Only Two Shuffles Perform Card-Based Zero-Knowledge Proof for Sudoku of Any Size

Kodai Tanaka* Shun Sasaki† Kazumasa Shinagawa‡ Takaaki Mizuki§

Abstract

Sudoku is a popular pencil puzzle where a player fills in the empty cells with numbers on an $n \times n$ board so that each row, column, and $(\sqrt{n} \times \sqrt{n})$-block must contain all the numbers from 1 to n; a typical puzzle size is $n = 9$. In 2007, Gradwohl, Naor, Pinkas, and Rothblum proposed a physical zero-knowledge proof protocol for Sudoku using a physical deck of cards; their card-based protocol requires $3n\ell$ shuffles, where ℓ is a security parameter to eliminate the soundness error. Since the invention of this seminal protocol, several soundness-error-free protocols were constructed to reduce the number of required shuffles; the state-of-the-art one was designed in 2023, which uses $7\sqrt{n}-5$ shuffles. In this paper, we show that only three or two shuffles are sufficient to construct a zero-knowledge proof protocol for Sudoku, no matter how large n is, i.e., we propose two card-based protocols using constant numbers (namely, 3 and 2) of shuffles. Our proposed protocols are simple and efficient enough for people to execute for a 9×9 Sudoku puzzle in reality.

1 Introduction

Sudoku[1] is one of the most popular pencil puzzles. A standard Sudoku puzzle consists of a 9×9 board divided into nine (3×3)-blocks along with numbers from 1 to 9 written on some cells of the board; we call such a cell (having a number) a *hint cell* and call other cells (having no number) *empty cells*. The objective is for a player to fill in the empty cells with numbers so that all the numbers from 1 to 9 appear in each row, each column, and each (3×3)-block. Figure 1 shows an example of a Sudoku puzzle and its solution. For a perfect square n, we have a (generalized) $n \times n$ Sudoku puzzle in a similar way.

This paper deals with zero-knowledge proof protocols for Sudoku.

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

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

Figure 1: A Sudoku puzzle and its solution.

*Tohoku University.
†Ibaraki University.
‡Ibaraki University / National Institute of Advanced Industrial Science and Technology.
§Tohoku University / National Institute of Advanced Industrial Science and Technology.
[1]Sudoku is a trademark or registered trademark of Nikoli Co., Ltd.

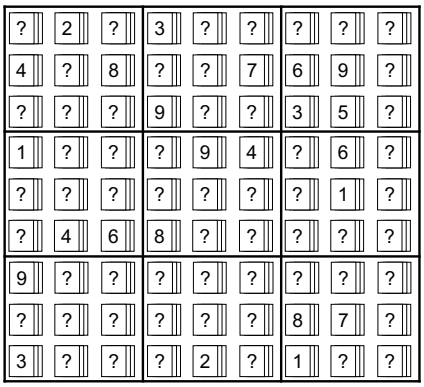


Figure 2: Input card placement.

1.1 Zero-Knowledge Proofs for Sudoku Assume two Sudoku lovers P and V, where P brought a challenging Sudoku puzzle to which P knew a solution, and P asked V to solve it. Then, V has spent much time and effort to solve the puzzle, but V cannot find a solution. Therefore, V wonders whether a solution really exists to the Sudoku puzzle presented by P. On the other hand, P does not want to reveal the solution. This is the right occasion for a *prover* P and a *verifier* V to make use of the *zero-knowledge proof* [4, 5] technique.

This paper considers constructing physical zero-knowledge proof protocols using a deck of cards, i.e., we focus on *card-based* zero-knowledge proof protocols for Sudoku.

1.2 The First Protocol In 2007, Gradwohl, Naor, Pinkas, and Rothblum [6,7] constructed the first card-based zero-knowledge proof protocol for Sudoku. Their protocol uses physical cards like

$$\boxed{1}\boxed{2}\boxed{3}\boxed{4}\boxed{5}\boxed{6}\boxed{7}\boxed{8}\boxed{9},$$

whose backs are all identical $\boxed{?}$.

The idea behind their protocol is as follows. Assume that only the prover P knows the solution to the puzzle shown in Figure 1. Considering the first row of the puzzle, P places nine cards according to the solution:

$$\underbrace{\boxed{?}}_{5}\boxed{2}\underbrace{\boxed{?}}_{9}\boxed{3}\underbrace{\boxed{?}}_{4}\underbrace{\boxed{?}}_{6}\underbrace{\boxed{?}}_{7}\underbrace{\boxed{?}}_{8}\underbrace{\boxed{?}}_{1}, \tag{1.1}$$

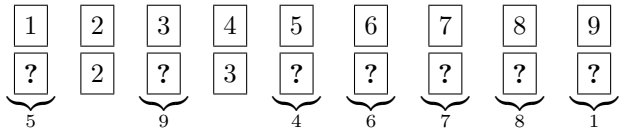
where the cards related to the solution are secretly placed face down on the empty cells by P (without the verifier V seeing the numbers) and the number attached to a face-down card means its value written on the face, for convenience sake. After turning over the face-up two cards on the hint cells, we apply a (complete) shuffle to the nine cards and turn them over. If every number from 1 to 9 appears, V is convinced that the sequence of face-down cards placed by P as the first row meets the rule of Sudoku.

Bearing this idea in mind, since we have to verify that every row, every column, and every (3×3)-block satisfy the rule of Sudoku, let P place three cards on every cell, as illustrated in Figure 2. After turning over all the face-up cards, for each of rows, columns, and blocks, we pick nine cards by randomly choosing one from the three cards on a cell, apply a shuffle to the picked nine cards, and turn them over to check that every number appears. Unfortunately, this process has a soundness error (because the three cards placed by P may not be identical). Therefore, the process needs to be repeated several times, i.e., ℓ times for a security parameter ℓ, to reduce the soundness error probability. Thus, the protocol requires 27ℓ shuffles.

