Card-Based Zero-Knowledge Proof for Sudoku

Tatsuya Sasaki
Graduate School of Information Sciences, Tohoku University
6–3–09 Aramaki-Aza-Aoba, Aoba, Sendai 980–8579, Japan
tatsuya.sasaki.p2@dc.tohoku.ac.jp

Takaaki Mizuki
Cyberscience Center, Tohoku University
6–3 Aramaki-Aza-Aoba, Aoba, Sendai 980–8578, Japan
tm-paper+cardsudk@g-mail.tohoku-university.jp

Hideaki Sone
Cyberscience Center, Tohoku University
6–3 Aramaki-Aza-Aoba, Aoba, Sendai 980–8578, Japan

Abstract

In 2009, Gradwohl, Naor, Pinkas, and Rothblum proposed physical zero-knowledge proof protocols for Sudoku. That is, for a puzzle instance of Sudoku, their excellent protocols allow a prover to convince a verifier that there is a solution to the Sudoku puzzle and that he/she knows it, without revealing any information about the solution. The possible drawback is that the existing protocols have a soundness error with a non-zero probability or need special cards (such as scratch-off cards). Thus, in this study, we propose new protocols to perform zero-knowledge proof for Sudoku that use a normal deck of playing cards and have no soundness error. Our protocols can be easily implemented by humans with a reasonable number of playing cards.

2012 ACM Subject Classification Security and privacy → Information-theoretic techniques

Keywords and phrases Zero-knowledge proof, Card-based cryptography, Sudoku

Digital Object Identifier 10.4230/LIPIcs.FUN.2018.29

Funding This work was supported by JSPS KAKENHI Grant Number JP17K00001.

Acknowledgements The authors would like to express their deepest appreciation to Kazumasa Shinagawa for helpful discussions during this work.

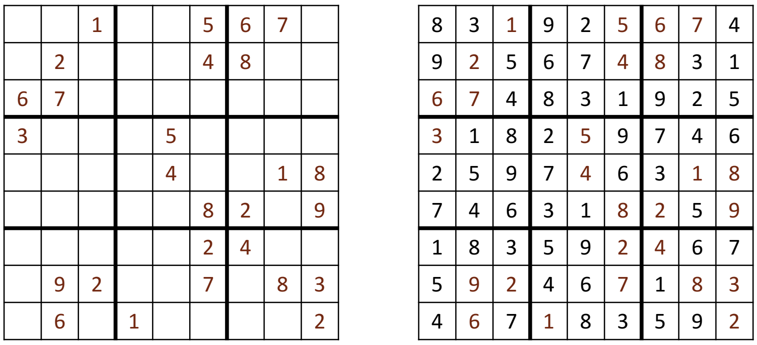
1 Introduction

Sudoku is one of the most famous puzzles. In a standard challenge, a 9×9 grid is used, which is divided into 3×3 subgrids. Some of the cells are already filled with numbers between 1 and 9. The goal of Sudoku is to fill all the empty cells with numbers so that each row, each column, and each subgrid contains all the numbers from 1 to 9. Figure 1 shows an example of a standard Sudoku challenge, and its solution.

We address a generalized version of Sudoku in this study. That is, a Sudoku puzzle where a grid is $n \times n$ cells, a subgrid is $k \times k$ cells, and numbers from 1 to n are used. Note that $n = k^2$; the standard size of a Sudoku puzzle corresponds to $n = 9$ and $k = 3$.

We solicit zero-knowledge proof protocols for Sudoku. That is, for a certain Sudoku puzzle, we assume a prover P who knows the solution to the Sudoku puzzle and a verifier V who does not know it, and suppose that P wants to convince V of the following without revealing any information about the solution.

