S. and Ph.D. degrees in information sciences from Tohoku University, Japan, in 1995, 1997, and 2000, respectively. He is currently an associate professor of the Cyberscience Center, Tohoku University. His research interests include card-based cryptography. He is a member of IEICE, IEEE, and IPSJ.

Hideaki Sone received his B.E. degree in electrical engineering, and his M.E. and Ph.D. degrees in electrical communications from Tohoku University, Japan. He joined the Faculty of Engineering, Tohoku University, as a research associate in 1980. He became a professor at the Information Synergy Center, Tohoku University in 2001. His main research interests lie in the fields of information telecommunication systems and instrumentation electronics. He is a member of IEICE, IEEE, SICE, and IEEJ.