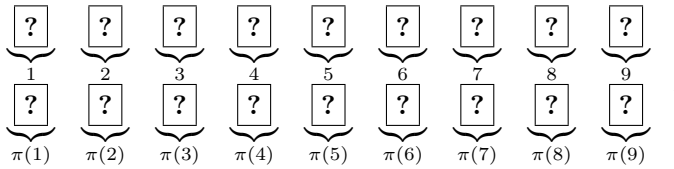
This is the seminal protocol by Gradwohl, Naor, Pinkas, and Rothblum. For an $n \times n$ Sudoku puzzle, the protocol requires $3n\ell$ shuffles.

1.3 Soundness-Error-Free Protocols In 2018, Sasaki, Mizuki, and Sone [33, 34] succeeded in completely removing the soundness error by devising the following technique.

Let us place nine cards $\boxed{1}$ through $\boxed{9}$ above the sequence (1.1):



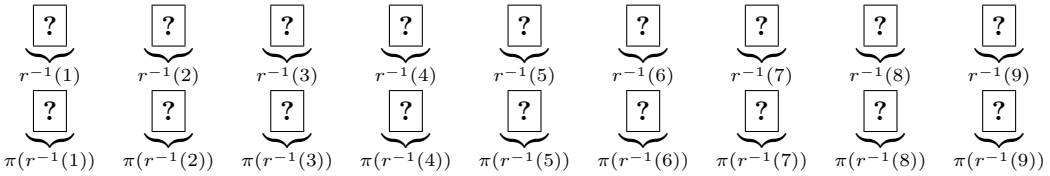
Turn over the face-up cards; then, we have



where π is a permutation such that

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 5 & 2 & 9 & 3 & 4 & 6 & 7 & 8 & 1 \end{pmatrix}.$$

By regarding each (vertical) pair of cards as a pile, apply a "pile-scramble shuffle" (the details of which will be shown in Section 2.2); then, the two sequences are permuted simultaneously according to the same random permutation $r \in S_9$, where S_m for a positive integer m denotes the symmetric group of degree m hereinafter:



Reveal all the cards of the second sequence to check that every number appears. After turn the second sequence face down and apply a pile-scramble shuffle again, we reveal the first sequence and sort the piles in ascending order based on the revealed numbers, so that we can revert the original sequence. In this way, we can verify that a sequence of nine face-down cards consists of all the numbers from 1 to 9 without leaking any information about its order, while keeping the original sequence intact.

By the virtue of this technique, we can use the same input card placement for verifying the columns and blocks as the rows, and hence, no soundness error arises. Since each row or each column needs two pile-scramble shuffles and each block needs one shuffle, the protocol requires 45 shuffles. For an $n \times n$ Sudoku puzzle, the protocol requires $5n$ shuffles. There are other variants (named Protocols B and C); see Tables 1 and 2.

1.4 Recent Protocols In 2021, Ruangwises [24, 25] proposed two protocols using standard decks of playing cards like

His protocols are very efficient in terms of the number of required decks which are commercially easily available, i.e., they require only two standard decks for a 9×9 Sudoku puzzle although they need a relatively large number of shuffles. See Tables 1 and 2 for the details of their performances[2].

In 2023, Tanaka and Mizuki [37] proposed a protocol for a 9×9 Sudoku puzzle using two UNO[3] decks consisting of cards like

[2]The number of shuffles of Protocol B can be reduced by two if n is an even number.
[3]UNO is a trademark or registered trademark of Mattel, Inc.

Table 1: Zero-knowledge proof protocols for 9×9 Sudoku

Protocols	Number of cards	Number of shuffles
Gradwohl et al. [7]	243	27ℓ (ℓ: sec. param.)
Sasaki et al. [33] (Protocol A)	90	45
Sasaki et al. [33] (Protocol B)	171	36
Sasaki et al. [33] (Protocol C)	243	28
Ruangwises [25] (Protocol A)	120	108
Ruangwises [25] (Protocol B)	108	322
Tanaka and Mizuki [37]	117	16
Section 3 (Three-shuffle protocol)	324	3
Section 4 (Two-shuffle protocol)	324	2

Table 2: Zero-knowledge proof protocols for $n \times n$ Sudoku

Protocols	Number of cards	Number of shuffles
Gradwohl et al. [7]	$3n^2$	$3n\ell$ (ℓ: sec. param.)
Sasaki et al. [33] (Protocol A)	$n^2 + n$	$5n$
Sasaki et al. [33] (Protocol B)	$2n^2 + n$	$4n$
Sasaki et al. [33] (Protocol C)	$3n^2$	$3n + 1$
Ruangwises [25] (Protocol A)	$n^2 + n(\sqrt{n} + 1) + \sqrt{n}$	$4n\sqrt{n}$
Ruangwises [25] (Protocol B)	$n^2 + 2n + 3\sqrt{n}$	$2n^2(\sqrt{n} - 1) + 2$
Tanaka and Mizuki [37]	$n^2 + n(\sqrt{n} + 1)$	$7\sqrt{n} - 5$
Section 3 (Three-shuffle protocol)	$4n^2$	3
Section 4 (Two-shuffle protocol)	$4n^2$	2

By making use of color information of cards, we can verify three rows (columns, blocks) simultaneously, leading to greatly reducing the number of required shuffles. Thus, the protocol uses only 16 shuffles for a 9×9 puzzle and $7\sqrt{n} - 5$ shuffles for an $n \times n$ puzzle. This is the state-of-the-art protocol in terms of the number of required shuffles. As for the number of cards, see Tables 1 and 2[4].

1.5 Our Contribution As explained in the previous subsection, the currently known upper bound on the number of required shuffles is $7\sqrt{n} - 5$. Can one further improve?