9th International Conference on Fun with Algorithms (FUN 2018).
Editors: Hiro Ito, Stefano Leonardi, Linda Pagli, and Giuseppe Prencipe; Article No. 29; pp. 29:1–29:10
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



		1			5	6	7	
	2				4	8		
6	7							
3				5				
				4			1	8
					8	2		9
					2	4		
	9	2			7		8	3
	6		1					2

8	3	1	9	2	5	6	7	4
9	2	5	6	7	4	8	3	1
6	7	4	8	3	1	9	2	5
3	1	8	2	5	9	7	4	6
2	5	9	7	4	6	3	1	8
7	4	6	3	1	8	2	5	9
1	8	3	5	9	2	4	6	7
5	9	2	4	6	7	1	8	3
4	6	7	1	8	3	5	9	2

Figure 1 Example of the standard Sudoku challenge, and its solution.

- There is a solution to the puzzle;
- P knows the solution.

Unlike in conventional zero-knowledge proof (see, e.g., [4]), in our setting we do not want to use electronic devices such as computers. Instead, we want to use only everyday items to execute a protocol manually. The prover P and the verifier V are assumed to be in the same place. Such a restricted zero-knowledge proof is called a *physical zero-knowledge proof* [1, 2, 5].

In 2009, Gradwohl, Naor, Pinkas, and Rothblum proposed a few physical zero-knowledge proof protocols for Sudoku [5]. Among them, Protocol 3 (hereinafter referred to as *GNPR Protocol 3*) utilizes a deck of cards having numbers on their faces, such as playing cards[1]. This protocol needs $3n^2$ cards and has a soundness error with a non-zero probability. In contrast, Protocol 5 (hereinafter referred to as *GNPR Protocol 5*) avoids soundness error by utilizing special cards (namely, scratch-off cards that allow the colors to be covered) and scissors[2]. Unfortunately, GNPR Protocol 5 consumes non-reusable scratch-off cards at every execution of the protocol. Therefore, it is preferable to construct a protocol that can be implemented with only reusable everyday objects such as playing cards.

Thus, in this paper, we propose zero-knowledge proof protocols that satisfy the following: (i) they utilize the same items as GNPR Protocol 3, namely a standard deck of playing cards, (ii) they are implementable with fewer cards than GNPR Protocol 3, and (iii) they have no soundness error. The main idea behind our protocols is to apply techniques of card-based cryptography (see, e.g., [6], [8]). In particular, copy computation, which is an important primitive in the field of card-based cryptography, prevents the prover P from inputting incorrect numbers.

The remainder of this paper is organized as follows. In Section 2, we review zero-knowledge proof, GNPR Protocol 3, and GNPR Protocol 5. In Section 3, we present our proposed protocols. In Section 4, we compare our protocols with the existing ones, and conclude this paper.

2 Preliminaries

In this section, we first review zero-knowledge proof and then introduce two existing protocols, GNPR Protocols 3 and 5.

[1] Protocols 1 and 2 presented in [5] are conventional (non-physical) zero-knowledge proof protocols.
[2] Protocol 4 in [5] is a variation of GNPR Protocol 3.

2.1 Zero-Knowledge Proof

A zero-knowledge proof is an interactive proof between a prover P and a verifier V. They both have an instance of problem x and only P knows w, which is some information about a solution or a witness. The verifier V is computationally bounded so that V cannot obtain w from x. Under these assumptions, P wants to convince V that he/she knows w without revealing any information about w. Such a proof is called a *zero-knowledge proof*, which must satisfy the following three properties.

Completeness If P knows w, P is able to convince V.

Soundness If P does not know w, P cannot convince V (with a high probability).

Zero-knowledge V cannot obtain any information about w.

The probability that V will be convinced although P does not know w is called the *soundness error*. If we have a zero-knowledge proof protocol, the soundness error of which is $\delta > 0$, repeating the protocol ℓ times allows V to detect that P does not know w with a probability $1 - \delta^{\ell}$. Therefore, in general, even if the soundness error of a protocol is not 0, we can in practice establish zero-knowledge proof with a negligible soundness error by repeating the protocol. However, since we assume that a protocol is executed by human hands, it is impractical to repeat the protocol many times. Therefore, it is indispensable to design a protocol with no soundness error.

