Balance-Based ZKP Protocols for Pencil-and-Paper Puzzles

Shohei Kaneko, Pascal Lafourcade, Lola-Baie Mallordy, Daiki Miyahara, Maxime Puys, Kazuo Sakiyama

Balance-Based ZKP Protocols for Pencil-and-Paper Puzzles

No Author Given

No Institute Given

Abstract. In this paper, we propose zero-knowledge proof (ZKP) protocols using physical objects for four pencil-and-paper puzzles: the well-known Sudoku as well as Makaro, Futoshiki, and Kakuro. That is, our protocols allow a prover to convince a verifier that the prover knows a solution to a puzzle without relying on the use of computers. While previous physical ZKP protocols for puzzles have mainly relied on decks of cards, our research introduces a novel approach utilizing a balance scale and coins to design balance-based ZKP protocols; moreover we show its flexibility by adapting it to the four different puzzles. We compare the number of coins and operations in our protocols with the existing card-based protocols and show that, for certain puzzles, our balance-based protocol outperforms the card-based method. Finally, we prove that our protocols achieve perfect completeness, perfect soundness and are perfectly zero-knowledge.

Keywords: Cryptology · Zero-Knowledge Proof · Balance Scale · Pencil-and-Paper Puzzle · Sudoku.

1 Introduction

Alice, a hungry girl, goes to a fish market in the East Coast of the US with no money as depicted in Fig. 1. She spots a stand selling fish, with a big sign claiming *"Free fish for anyone who can solve my four puzzles"*. She comes closer and sees that the puzzles are four pencil games: Sudoku, Makaro, Futoshiki, and Kakuro. She cannot miss such a golden opportunity, and starts searching for the solutions. After several hours racking her brain without finding any solution, she screams at the merchant: *"Grifter, your puzzles are impossible!"*. The merchant calmly tells her *"I can prove I know all of the solutions"*. The merchant cannot give away its solutions, or people would come flocking to its stand asking for free fish. What he needs to do is a *zero-knowledge proof* (ZKP) to Alice, allowing to convince her that he knows a solution without revealing it. He remembers that Murata et al. [22] proposed similar protocols using a PEZ dispenser. However, there are no PEZ dispensers at the fish market; the merchant only has a *balance* and *coins* on his stand, to weight the fishes he sells. In this research, we propose a method to assist merchants, designing ZKP protocols using a balance and coins.

1.1 Zero-Knowledge Proof

Zero-knowledge proofs (ZKPs), introduced in 1985 by Goldwasser et al. [9], allow a prover P to convince a verifier V that a given statement is true without revealing any further information. ZKPs give a model that is not limited to computer use, but may also be applied in real life using everyday objects. In 1990, Quisquater et al. [23] published the well-known story of the Ali Baba cave to illustrate this concept, which made the first instance of a physical ZKP.

A ZKP protocol for a solution to a pencil-and-paper puzzle should satisfy three properties as follows:

Completeness: If P knows a solution of a given grid, it can convince V.
Soundness: If P does not provide a correct solution of a given grid, V rejects P with a sufficiently high probability.
Zero-knowledge: The verifier V is not given any information other than that the prover P can solve the puzzle.

These properties can come in three different flavours: perfect, statistical and computational. Perfect completeness means that an honest prover will always convince an honest verifier on a true statement, perfect soundness means that it is impossible to prove a false statement, and perfect zero-knowledge means that transcripts can be perfectly simulated and leak no information whatsoever. Perfect soundness can be relaxed to statistical soundness, where a prover must have a negligible probability of falsely convincing the verifier. It can be relaxed further to computational soundness, where any way to cheat must be computationally infeasible. Completeness and zero-knowledge can be relaxed in the same way. Our proposed protocols achieve the stronger versions of these properties: perfect completeness, perfect soundness, and perfect zero-knowledge based on some physical assumptions.

It was shown that for any NP-complete problem, there exists an interactive ZKP [8]. An extension by Ben-Or et al. [3] showed that every provable statement can be proved in zero-knowledge. The puzzles introduced in this paper have all been proven to be NP-complete: Sudoku and Kakuro in 2003 [33], Makaro in 2018 [14], and Futoshiki in 2021 [16]. Thus, there should exist ZKP protocols for such puzzles; however, a concrete procedure using a balance has not been addressed.