In this paper, we tackle this problem and show that only three or two shuffles are sufficient to construct a card-based zero-knowledge proof protocol for an $n \times n$ Sudoku puzzle, no matter how large n is. That is, for an $n \times n$ Sudoku puzzle, we first propose a protocol using only three shuffles, and then, we modify it to have a protocol using only two shuffles. Since the second protocol is a little bit elaborate to reduce the number of shuffles, our three-shuffle protocol is somewhat simpler than our two-shuffle protocol.

The main idea behind our proposed protocols is to attach three cards to the card on each cell, as illustrated

[4]The number of cards can be reduced by n if n is an even number.

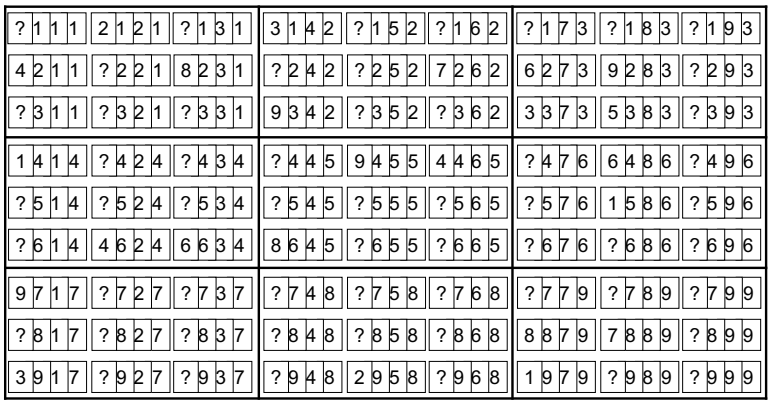


Figure 3: Input placement for our protocols.

in Figure 3. The attached three cards hold information about which row, column, and block the corresponding card is located at. More precisely, for a cell on the a-th row, b-th column, and c-th block, we prepare three cards $\boxed{a}\boxed{b}\boxed{c}$, and attach them in this order to the card on the cell. Thus, our protocols use $4n^2$ cards in total for an $n \times n$ puzzle. As will be seen later, thanks to the cards holding the row (column or block) information, we can verify all rows (columns or blocks) simultaneously using a single pile-scramble shuffle. Figure 4 shows a real card placement.

Figure 4: Real card placement.

Let us stress again that each of our proposed protocols requires a constant number of shuffles, regardless of n. See Tables 1 and 2 again. In addition, our proposed protocols are simple and efficient enough for people to execute for a 9×9 Sudoku puzzle in reality.

After giving preliminaries in Section 2, we present our three-shuffle protocol in Section 3 and our two-shuffle protocol in Section 4. This paper concludes in Section 5 with some discussion and an open problem.

1.6 Related Work Aside from Sudoku, many card-based zero-knowledge proof protocols have been constructed for other puzzles and games. Recently published examples are as follows: 15-Puzzle [36], ABC End View [3, 28], Ball Sort Puzzle [27], Five Cells [17, 31], Goishi Hiroi [28], Herugolf [17], Heyawake [22], Hitori [22], Kurodoko [23], Makaro [30], Meadows [31], Moon-or-Sun [8], Nonogram [26], Nurikabe [22], Nurimisaki [23],

Pancake Sorting [13], Shikaku [29], Suguru [21], Sukoro [32], Sumplete [9], Toichika [28], Topswops [14], and Usowan [17].

Many puzzles as above have more complex rules than Sudoku, and constructing card-based protocols for such a complex puzzle has led to the development of various techniques for card-based cryptography. On the other hand, Sudoku is the most popular puzzle in the world because of the simplicity of its rules. Therefore, card-based zero-knowledge proofs for Sudoku are a well-motivated educational tool for introducing cryptography to ordinary people, and also have a special presence in the field of card-based cryptography. In fact, card-based zero-knowledge proofs for Sudoku have been the most deeply researched in the field of card-based zero-knowledge proofs.

2 Preliminaries

This section introduces cards and actions used in card-based protocols along with the pile-scramble shuffle and the properties of zero-knowledge proof protocols. Hereinafter, for a positive integer m, $[m]$ denotes $\{1, 2, \ldots, m\}$.

2.1 Cards and Actions As already seen in Section 1, we use *numbered* cards: The face of every card has a number from 1 through n and their backs are all identical, illustrated as

face up: $\boxed{1}\boxed{2}\boxed{3}\cdots\boxed{n}$, face down: $\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}$.

As will be seen in Section 4, our second protocol uses cards with bars in addition to the cards listed above:

face up: $\boxed{\overline{1}}\boxed{\overline{2}}\boxed{\overline{3}}\cdots\boxed{\overline{n}}$, face down: $\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}$.

We refer to the former as *normal cards* and the latter as *overlined cards*.

To accommodate such cards, for example, from standard decks, we can choose ♣-suit cards $\boxed{\clubsuit A}\boxed{\clubsuit 2}\boxed{\clubsuit 3}\cdots$ as normal cards and ♡-suit cards $\boxed{\heartsuit A}\boxed{\heartsuit 2}\boxed{\heartsuit 3}\cdots$ as overlined cards. As another example, from UNO decks, we can choose blue cards $\boxed{\text{B1}}\boxed{\text{B2}}\boxed{\text{B3}}\cdots$ as normal cards and red cards $\boxed{\text{R1}}\boxed{\text{R2}}\boxed{\text{R3}}\cdots$ as overlined cards.

Our proposed protocols follow the standard computational model of card-based cryptographic protocols [11, 12, 18, 19], in which a protocol is formally defined as an abstract machine. In the formalization, there are three main actions: *permute*, *turn*, and *shuffle*. They are applied to a sequence of cards; below, we assume a sequence of m cards.