A zero-knowledge proof was first defined by Goldwasser, Micali, and Rackoff [4], and it was proved that (computational) zero-knowledge proofs exist for any NP problems [3]. Because it is known that Sudoku is NP-complete [9], we can construct conventional (computational) zero-knowledge proof protocols for it [5]. Remember, however, that this paper is focused not on a conventional zero-knowledge but on a physical zero-knowledge proof for Sudoku. Hence, we introduce the existing physical protocols, GNPR Protocols 3 and 5, in the following two subsections.

2.2 Gradwohl, Naor, Pinkas, and Rothblum Protocol 3

Here, we review GNPR Protocol 3 [5]. This protocol utilizes physical cards, the face side of each of which has one number between 1 and n, such as $\boxed{1}$ $\boxed{2}$ ... $\boxed{n}$; all the back sides are identical, for example, $\boxed{?}$ $\boxed{?}$ ... $\boxed{?}$. The protocol uses $3n$ sets of such n cards, namely, $3n^2$ cards in total.

Before presenting the protocol, we define a shuffle operation for cards. Given a sequence of ℓ cards $(c_1, c_2, c_3, ..., c_\ell)$, a *shuffle* results in a sequence

$$\left(c_{r^{-1}(1)}, c_{r^{-1}(2)}, c_{r^{-1}(3)}, ..., c_{r^{-1}(\ell)}\right),$$

where $r \in S_\ell$ is a uniformly random permutation and S_ℓ is the symmetric group of degree ℓ.

GNPR Protocol 3 proceeds as follows.

- The prover P places three face-down cards on each cell according to the Sudoku solution. On the filled-in cells, P places three face-up cards corresponding to the numbers filled in. After V confirms the values of the face-up cards, P turns them over.
- The verifier V picks one card randomly from each cell of a row to make a packet of n cards corresponding to the row. Because there are n rows, n packets are created. The same procedure is applied for each column and each subgrid. Thus, V makes $3n$ packets in total and passes them to P.
- P who received the packets from V applies a shuffle to the cards in each packet and returns the n shuffled packets to V.
- V opens all the cards in all the packets and checks that each packet contains all the numbers from 1 to n.

This is GNPR Protocol 3, which satisfies the three properties of zero-knowledge proof, as follows.

Completeness If P places the face-down cards correctly according to the solution, every packet made by V must contain all the numbers from 1 to n. By checking them, V is convinced that all the cards have been placed according to the solution. Furthermore, V is convinced that the packets are not a solution to another puzzle instance, because V sees the face-up cards corresponding to the values of the filled-in cells.

Soundness Consider a situation where V is convinced, in spite of an illegal input by P. Such a situation occurs when the three cards placed on each cell are not identical. The soundness error was shown to be at most 1/9 [5].

Zero-knowledge Assume a simulator S that simulates the conversation between P and V. Although S does not have any information about the witness w, S is allowed to replace packets with arbitrary packets. S acts as follows.

- The simulator S places three arbitrary face-down cards on each cell. On filled-in cells, S places three face-up cards, according to the filled-in cells. After V confirms the values, S turns them over.
- V makes $3n$ packets using the same procedure as in GNPR Protocol 3 and passes them to S.
- S shuffles the cards in each packet. Before passing the packets to V, S replaces them all by new ones, each of which contains all the cards numbered from 1 to n.
- V opens all the packets and checks that each packet contains all the numbers from 1 to n.

Since the conversation of S is indistinguishable from that of P, the protocol satisfies the zero-knowledge property.

In this protocol, P places three cards on each cell, and hence, the protocol uses $3n^2$ cards in total. For example, in the case of a Sudoku puzzle consisting of a 9 × 9 grid, the protocol needs 243 cards. Because a physical zero-knowledge proof protocol is supposed to be executed by human hands, it is preferable that the number of cards used in a protocol is as small as possible. In addition, as mentioned in Section 2.1, a protocol with no soundness error is also preferable.