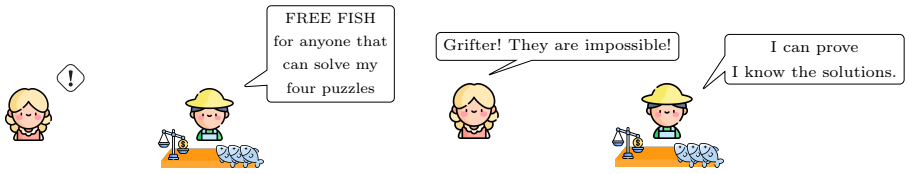

Fig. 1: Alice visits the fish market.

Table 1: Comparison of the complexity of our balance-based protocols with the existing card-based ones. We consider $n \times n$ grids except for the standard 9×9 Sudoku grid. In Futoshiki, i is the number of inequality symbols. In Makaro, n_s represents the number of rooms with size s, $2 \leq s \leq s_0$ for some s_0, c is the number of arrow cells, d is the number of cells adjacent to arrow cells, and e is the number of bold lines between two adjacent cells in different rooms. In Kakuro, t represents the number of triangular cells, n_h is the number of uninterrupted rows and columns of length h, $2 \leq h \leq h_0 < n$ for some h_0, and w is the number of white cells.

	Balance-Based			Card-Based	
	Coins	Shuffles	Comparisons	Cards	Shuffles
Sudoku [29]	90	2376	1215	90	45
Futoshiki	$n^2 + n$	$4n(\sum_{k=2}^{n} k) + 1$	$2n(\sum_{k=2}^{n} k) + i$	–	–
Makaro [5]	$\sum_{k=1}^{s_0} n_k k^2 + 2e$	$2\sum_{s=2}^{s_0}(n_s \times \sum_{k=2}^{s} k) + e$	$\sum_{s=2}^{s_0}(n_s \times \sum_{k=1}^{s} k) + d - c + e$	$2s_0 - 1 + s_0 \sum_{s=2}^{s_0} n_s) + (s_0 - 1)s_0$	$2(\sum_{s=2}^{s_0} n_s + c + e)$
Kakuro [4]	$t + 2w$	t	$t + \sum_{\ell=2}^{\ell_0}(n_\ell \binom{\ell}{2})$	$81(t + 81)$	$3t + 1$

1.2 Contributions

We propose a new perspective on ZKPs for pencil puzzles, replacing decks of cards with a balance and coins. To prove our method's adaptability, we show it can be applied to four different puzzles. We develop ZKP protocols for Sudoku, Makaro, Futoshiki, and Kakuro, which all provide perfect completeness, perfect soundness, and perfect zero-knowledge.

Table 1 indicates the number of coins, shuffles, and comparisons used in our balance-based protocol, as well as the number of cards and shuffles used in the existing card-based protocols. In Kakuro, it can be observed that the number of coins used in the balance-based protocol is less than the number of cards used in the card-based protocol [19]. In Futoshiki, our balance-based protocol directly verifies an inequality using the property of a balance, although there is no card-based protocol yet. From these observations, it can be inferred that in certain puzzles, balance-based protocols may reduce the number of physical entities and rounds of operation, making them easier to execute compared to card-based protocols. We note that a type of shuffle used in card-based ZKP protocols is costly to implement, while our balance-based protocol uses a common and easy-to-implement shuffle.

1.3 Related Work

In 2009, the first physical ZKP applied to Sudoku was proposed [10], using a deck of cards. This leads to several improvement results [27, 29, 31]. In addition to Sudoku, there are many card-based ZKP proposed, such as Nurimisaki [26],

Takuzu [18], Juosan [18], Usowan [25], Nurikabe [24], Hitori [24], Heyawake [24], Makaro [5,27], Kurodoko [26], Akari [4], Kakuro [4,19], KenKen [4], Sumplete [12], Ripple Effect [28], and Moon-or-Sun [11]. However, few solutions try to incorporate other everyday objects other than cards. Such examples include: PEZ dispenser [1, 2], coins (where their weights are not considered, but rather the toss of coins with result either head or tail) [15], polarizing plates [30], dial lock [20], tamper-evident seals [21], balls in bags [17], and marbles in an auction protocol [6]. These protocols are not for ZKPs, but for secure multiparty computations, which enable us to compute a given function over private inputs without revealing anything. Our study proposes a ZKP protocol using a balance and coins, which to our knowledge is the first of its kind. That is, our study explores computations performed by comparisons. It should be noted that a ZKP protocol for Sudoku can be constructed by comparisons. As for Futoshiki, we are the first ones to propose a physical ZKP protocol.