Permute. This action is denoted by (perm, π), where $\pi \in S_m$ is a permutation applied to the sequence of cards as follows:

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\cdots\overset{m}{\boxed{?}} \xrightarrow{(\mathsf{perm},\pi)} \overset{\pi^{-1}(1)}{\boxed{?}}\overset{\pi^{-1}(2)}{\boxed{?}}\cdots\overset{\pi^{-1}(m)}{\boxed{?}}.$$

Turn. This action is denoted by (turn, T), where $T \subseteq [m]$ is a set of indices, indicating that for every $t \in T$, the t-th card is turned over as follows:

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\cdots\overset{t\in T}{\boxed{?}}\cdots\overset{m}{\boxed{?}} \xrightarrow{(\mathsf{turn},T)} \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\cdots\overset{t\in T}{\boxed{1}}\cdots\overset{m}{\boxed{?}}.$$

Shuffle. This action is denoted by (shuf, Π), where $\Pi \subseteq S_m$ is a set of permutations, indicating that $\pi \in \Pi$ is drawn uniformly and is applied to the sequence of cards as follows:

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\cdots\overset{m}{\boxed{?}} \xrightarrow{(\mathsf{shuf},\Pi)} \overset{\pi^{-1}(1)}{\boxed{?}}\overset{\pi^{-1}(2)}{\boxed{?}}\cdots\overset{\pi^{-1}(m)}{\boxed{?}}.$$

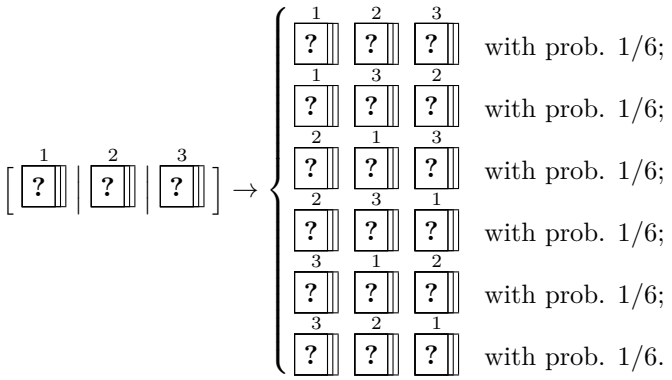
If Π is a subgroup of S_m, we call (shuf, Π) a *closed shuffle*; otherwise, we call it a *non-closed shuffle*. For example, the (perfect) shuffle applied to nine cards as seen in Section 1.2 can be written as (shuf, S_9), and it is a closed shuffle.

Each of our protocols, which will be presented in the succeeding sections, starts with an input sequence of $4n^2$ cards as illustrated in Figure 3, and applies a series of actions, such as (shuf, Π) for some Π and (turn, T) for some T.

Although a protocol should be defined with a combination of the actions above (or an abstract machine), hereinafter, we use a natural language to describe a protocol for simplicity.

2.2 Pile-Scramble Shuffle In this subsection, we introduce *pile-scramble shuffles* [10], which will be used to construct our protocols. A pile-scramble shuffle is a shuffling action for piles of cards (where the sizes of all piles are the same); after this shuffle, the piles are uniformly permuted while preserving the order of the cards in each pile.

For example, applying a pile-scramble shuffle to three piles, each consisting of three cards, results in the following six possibilities, each with a probability of 1/6:



Here, $[\cdot \mid \cdot \mid \cdot]$ denotes applying a pile-scramble shuffle to the sequence of cards. Hereinafter, we denote applying a pile scramble shuffle as $[\cdot \mid \cdots \mid \cdot]$. We note that, since the permutation set Π underlying a pile-scramble shuffle for k piles is isomorphic to the symmetric group S_k, it is a closed shuffle.

Pile-scramble shuffles are known to be easy to implement physically. To apply a pile-scramble shuffle to k piles, first prepare k identical envelopes, then put each pile in an envelope, and finally scramble these k envelopes sufficiently to obtain a uniformly random permutation over k envelopes.

2.3 Properties of Card-Based Zero-Knowledge Proof Same as (non-physical) zero-knowledge proofs, a card-based zero-knowledge proof protocol requires three properties: completeness, soundness, and zero-knowledge.

Completeness. If the input sequence corresponds to a solution of the puzzle, then the protocol always terminates with accepting the proof.

Soundness. If the input sequence does not correspond to a solution of the puzzle, then the protocol always terminates with rejecting the proof.

Zero-knowledge. There exists an algorithm called a simulator such that the distribution of the opened symbols when the protocol starting with an input sequence corresponding to a solution of the puzzle and the distribution of the output of the simulator are identically distributed.

As mentioned before, our protocols follow the standard computation model formalized by Mizuki and Shizuya [18]: Given an input sequence of cards placed by P, each of our protocols can be executed by either P or V, or even a third party. A more formal treatment for such card-based zero-knowledge proof protocols within the model appears in [16]. Although most of the existing protocols fall into this model, some protocols (e.g. Protocol C of [33] and Protocols A and B of [25]) use P's knowledge not only at the beginning but also during the protocol.

3 Three-Shuffle Protocol

In this section, we describe our *three-shuffle protocol*, which is a card-based zero-knowledge proof protocol for an $n \times n$ Sudoku puzzle using three shuffles. Our three-shuffle protocol uses three pile-scramble shuffles and $4n^2$ cards consisting of $4n$ sets of $\boxed{1}\boxed{2}\cdots\boxed{n}$.

3.1 Protocol Description At the beginning of the protocol, the prover P places four cards on each cell α as follows: Suppose that α is on the a-th row, b-th column, and c-th block. If α is a hint cell with number $x_\alpha \in [n]$, P places the following four cards:

$$\boxed{x_\alpha}\boxed{a}\boxed{b}\boxed{c}.$$

Otherwise, P places the following four cards:

$$\boxed{?}\boxed{a}\boxed{b}\boxed{c},$$

where the face-down card represents a solution $x_\alpha \in [n]$ of the cell. See Figure 3 again for an example of such a placement when $n = 9$.

Our protocol applies the *block verification*, *column verification*, and *row verification*, in this order. If an opened card in each verification differs from the protocol specification, the protocol terminates at that point and the proof is rejected. Otherwise, if all verification phases are passed, the proof is accepted.