2.3 Gradwohl, Naor, Pinkas, and Rothblum Protocol 5

As mentioned in the previous subsection, a soundness error during an execution of GNPR Protocol 3 would occur if the prover P did not place three identical cards on a cell. Therefore, if we could guarantee that all three cards placed on each cell are identical, a soundness error would never occur. This can be realized by using the following special scratch-off cards [5].

Consider scratch-off cards that cover any one of n colors. Assume that P and V agree on a one-to-one correspondence from a color to a Sudoku number. Suppose that only P knows which scratch-off card covers which color. Under these assumptions, P places such a scratch-off card on each cell according to the Sudoku solution. Next, V cuts every scratch-off card into three pieces with scissors so that they have three small cards having the same shape and size. Because the verifier V cuts the cards him/herself, it is possible to guarantee that the three small obtained cards are identical and cover the same color. Thus, scratch-off cards and scissors provide a protocol with no soundness error; this is GNPR Protocol 5 [5].

However, such scratch-off cards do not seem to be ordinary everyday items, and in addition, they are non-reusable. Therefore, in the next section, we propose a method to guarantee that the three cards placed on each cell are identical without any use of special cards.

3 Our Protocols

In this section, we propose efficient zero-knowledge proof protocols for Sudoku with no soundness error in which card-based cryptography perspectives are applied. Our protocols utilize the same type of cards as GNPR Protocol 3, but require fewer cards. We first design a fundamental protocol, and then modify it to attain more efficient protocols.

The outline of our fundamental protocol is as follows.

- P places exactly one face-down card on each cell corresponding to the solution (as seen in Section 3.1).
- V checks that the format of the packet of face-down cards placed on each subgrid is correct while making two identical copies of the packet (as seen in Section 3.2), which will be used for verifying rows and columns.
- V verifies that each row and each column contains all the numbers from 1 to n (as seen in Sections 3.3 and 3.4).

In Section 3.5, we show that our protocols satisfy the zero-knowledge proof properties.

3.1 Commitment as Input

In our protocols, the prover P places a single face-down card on each cell according to the solution. On filled-in cells, P places face-up cards. After V confirms the value of the face-up cards, P turns them over. Now, there are exactly n^2 cards placed on the grid. We call a sequence of n face-down cards corresponding to each subgrid a *commitment*.

For example, in the case of the top-left subgrid in Figure 1, P places nine cards according to the solution; after V confirms the value of the face-up cards, P turns them over:

$$\begin{array}{ccc} \boxed{?} & \boxed{?} & \boxed{1} \\ \boxed{?} & \boxed{2} & \boxed{?} \\ \boxed{6} & \boxed{7} & \boxed{?} \end{array} \quad \rightarrow \quad \begin{array}{ccc} \boxed{?} & \boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?} & \boxed{?} \\ \boxed{?} & \boxed{?} & \boxed{?} \end{array} \quad .$$

This is a commitment corresponding to this subgrid, and we regard it as a sequence:

$$\overbrace{\boxed{?}\;\boxed{?}\;\boxed{?}}^{\text{1st row}}\;\overbrace{\boxed{?}\;\boxed{?}\;\boxed{?}}^{\text{2nd row}}\;\overbrace{\boxed{?}\;\boxed{?}\;\boxed{?}}^{\text{3rd row}} \quad .$$

Thus, P and V generate n commitments corresponding to n subgrids.

3.2 Subgrid Copy

After n commitments, each of which corresponds to a subgrid, have been generated, P and V want to copy these commitments to verify each row and each column. In addition, they also want to make sure that each commitment contains all the numbers from 1 to n. Therefore, in this subsection, we propose a method to verify that the format of a given commitment is correct, which involves making two copied commitments.