The computational complexity of pencil-and-paper puzzles has been widely studied [13], and a large number of puzzles are proved to be NP-complete as shown in a recent survey [32]. To the best of our knowledge, no research has employed an NP-hardness proof to construct a physical ZKP protocol because such an NP-hardness proof is shown by a reduction (mostly from the SAT problem), while a physical ZKP protocol is constructed directly to eliminate overhead.

2 Model

To describe our ZKP protocols visually, we introduce our use of coins and balances. We use balances similar to the Roberval balance, which is depicted in Fig. 2. We assume an ideal balance that tilts at a constant angle toward the heavier side regardless of the weight difference. That is, from seeing how the balance tilts, we cannot obtain information on the weights of coins placed on its plates other than which is heavier.

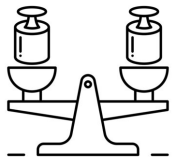

Fig. 2: The Roberval balance.

Our protocols use a coin represented as ◯ of various weights (their weights are indistinguishable by just looking at them). We use two types of coins, marked a and b on its face, represented as ⓐ and ⓑ, respectively. Their back remains unmarked and is represented as ◯; on this side the coins are indistinguishable. We sometimes denote a stack of several coins by ◯◯◯ for simplicity. Essentially, the coin ⓐ will serve as the prover's commitment to its solution, and coin ⓑ will serve as the verifier's challenge. In addition to moving coins, our protocols make use of three specific operations on coins: *Compare*, *Flip*, and *Shuffle* as well as one subprotocol: *Return Protocol*.

Compare: We represent a comparison of two stacks of coins using a balance as follows: ◯◯◯ | ◯◯◯. This operation returns three results: *Left*, *Right*, and *Even*, referring to the heavier stack between the two stacks, and no more information.

Flip: Flipping a coin is represented as follows: $\textcircled{a} \to \bigcirc$ or $\bigcirc \to \textcircled{a}$.

Shuffle: Shuffling several coins is represented as follows: $[\bigcirc_1 \bigcirc_2 \cdots \bigcirc_m] \to [\bigcirc_{r(1)} \bigcirc_{r(2)} \cdots \bigcirc_{r(m)}]$, where r is a uniformly distributed random permutation chosen from the symmetric group of degree m. This operation returns the coins rearranged completely random: after shuffling, the order of m coins is rearranged according to r (the underscripts number are given to identify the new positions, however the coins are indistinguishable in practice).

Return Protocol: This subprotocol takes m face-down coins $\bigcirc$s as input and returns k face-up coins $\textcircled{a}$s and ℓ face-up coins $\textcircled{b}$s such that $k+\ell = m$. For this, we first apply a shuffle to the m coins: $[\,\bigcirc\!\!\bigcirc\!\!\bigcirc\,]$. Then, by flipping the m coins, we check the marks on the coins and return k $\textcircled{a}$s and ℓ $\textcircled{b}$s. Note that they should appear in a randomized order when flipping them due to the application of shuffling.

In a balance-based ZKP protocol, a prover P and a verifier V monitor each other so that they do not deviate from protocol's description. Note that a magician is the outside of the model. Its efficiency is evaluated by the number of coins used (as memory) and the number of shuffles and comparisons performed (as CPU).

3 Sudoku

Sudoku is a famous puzzle, which gained popularity in 1986 when it was published by the Japanese puzzle company, Nikoli[1]. In this game, any number from 1 to 9 is placed in an empty cell. A typical Sudoku grid is a 9×9 grid, divided into 3×3 blocks. Initially, some cells are filled with numbers. In Fig. 3, we give a simple example of a Sudoku grid and its solution. The goal is to fill the cells so that each row (there are 9 rows), each column (there are 9 columns), and each block (there are 9 blocks) contains distinct numbers from 1 to 9.