3.1.1 Block Verification. In this phase, we verify that each block consists of the numbers 1 to n. The procedure is as follows.

1. Turn over all face-up cards as follows:

$$\boxed{x_\alpha}\boxed{a}\boxed{b}\boxed{c} \to \boxed{?}\boxed{?}\boxed{?}\boxed{?};$$
$$\boxed{?}\boxed{a'}\boxed{b'}\boxed{c'} \to \boxed{?}\boxed{?}\boxed{?}\boxed{?}.$$

2. Apply a pile-scramble shuffle to n^2 piles, where the four cards in each cell are regarded as a pile:

$$\left[\boxed{?}\boxed{?}\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\boxed{?}\boxed{?}\middle|\cdots\middle|\boxed{?}\boxed{?}\boxed{?}\boxed{?}\right].$$

3. Turn over the leftmost and rightmost cards in each pile. Let $x_i, c_i \in [n]$ be the opened values in the i-th pile ($i \in [n^2]$) as follows:

$$\boxed{?}\boxed{?}\boxed{?}\boxed{?} \to \boxed{x_i}\boxed{?}\boxed{?}\boxed{c_i}.$$

   If $\{x_i \mid c_i = c\}$ equals to $[n]$ for every $c \in [n]$, the protocol proceeds to the next phase. Otherwise, the protocol terminates and the proof is rejected.

3.1.2 Column Verification. In this phase, we verify that each column consists of the numbers 1 to n. The procedure is as follows.

1. Remove the rightmost card in each pile as follows:

$$\boxed{x_i}\boxed{?}\boxed{?}\boxed{c_i} \to \boxed{x_i}\boxed{?}\boxed{?}.$$

   (Note that this is not an actual operation in the abstract model. Removing cards can be done by just ignoring the cards in the subsequent steps.)

2. Turn over all face-up cards as follows:

$$\boxed{x_i}\boxed{?}\boxed{?} \to \boxed{?}\boxed{?}\boxed{?}.$$

3. Apply a pile-scramble shuffle to the n^2 piles, where each pile consists of three cards:

$$\left[\boxed{?}\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\boxed{?}\middle|\cdots\middle|\boxed{?}\boxed{?}\boxed{?}\right].$$

4. Turn over the leftmost and rightmost cards in each pile. Let $x'_i, b'_i \in [n]$ be the opened values in the i-th pile ($i \in [n^2]$) as follows:

$$\boxed{?}\boxed{?}\boxed{?} \to \boxed{x'_i}\boxed{?}\boxed{b'_i}.$$

   If $\{x'_i \mid b'_i = b\}$ equals to $[n]$ for every $b \in [n]$, the protocol proceeds to the next phase. Otherwise, the protocol terminates and the proof is rejected.

3.1.3 Row Verification. In this phase, we verify that each row consists of the numbers 1 to n. The procedure is as follows.

1. Remove the rightmost card in each pile as follows:

$$\boxed{x'_i}\boxed{?}\boxed{b'_i} \to \boxed{x'_i}\boxed{?}.$$

2. Turn over all face-up cards as follows:

$$\boxed{x'_i}\boxed{?} \to \boxed{?}\boxed{?}.$$

3. Apply a pile-scramble shuffle to the n^2 piles, where each pile consists of two cards:

$$\left[\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\middle|\cdots\middle|\boxed{?}\boxed{?}\right].$$

4. Turn over all cards. Let $x''_i, a''_i \in [n]$ be the opened values in the i-th pile ($i \in [n^2]$) as follows:

$$\boxed{?}\boxed{?} \to \boxed{x''_i}\boxed{a''_i}.$$

If $\{x''_i \mid a''_i = a\}$ equals to $[n]$ for every $a \in [n]$, the proof is accepted. Otherwise, the proof is rejected.

3.2 Properties of Our Protocol As seen in the previous subsection, our protocol uses three pile-scramble shuffles, each of which is done for n^2 piles. In this subsection, we show that our protocol satisfies the necessary properties.

3.2.1 Completeness. Suppose that an input sequence corresponds to a correct solution to a given Sudoku puzzle. For each block, the set of x_α for all cells α in the block forms $[n]$. Then we have $\{x_i \mid c_i = c\} = [n]$ for every $c \in [n]$, and thus the protocol proceeds to the block verification. Similarly, the column and row verifications are passed and the proof is accepted. Therefore, this protocol satisfies the completeness.

3.2.2 Soundness. Suppose that an input sequence does not correspond to a solution, i.e., there exists a block, row, or column that does not contain all numbers from 1 to n. If there exists such a block, say the c-th block, the set of x_α for all cells α in the c-th block does not match $[n]$. Then, in Step 3 of the block verification, $\{x_i \mid c_i = c\}$ does not match $[n]$, and thus the proof will be rejected. Similarly, if such an illegal row or column exists, the proof will be rejected for the same reason. Therefore, this protocol satisfies the soundness.

3.2.3 Zero-Knowledge. To show the zero-knowledge property, we need to construct a simulator that outputs the opened symbols indistinguishable to those in the real execution. For the block verification, the opened symbols can be simulated by a permuted sequence of $(1, ?, ?, 1), (1, ?, ?, 2), \ldots, (n, ?, ?, n)$ whose permutation is chosen uniformly at random. For the column verification, the opened symbols can be simulated by a permuted sequence of $(1, ?, 1), (1, ?, 2), \ldots, (n, ?, n)$ whose permutation is chosen uniformly at random. For the row verification, the opened symbols can be simulated by a permuted sequence of $(1, 1), (1, 2), \ldots, (n, n)$ whose permutation is chosen uniformly at random. They are exactly the same distributions as the opened symbols in the real execution due to the pile-scramble shuffles. Therefore, this protocol is zero-knowledge.