To this end, we first introduce a well-known shuffle operation called *pile-scramble shuffle* [7]. Assume that there are m piles, each of which consists of the same number of face-down cards; we denote this by

$$(pile_1, pile_2, pile_3, ..., pile_m).$$

For such a sequence of piles, applying a pile-scramble shuffle results in

$$(pile_{r^{-1}(1)}, pile_{r^{-1}(2)}, pile_{r^{-1}(3)}, ..., pile_{r^{-1}(m)}),$$

where $r \in S_m$ is a uniformly distributed random permutation. A pile-scramble shuffle can be implemented with the help of clips, envelopes, or similar items.

We now borrow two existing ideas: (i) a method for regarding a commitment as a permutation and a technique for inverting a permutation, which were given by Hashimoto, Shinagawa, Nuida, Inamura, and Hanaoka [6], and (ii) a technique for checking the format of a sequence of face-down cards, which was given by Mizuki and Shizuya [8]. That is, we regard a commitment consisting of n cards as a permutation $v \in S_n$:

$$\boxed{?}\ \boxed{?}\ \boxed{?}\ \cdots\ \boxed{?}\qquad (v),$$

where a card having number i, $1 \leq i \leq n$, on its face side is placed at the $v(i)$-th position, and a permutation with parentheses, such as (v), means that the permutation is hidden (because the cards are face-down). Given a commitment to a permutation $v \in S_n$, we construct a method to check whether the commitment consists of all the numbers from 1 to n that involves making two identical copied commitments. We call this method *subgrid copy* and it operates as follows.

1. V puts n cards numbered from 1 to n in this order to generate a card sequence corresponding to the identity permutation id under the commitment to v:

$$\begin{array}{ccccccl} \boxed{?} & \boxed{?} & \boxed{?} & \cdots & \boxed{?} & & (v) \\ \boxed{1} & \boxed{2} & \boxed{3} & \cdots & \boxed{n} & & \mathsf{id}\ . \end{array}$$

2. V turns over all the face-up cards in the bottom row and stacks the cards in each column so that there are n two-card piles:

$$\begin{array}{c|c|c|c} \boxed{?} & \boxed{?} & & \boxed{?} \\ \boxed{?} & \boxed{?} & \cdots & \boxed{?} \end{array}\quad \begin{array}{l} (v) \\ \mathsf{id}\ . \end{array}$$

P applies a pile-scramble shuffle to them and obtains a commitment to $rv \in S_n$ and a commitment to $r \in S_n$, where $r \in S_n$ is a uniformly distributed random permutation:

$$\left[\ \begin{array}{c|c|c|c} \boxed{?} & \boxed{?} & & \boxed{?} \\ \boxed{?} & \boxed{?} & \cdots & \boxed{?} \end{array}\ \right] \rightarrow \begin{array}{ccccl} \boxed{?} & \boxed{?} & \cdots & \boxed{?} & \quad (rv) \\ \boxed{?} & \boxed{?} & \cdots & \boxed{?} & \quad (r)\ . \end{array}$$

3. V turns over all the cards in the top row and checks the opened cards. If there are all cards numbered from 1 to n, V is convinced that the face-down cards placed by P on the subgrid are compatible with the puzzle solution. Since V learns only the value of rv, which is also a random permutation, no information about v leaks.
4. P sorts the n columns so that the top row becomes id. This means that a permutation $(rv)^{-1}$ is multiplied to each row, and hence, the bottom row becomes a commitment to $(rv)^{-1}r = v^{-1}$, i.e., the inverse of v:

$$\begin{array}{ccccccl} \boxed{1} & \boxed{2} & \boxed{3} & \cdots & \boxed{n} & & \mathsf{id} \\ \boxed{?} & \boxed{?} & \boxed{?} & \cdots & \boxed{?} & & (v^{-1}). \end{array}$$

5. From now on, we make two copied commitments to v. V places two identity permutations id under the commitment to v^{-1};

$$\begin{array}{cccccc} \boxed{?} & \boxed{?} & \boxed{?} & \ldots & \boxed{?} & (v^{-1}) \\ \boxed{1} & \boxed{2} & \boxed{3} & \ldots & \boxed{n} & \mathsf{id} \\ \boxed{1} & \boxed{2} & \boxed{3} & \ldots & \boxed{n} & \mathsf{id} \end{array} .$$