We present a ZKP of knowledge of Sudoku using a balance and coins. The objective is for the prover P to convince the verifier V that it possesses a solution for a given Sudoku grid without revealing anything to V on its solution. Our protocol uses 81 $\textcircled{a}$s, with nine coins for each weight in $\{1, \ldots, 9\}$ grams. It also uses nine $\textcircled{b}$s, with one coin for each weight in $\{1, \ldots, 9\}$ grams. Its security proof is given in Appx. A.

Setup Phase: The nine $\textcircled{b}$s are placed aligned next to the Sudoku grid with their faces down and are shuffled: $[\,\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\,] \to \bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc\!\!\bigcirc$. Let us call these coins the *challenge*. According to P's solution, the prover P places the one $\textcircled{a}$ on each cell with its face down so that the weight of each coin placed on a cell is equal to the number filling the cell. More precisely, the coins are placed in two phases:

1. For each initially filled cell, V places a face-down $\textcircled{a}$ with the corresponding weight.
2. For each empty cell, P places a face-down $\textcircled{a}$ according to its solution.

[1] https://www.nikoli.co.jp/en/

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

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

Fig. 3: An example of a Sudoku puzzle and its solution introduced in the Nikoli's website: https://www.nikoli.co.jp/en/puzzles/sudoku/.

Verification Phase: The prover and the verifier execute the following operations for each row (resp. column or block), to verify that the nine coins placed in the row (resp. column or block) are identical to the challenge, *i.e.*, the numbers 1 through 9 each appear only once.

1. P picks the coin ◯ placed on any cell of the row (resp. column or block) to be checked. Let us call this coin the *commitment*.
2. In the following, V verifies that only a single ◯ among the challenge has the same weight as the commitment ◯. For this, it proceeds as follows, for each coin ◯ of the challenge.
   (a) P shuffles the coin of the challenge and the commitment: [◯◯].
   (b) P compares the two coins: ◯ | ◯.
   - If the comparison results in even, then P flips the two coins and places the commitment ⓐ on the cell but removes the coin ⓑ from the challenge. The verification goes to step 4.
   - Otherwise, P applies the return protocol to the two coins to obtain the commitment ⓐ and the ⓑ. The coin ⓑ remains in the challenge.
3. If none of the above comparisons result in even, V rejects P's solution.
4. P and V repeat the above steps for each of the remaining cells of the row (resp. column or block).
5. P returns the nine coins ⓑs removed to their original positions, *i.e.*, next to the grid.

In this way, the verifier is convinced that each row (resp. column or block) contains distinct numbers from 1 to 9 because for each cell, exactly one comparison results in even between the commitment and challenge, and the coin resulting in even is removed from the challenge. Note that P is the only one manipulating the coins, otherwise V could learn information on their weights when manipulating them.

Efficiency: This protocol uses 90 coins. When verifying each cell of a given row, we compare the challenge of nine ◯s and a commitment of one ◯ at first. Subsequently, the number of coins for the challenge decreases by one for each verification of a cell. Thus, in the worst case, we need 45 ($= \sum_{k=1}^{9} k$) comparisons.

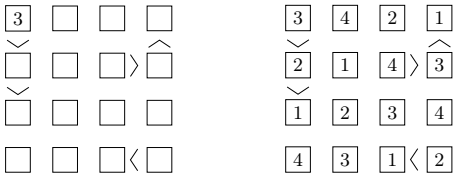

Fig. 4: An example of a Futoshiki puzzle and its solution.

Because there are nine rows/columns/blocks, the total number of comparisons becomes 1215 ($= 45 \times 9 \times 3$). As for the number of shuffles, because shuffles are applied just before and after each comparison, the number of shuffles will be twice the number of comparisons, totaling 2430. However, when verifying the last cell of a given row, we do not apply a shuffle just before and after the comparison because the comparison should result in even. Hence, the total number of shuffles is 2377 ($= 2430 - 2 \times 9 \times 3 + 1$), considering a shuffle applied in the setup phase. For an $n \times n$ Sudoku grid, this protocol uses $n^2 + n$ coins and requires $6n(\sum_{k=2}^{n} k) + 1$ shuffles and $3n \sum_{k=1}^{n} k$ comparisons. Table 1 shows the case for $n = 9$.