4 Two-Shuffle Protocol

In this section, we describe our *two-shuffle protocol*, which is a card-based zero-knowledge proof protocol for an $n \times n$ Sudoku puzzle using two shuffles. Our two-shuffle protocol uses two pile-scramble shuffles and $4n^2$ cards consisting of $3n$ sets of normal cards $\boxed{1}\boxed{2}\cdots\boxed{n}$ and n sets of overlined cards $\boxed{\overline{1}}\boxed{\overline{2}}\cdots\boxed{\overline{n}}$.

4.1 Protocol Description At the beginning of the protocol, the prover P places four cards on each cell α as follows: Suppose that α is on the a-th row, b-th column, and c-th block. If α is a hint cell with number $x_\alpha \in [n]$, P places the following four cards:

$$\boxed{x_\alpha}\boxed{a}\boxed{b}\boxed{\overline{c}},$$

where $\boxed{\overline{c}}$ is an overlined card. Otherwise, P places the following four cards:

$$\boxed{?}\boxed{a}\boxed{b}\boxed{\overline{c}},$$

where the left card is a normal card representing a solution $x_\alpha \in [n]$ of the cell. That is, the placement is exactly the same as in Figure 3 except that every rightmost card is an overlined card.

Our protocol first applies the *block verification*, and then applies *column and row verification*. If an opened card in each verification differs from the protocol specification, the protocol terminates at that point and the proof is rejected. Otherwise, if all verification phases are passed, the proof is accepted.

4.1.1 Block Verification. In this phase, we verify that each block consists of the numbers 1 to n. The procedure is as follows.

1. Turn over all face-up cards as follows:

$$\boxed{x_\alpha}\boxed{a}\boxed{b}\boxed{\overline{c}} \to \boxed{?}\boxed{?}\boxed{?}\boxed{?};$$
$$\boxed{?}\boxed{a'}\boxed{b'}\boxed{\overline{c'}} \to \boxed{?}\boxed{?}\boxed{?}\boxed{?}.$$

2. Apply a pile-scramble shuffle to n^2 piles, where the four cards in each cell are regarded as a pile:

$$\left[\boxed{?}\boxed{?}\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\boxed{?}\boxed{?}\middle|\cdots\middle|\boxed{?}\boxed{?}\boxed{?}\boxed{?}\right].$$

3. Turn over the leftmost and rightmost cards in each pile. For the i-th pile ($i \in [n^2]$), let $x_i, c_i \in [n]$ be the opened values as follows:

$$\boxed{?}\boxed{?}\boxed{?}\boxed{?} \to \boxed{x_i}\boxed{?}\boxed{?}\boxed{\overline{c_i}}.$$

   If $\{x_i \mid c_i = c\}$ equals to $[n]$ for every $c \in [n]$, the protocol proceeds to the next phase. Otherwise, the protocol terminates and the proof is rejected.

4.1.2 Column and Row Verification. In this phase, we verify that each row and each column consists of the numbers 1 to n. The procedure is as follows.

1. Remove the rightmost cards in each pile as follows:

$$\boxed{x_i}\boxed{?}\boxed{?}\boxed{\overline{c_i}} \to \boxed{x_i}\boxed{?}\boxed{?},$$

   then we obtain n sets of overlined cards $\boxed{\overline{1}}\boxed{\overline{2}}\cdots\boxed{\overline{n}}$. Then, for each pile containing a face-up card $\boxed{x_i}$, place the overlined card with the same number x_i to the right as follows:

$$\boxed{x_i}\boxed{?}\boxed{?} \to \boxed{x_i}\boxed{?}\boxed{?}\boxed{\overline{x_i}}.$$

   (Note that, in the abstract model, the above operations can be done by a permute operation instead of removing and placing.) In this way, we can make two copied cards of x_i, so that one card can be used for column verification and the other card can be used for row verification.

2. Turn over all face-up cards as follows:

$$\boxed{x_i}\boxed{?}\boxed{?}\boxed{\overline{x_i}} \to \boxed{?}\boxed{?}\ \boxed{?}\boxed{?}.$$

   Regard the four cards as two piles of two cards.

3. Apply a pile-scramble shuffle to $2n^2$ piles (two piles per cell) as follows:

$$\left[\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\middle|\cdots\middle|\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\right].$$

4. Turn over all cards. We refer to piles of two normal cards as *normal piles* and piles containing overlined cards as *overlined piles*. Let $x'_i, a'_i \in [n]$ be the opened values in the i-th normal pile ($i \in [n^2]$) and $b'_i, y'_i \in [n]$ be the opened values in the i-th overlined pile ($i \in [n^2]$) as follows:

$$\cdots \boxed{?}\boxed{?} \quad \boxed{?}\boxed{?} \cdots \to \cdots \boxed{x'_i}\boxed{a'_i} \quad \boxed{b'_i}\boxed{\overline{y'_i}} \cdots .$$

If $\{x'_i \mid a'_i = a\}$ equals to $[n]$ for every $a \in [n]$, and $\{y'_i \mid b'_i = b\}$ equals to $[n]$ for every $b \in [n]$, the proof is accepted. Otherwise, the proof is rejected.

4.2 Properties of Our Protocol As seen in the previous subsection, our protocol uses two pile-scramble shuffles, where the former one is done for n^2 piles and the latter one is done for $2n^2$ piles. In this subsection, we show that our protocol satisfies the necessary properties.

4.2.1 Completeness. Suppose that an input sequence corresponds to a correct solution to a given Sudoku puzzle. Then the block verification will pass for the same reason of our three-shuffle protocol. For the column and row verification, since the value x_i is copied to the overlined card in Step 2, the set of x_α for all cells α in the column or row forms $[n]$. Thus the column and row verification are passed and the proof is accepted. Therefore, this protocol satisfies the completeness.

4.2.2 Soundness. Suppose that an input sequence does not correspond to a solution. Then, similar to our three-shuffle protocol, the block verification or the column and row verification will be rejected because there exists a block, column, or row that does not contain all numbers from 1 to n. Therefore, this protocol satisfies the soundness.