6. By applying a procedure similar to Steps 2 and 3, the bottom-most two rows become commitments to v:

$$\begin{array}{cccccc} \boxed{1} & \boxed{2} & \boxed{3} & \ldots & \boxed{n} & \mathsf{id} \\ \boxed{?} & \boxed{?} & \boxed{?} & \ldots & \boxed{?} & (v) \\ \boxed{?} & \boxed{?} & \boxed{?} & \ldots & \boxed{?} & (v) \end{array} .$$

Thus, we can verify that a given commitment corresponding to a subgrid contains all the numbers from 1 to n, while making two copied commitments. This requires $2n$ cards in addition to the input commitment. The copied commitments are used for verifying rows and columns, as we describe in the complete protocol in the next subsection.

3.3 Fundamental Protocol

We are now ready to describe our fundamental protocol. The protocol proceeds as follows.

1. P places a commitment on each subgrid (as already described in Section 3.1).
2. V and P apply the subgrid copy (as explained in Section 3.2) to all subgrids. Then, there are two cards on each cell of the grid. Note that the verification that every commitment contains all cards numbered from 1 to n has been completed.
3. As in a similar way to GNPR Protocol 3, P makes $2n$ packets corresponding to n rows and n columns. Each packet is shuffled.
4. V opens all the packets and checks that each packet includes all the numbers from 1 to n.

Let us count how many cards we use in this protocol. Immediately before applying the subgrid copy to the final subgrid, there are $2(n^2 - n) + n$ cards on the grid. To apply the subgrid copy to the n-th subgrid, we need $2n$ more additional cards, and hence, we need $2n^2 + n$ cards in total. This is the maximum number of required cards during any execution. Therefore, the protocol requires $2n^2 + n$ cards.

3.4 Compact Protocol

In the fundamental protocol presented in Section 3.3, a subgrid copy was applied to all the subgrids before the verifications of each row and each column were performed. However, we do not have to wait until all the copy actions of n subgrids are complete; when verification of rows or columns becomes applicable, we can stop the subgrid copy action, and instead, start to verify rows or columns so that we have reusable opened cards, and consequently, it is possible to reduce the number of required cards. Thus, we have a compact protocol, as follows.

As mentioned in Section 1, an $n \times n$ grid of Sudoku can be regarded as a subgrid matrix of $k \times k$. We refer to such rows and columns of subgrids as subgrid-rows and subgrid-columns, respectively. The protocol proceeds as follows.

1. P places a commitment for each subgrid (as explained in Section 3.1).
2. V and P apply subgrid copy (as explained in Section 3.2) to all the subgrids in the first subgrid-row.
3. After Step 2, two cards are placed on each cell in the first subgrid-row. V verifies that each of the first k rows (which constitute the first subgrid-row) contains all the numbers from 1 to n. After the verification is complete, there is one card on every cell.
4. P and V repeat the same procedure as Steps 2 and 3 for every subgrid-row from the second to the (k-1)-th.
5. In the k-th subgrid-row, V and P will operate a similar procedure to verify the columns. First, the subgrid copy action is applied to the first subgrid in the k-th subgrid-row. Then, P and V verify the first k columns.
6. P and V repeat the same procedure as Step 5 for every subgrid-column from the second to the k-th.
7. Finally, P and V verify the rows in the k-th subgrid-row, so that the verification is complete for all rows, columns, and subgrids.

Let us consider at which point the number of used cards becomes largest. It is when the subgrid copy is applied to the last subgrid in Step 4, and at that point we use $n^2 + (k+1)n$ cards. Therefore, this compact protocol requires $n^2 + (k+1)n$ cards.

3.5 Correctness of Proposed Protocols

In this subsection, we show that our protocols proposed in Sections 3.3 and 3.4 satisfy the properties of zero-knowledge proof.

Completeness A prover P who knows the solution can place cards so that each row, each column, and each subgrid contains all the numbers. Whether the format of each subgrid is correct can be checked by using the subgrid copy, and whether the format of each row and each column is correct can be checked by using the copied commitments. Further, V is convinced that P's input is not a solution to another problem by comparing the face-up cards and the corresponding value of the filled-in cells.

Soundness Since P and V use copied commitments, it is guaranteed that the cards placed on each cell are identical. Therefore, the protocol has no soundness error.

Zero-Knowledge Assume a simulator that simulates the conversation as in Section 2.2. When verifying each row and each column, information about knowledge w does not leak for the same reason as in GNPR Protocol 3. Thus, it is sufficient to show the zero-knowledge property of the subgrid copy.

- The simulator S places one arbitrary face-down card on each cell. On filled-in cells, S places face-up cards. After V confirms them, S turns them over.
- V makes a sequence of n piles using the same procedure as Step 2 in Section 3.2, and passes them to S.
- S replaces the sequence by the following two identity permutations id. S applies a pile-scramble shuffle to the replaced sequence, and passes it to V.

$$\boxed{?}\ \boxed{?}\ \boxed{?}\ \cdots\ \boxed{?}\quad (\mathsf{id})$$
$$\boxed{?}\ \boxed{?}\ \boxed{?}\ \cdots\ \boxed{?}\quad (\mathsf{id})$$

- V opens the cards in the top row, and checks whether it contains all the numbers from 1 to n. Then, V operates the same procedure as described in Section 3.2, and outputs the bottom row.

- V operates Step 5 in Section 3.2, makes n packets, and passes them to S.
- S applies a pile-scramble shuffle to the sequence and passes it to V. In this case, S does not need to replace packets.

Since the conversation of S is indistinguishable from that of P, the protocol satisfies the zero-knowledge property.

4 Conclusion

In this paper, we proposed two card-based zero-knowledge proof protocols for Sudoku. We now compare our protocols with the existing protocols, GNPR Protocols 3 and 5, in terms of the number of cards, the number of shuffles, and the soundness error. Table 1 shows the performance of the protocols.

Our fundamental protocol and our compact protocol use $2n^2+n$ and $n^2+(k+1)n$ cards, respectively, as described in Sections 3.3 and 3.4. Let us count the number of shuffles in our proposed protocols. The subgrid copy procedure requires two shuffles, and this procedure is performed for each of n subgrids. The packet shuffle is performed once in the verification of each row and each column, and their number is $2n$. Therefore, the total number of shuffles is $2n+2n=4n$. Furthermore, as shown previously, our protocols have no soundness error. See Table 1 again.

In Sudoku's standard size $n=9$, the compact protocol can be implemented with less than half the number of cards used in GNPR Protocol 3 (117 versus 241). Further, when GNPR Protocol 3 is executed more than once, the number of shuffles is larger than that in our protocol. As compared to GNPR Protocol 5, the number of shuffles in our protocol is larger; however, in our opinion, a protocol that uses no special cards is superior.

Finally, we attempt to reduce the number of cards and shuffles further (but it looks crafty).

Further reduction of the number of cards

Thus far, we assumed that P places his/her inputs on all the cells simultaneously. If we allow P to input at multiple timings, it is possible to construct a protocol with fewer cards. The outline is as follows.

1. P places a commitment on one subgrid.
2. P and V apply the subgrid copy action to the subgrid.
3. P and V apply Steps 1 and 2 also to the other subgrids, and, as in the compact protocol, perform verification when it becomes possible to verify the rows and columns.

This protocol uses n^2+n cards. For example, when $n=9$, the number of required cards is 90.

Reduction of the number of shuffles[3]

In the subgrid copy action, P performs a pile-scramble shuffle twice; the output commitment obtained by the first pile-scramble shuffle is an inverse of v, and hence, P needs to shuffle again to obtain a commitment v. However, if P places a commitment of an inverse as input, P can omit one pile-scramble shuffle.