4.2.3 Zero-Knowledge. To show the zero-knowledge property, we need to construct a simulator outputting the opened symbols indistinguishable to those in the real execution. For the block verification, the opened symbols can be simulated by a permuted sequence of $(1, ?, ?, \overline{1}), (1, ?, ?, \overline{2}), \ldots, (n, ?, ?, \overline{n})$ whose permutation is chosen uniformly at random. For the column and row verification, the opened symbols can be simulated by a permuted sequence of $(1, 1), (1, 2), \ldots, (n, n), (1, \overline{1}), (1, \overline{2}), \ldots, (n, \overline{n})$ whose permutation is chosen uniformly at random. They are exactly the same distributions as the opened symbols in the real execution due to the pile-scramble shuffles. Therefore, this protocol is zero-knowledge.

5 Conclusion

This paper presented novel card-based zero-knowledge proof protocols for Sudoku. Specifically, we proposed a three-shuffle protocol and a two-shuffle protocol, both of which work for any size of Sudoku, i.e., an $n \times n$ Sudoku puzzle. The main idea is to attach three cards to the card on each cell, such that the attached three cards have information about the positions in terms of rows, columns, or blocks.

Our protocols are simple and efficient enough to execute for a 9×9 Sudoku puzzle. Figure 5 illustrates a real execution of our three-shuffle protocol, where we used UNO cards as numbered cards and used a card sleeve to make a pile of cards (instead of an envelope). It took about 20 minutes to complete the protocol.

We believe that: (i) as a demonstration of the concept of zero-knowledge proof, our protocols are arguably simpler than previous ones and thus they will be useful; (ii) physical zero-knowledge proof protocols are important in various settings, such as arms control verification [1] and forensics (DNA profiling) [2], and hence, the ideas and techniques in this paper could be also useful.

Remember that our proposed protocols use only the pile-scramble shuffle as a shuffling action. An intriguing open problem is to determine whether there exists a card-based zero-knowledge proof protocol using only a single pile-scramble shuffle for an $n \times n$ Sudoku puzzle. It should be noted that if we admit many cards and a somewhat complicated shuffle, theoretically, there exists a protocol using a single shuffle, due to the followings: (i) any Sudoku puzzle instance can be converted into a SAT (satisfiability problem) instance (e.g. [15]) and (ii) the card-based garbled circuit technique [20, 35, 38] can securely evaluate any logical circuit using only a single (but somewhat complicated) shuffle.

Figure 5: Real execution of our protocol.

Acknowledgements

We thank the anonymous reviewers, whose comments have helped us improve the presentation of the paper. This work was supported in part by JSPS KAKENHI Grant Numbers JP21K17702, JP23H00479, and JP24K02938, and JST CREST Grant Number JPMJCR22M1.

References

[1] G. Alexander, B. Boaz, and R. J. Goldston, *A zero-knowledge protocol for nuclear warhead verification*, Nature, 510 (2014), pp. 497–502.

[2] B. Fisch, D. Freund, and M. Naor, *Physical zero-knowledge proofs of physical properties*, in Advances in Cryptology—CRYPTO 2014, J. A. Garay and R. Gennaro, eds., vol. 8617 of LNCS, Berlin, Heidelberg, 2014, Springer, pp. 313–336.

[3] T. Fukasawa and Y. Manabe, *Card-based zero-knowledge proof for the nearest neighbor property: Zero-knowledge proof of ABC end view*, in Security, Privacy, and Applied Cryptography Engineering, L. Batina, S. Picek, and M. Mondal, eds., vol. 13783 of LNCS, Cham, 2022, Springer, pp. 147–161.

[4] S. Goldwasser, S. Micali, and C. Rackoff, *The knowledge complexity of interactive proof-systems*, in Annual ACM Symposium on Theory of Computing, STOC'85, New York, 1985, ACM, pp. 291–304.

[5] S. Goldwasser, S. Micali, and C. Rackoff, *The knowledge complexity of interactive proof systems*, SIAM J. Comput., 18 (1989), pp. 186–208.