The performance of this crafty protocol is also shown in Table 1.

In the literature, for several puzzles other than Sudoku, physical zero-knowledge proof protocols have been proposed [1, 2]. Therefore, interesting future work is to design more efficient zero-knowledge proof protocols for those puzzles with the help of card-based cryptography.

[3] The idea was introduced by Kazumasa Shinagawa.

Table 1 Comparison of protocols

	# of cards	# of shuffles	Soundness error
GNPR Protocol 3	$3n^2$	$3n \times \ell$	at most $(\frac{1}{9})^\ell$
GNPR Protocol 5	n^2 (special cards)	$3n$	0
Fundamental Protocol	$2n^2 + n$	$4n$	0
Compact Protocol	$n^2 + (k+1)n$	$4n$	0
Crafty Protocol	$n^2 + n$	$3n$	0

References

1 Xavier Bultel, Jannik Dreier, Jean-Guillaume Dumas, and Pascal Lafourcade. Physical zero-knowledge proofs for akari, takuzu, kakuro and kenken. In Erik D. Demaine and Fabrizio Grandoni, editors, *8th International Conference on Fun with Algorithms, FUN 2016, June 8-10, 2016, La Maddalena, Italy*, volume 49 of *LIPIcs*, pages 8:1–8:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.FUN.2016.8.

2 Yu-Feng Chien and Wing-Kai Hon. Cryptographic and physical zero-knowledge proof: From sudoku to nonogram. In Paolo Boldi and Luisa Gargano, editors, *Fun with Algorithms, 5th International Conference, FUN 2010, Ischia, Italy, June 2-4, 2010. Proceedings*, volume 6099 of *Lecture Notes in Computer Science*, pages 102–112. Springer, 2010. doi:10.1007/978-3-642-13122-6_12.

3 Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991. doi:10.1145/116825.116852.

4 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989. doi:10.1137/0218012.

5 Ronen Gradwohl, Moni Naor, Benny Pinkas, and Guy N. Rothblum. Cryptographic and physical zero-knowledge proof systems for solutions of sudoku puzzles. *Theory Comput. Syst.*, 44(2):245–268, 2009. doi:10.1007/s00224-008-9119-9.

6 Yuji Hashimoto, Kazumasa Shinagawa, Koji Nuida, Masaki Inamura, and Goichiro Hanaoka. Secure grouping protocol using a deck of cards. In Junji Shikata, editor, *Information Theoretic Security - 10th International Conference, ICITS 2017, Hong Kong, China, November 29 - December 2, 2017, Proceedings*, volume 10681 of *Lecture Notes in Computer Science*, pages 135–152. Springer, 2017. doi:10.1007/978-3-319-72089-0_8.

7 Rie Ishikawa, Eikoh Chida, and Takaaki Mizuki. Efficient card-based protocols for generating a hidden random permutation without fixed points. In Cristian S. Calude and Michael J. Dinneen, editors, *Unconventional Computation and Natural Computation - 14th International Conference, UCNC 2015, Auckland, New Zealand, August 30 - September 3, 2015, Proceedings*, volume 9252 of *Lecture Notes in Computer Science*, pages 215–226. Springer, 2015. doi:10.1007/978-3-319-21819-9_16.

8 Takaaki Mizuki and Hiroki Shizuya. Practical card-based cryptography. In Alfredo Ferro, Fabrizio Luccio, and Peter Widmayer, editors, *Fun with Algorithms - 7th International Conference, FUN 2014, Lipari Island, Sicily, Italy, July 1-3, 2014. Proceedings*, volume 8496 of *Lecture Notes in Computer Science*, pages 313–324. Springer, 2014. doi:10.1007/978-3-319-07890-8_27.

9 Takayuki Yato and Takahiro Seta. Complexity and completeness of finding another solution and its application to puzzles. *IEICE Transactions*, 86-A(5):1052–1060, 2003.