[6] R. Gradwohl, M. Naor, B. Pinkas, and G. N. Rothblum, *Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles*, in Fun with Algorithms, P. Crescenzi, G. Prencipe, and G. Pucci, eds., vol. 4475 of LNCS, Berlin, Heidelberg, 2007, Springer, pp. 166–182.
[7] ———, *Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles*, Theory of Computing Systems, 44 (2009), pp. 245–268.
[8] S. Hand, A. Koch, P. Lafourcade, D. Miyahara, and L. Robert, *Efficient card-based ZKP for single loop condition and its application to Moon-or-Sun*, New Gener. Comput., 42 (2024), pp. 479–496.
[9] K. Hatsugai, S. Ruangwises, K. Asano, and Y. Abe, *NP-completeness and physical zero-knowledge proofs for Sumplete, a puzzle generated by ChatGPT*, New Gener. Comput., 42 (2024), pp. 429–448.
[10] R. Ishikawa, E. Chida, and T. Mizuki, *Efficient card-based protocols for generating a hidden random permutation without fixed points*, in Unconventional Computation and Natural Computation, C. S. Calude and M. J. Dinneen, eds., vol. 9252 of LNCS, Cham, 2015, Springer, pp. 215–226.
[11] 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*, in Advances in Cryptology—ASIACRYPT 2017, T. Takagi and T. Peyrin, eds., vol. 10626 of LNCS, Cham, 2017, Springer, pp. 126–155.
[12] A. Koch, S. Walzer, and K. Härtel, *Card-based cryptographic protocols using a minimal number of cards*, in Advances in Cryptology—ASIACRYPT 2015, T. Iwata and J. H. Cheon, eds., vol. 9452 of LNCS, Berlin, Heidelberg, 2015, Springer, pp. 783–807.
[13] Y. Komano and T. Mizuki, *Card-based zero-knowledge proof protocol for pancake sorting*, in Innovative Security Solutions for Information Technology and Communications, G. Bella, M. Doinea, and H. Janicke, eds., vol. 13809 of LNCS, Cham, 2023, Springer, pp. 222–239.
[14] ———, *Physical zero-knowledge proof protocols for Topswops and Botdrops*, New Gener. Comput., 42 (2024), pp. 399–428.
[15] I. Lynce and J. Ouaknine, *Sudoku as a SAT problem*, in 9th International Symposium on Artificial Intelligence and Mathematics, 2006, pp. 1–9.
[16] D. Miyahara, H. Haneda, and T. Mizuki, *Card-based zero-knowledge proof protocols for graph problems and their computational model*, in Provable and Practical Security, Q. Huang and Y. Yu, eds., vol. 13059 of LNCS, Cham, 2021, Springer, pp. 136–152.
[17] D. Miyahara, L. Robert, P. Lafourcade, and T. Mizuki, *ZKP protocols for Usowan, Herugolf, and Five Cells*, Tsinghua Science and Technology, 29 (2024), pp. 1651–1666.
[18] T. Mizuki and H. Shizuya, *A formalization of card-based cryptographic protocols via abstract machine*, Int. J. Inf. Secur., 13 (2014), pp. 15–23.
[19] ———, *Computational model of card-based cryptographic protocols and its applications*, IEICE Trans. Fundam., E100.A (2017), pp. 3–11.
[20] T. Ono, K. Shinagawa, T. Nakai, Y. Watanabe, and M. Iwamoto, *Single-shuffle card-based protocols with six cards per gate*, in Information Security and Cryptology, H. Seo and S. Kim, eds., vol. 14562 of LNCS, Singapore, 2024, Springer, pp. 157–169.
[21] L. Robert, D. Miyahara, P. Lafourcade, L. Libralesso, and T. Mizuki, *Physical zero-knowledge proof and NP-completeness proof of Suguru puzzle*, Inf. Comput., 285 (2022), p. 104858.
[22] L. Robert, D. Miyahara, P. Lafourcade, and T. Mizuki, *Card-based ZKP for connectivity: Applications to Nurikabe, Hitori, and Heyawake*, New Gener. Comput., 40 (2022), pp. 149–171.
[23] ———, *Physical ZKP protocols for Nurimisaki and Kurodoko*, Theor. Comput. Sci., 972 (2023), p. 114071.
[24] S. Ruangwises, *Two standard decks of playing cards are sufficient for a ZKP for Sudoku*, in Computing and Combinatorics, C.-Y. Chen, W.-K. Hon, L.-J. Hung, and C.-W. Lee, eds., vol. 13025 of LNCS, Cham, 2021, Springer, pp. 631–642.
[25] ———, *Two standard decks of playing cards are sufficient for a ZKP for Sudoku*, New Gener. Comput., 40 (2022), pp. 49–65.
[26] ———, *An improved physical ZKP for Nonogram and Nonogram color*, J. Comb. Optim., 45 (2023), p. 122.
[27] ———, *Physical zero-knowledge proof for ball sort puzzle*, in Unity of Logic and Computation, G. D. Vedova, B. Dundua, S. Lempp, and F. Manea, eds., vol. 13967 of LNCS, Cham, 2023, Springer, pp. 246–257.
[28] ———, *Verifying the first nonzero term: Physical ZKPs for ABC End View, Goishi Hiroi, and Toichika*, Journal of Combinatorial Optimization, 47 (2024), p. 69.
[29] S. Ruangwises and T. Itoh, *How to physically verify a rectangle in a grid: A physical ZKP for Shikaku*, in Fun with Algorithms, P. Fraigniaud and Y. Uno, eds., vol. 226 of LIPIcs, Dagstuhl, 2022, Schloss Dagstuhl, pp. 24:1–24:12.
[30] ———, *Physical ZKP for Makaro using a standard deck of cards*, in Theory and Applications of Models of Computation, D.-Z. Du, D. Du, C. Wu, and D. Xu, eds., vol. 13571 of LNCS, Cham, 2022, Springer, pp. 43–54.
[31] S. Ruangwises and M. Iwamoto, *Printing protocol: Physical ZKPs for decomposition puzzles*, New Gener. Comput., 42 (2024), pp. 331–343.

[32] S. Sasaki and K. Shinagawa, *Physical zero-knowledge proof for Sukoro*, New Gener. Comput., 42 (2024), pp. 381–398.
[33] T. Sasaki, D. Miyahara, T. Mizuki, and H. Sone, *Efficient card-based zero-knowledge proof for Sudoku*, Theor. Comput. Sci., 839 (2020), pp. 135–142.
[34] T. Sasaki, T. Mizuki, and H. Sone, *Card-based zero-knowledge proof for Sudoku*, in Fun with Algorithms, H. Ito, S. Leonardi, L. Pagli, and G. Prencipe, eds., vol. 100 of LIPIcs, Dagstuhl, Germany, 2018, Schloss Dagstuhl, pp. 29:1–29:10.
[35] K. Shinagawa and K. Nuida, *A single shuffle is enough for secure card-based computation of any Boolean circuit*, Discrete Applied Mathematics, 289 (2021), pp. 248–261.
[36] Y. Tamura, A. Suzuki, and T. Mizuki, *Card-based zero-knowledge proof protocols for the 15-puzzle and the token swapping problem*, in ACM ASIA Public-Key Cryptography Workshop, New York, 2024, ACM, pp. 11–22.
[37] K. Tanaka and T. Mizuki, *Two uno decks efficiently perform zero-knowledge proof for Sudoku*, in Fundamentals of Computation Theory, H. Fernau and K. Jansen, eds., vol. 14292 of LNCS, Cham, 2023, Springer, pp. 406–420.
[38] K. Tozawa, H. Morita, and T. Mizuki, *Single-shuffle card-based protocol with eight cards per gate*, in Unconventional Computation and Natural Computation, D. Genova and J. Kari, eds., vol. 14003 of LNCS, Cham, 2023, Springer, pp. 171–185.