4 Futoshiki

Futoshiki is a puzzle developed by Tamaki Seto in 2001, played on an $n \times n$ square grid. A Futoshiki grid includes white cells and inequality signs. In Fig. 4, we give a example of a 4×4 Futoshiki grid and its solution. The goal is to place one number in every white cells on the board according to the following constraints:

1. Each row and each column contains all the numbers 1 through n.
2. The numbers must satisfy the inequality signs.

The main difference from Sudoku is that the numbers must also satisfy the inequality rule. We detail our protocol for an $n \times n$ grid. Our protocol achieves perfect completeness, perfect soundness and is perfectly zero-knowledge. The proofs are given in Appx. B.

Our protocol uses n^2 ⓐs with n coins for each weight in $\{1, \ldots, n\}$ grams, and n ⓑs with one coin for each weight in $\{1, \ldots, n\}$ grams.

Setup: The setup phase is exactly the same as in our proposed protocol for Sudoku (Sect. 3). That is, the n face-down coins ⓑs are shuffled and placed aligned next to the grid. According to P's solution, the prover P places a face-down ⓐ on every cell.

Verification: P and V execute the following steps:

1. To verify that each row and column contains all the numbers from 1 to n, P and V use the same method as for Sudoku (Sect. 3).

2. To verify that the numbers on both sides of each inequality sign satisfy the rule, P compares two ◯s placed on both sides of the sign: ◯ | ◯. V observes that the balance gives the expected result; if not, V rejects P's solution. After performing each comparison, P moves the ◯s to their original positions.

Efficiency: Let i denote the number of inequality sings in a given $n \times n$ grid. This protocol uses n^2+n coins and performs $4n(\sum_{k=2}^{n} k)+1$ shuffles and $2n(\sum_{k=2}^{n} k)+i$ comparisons. Compared to Sudoku, the number of comparisons is reduced by $\sum_{k=2}^{n} k$ due to the absence of blocks, but it increases by i for the inequality verification. The shuffles are also reduced by $2n\sum_{k=2}^{n} k$ compared to Sudoku due to the absence of blocks. In the inequality verification, because no shuffling is performed, it does not impact the total number of shuffle operations.

5 Makaro

Makaro is another grid game proposed by Nikoli. A Makaro grid is made of white cells, and black cells filled with an arrow. In Fig. 5, we give an example of a 5×5 Makaro grid. The goal is to place one number in every white cells on the grid according to the following constraints:

1. The areas separated by bold lines are called rooms, and each room is filled with one number from 1 to the number of cells in that room.
2. In the case of a black cell with an arrow, the cell to which the arrow points must be the cell with the highest number out of the vertically and horizontally adjacent cells to that black cell.
3. Adjacent cells cannot have the same number.

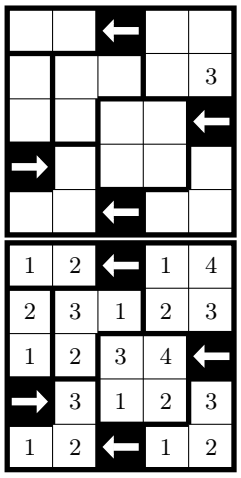

Fig. 5: An example of a Makaro puzzle and its solution.

The main difference from Sudoku is that the cells must be filled according to the arrow rule, *i.e.*, the number pointed by the arrow must be the highest among the adjacent cells. This property is easy to verify using a balance. Our protocol achieves perfect completeness, perfect soundness and is perfectly zero-knowledge. The proofs are given in Appx. C.

Setup: As in the protocol for Sudoku (Sect. 3), the challenge of face-down coins ⓑs are shuffled and placed aligned next to the grid. According to P's solution, the prover P places the commitments of face-down coins ⓐs (not a single) of the corresponding weight on every cell. In the following, we assume that the number of coins placed is sufficient for clarity. The correct number of coins is computed later.

Verification: The prover P and the verifier V execute as follows:

1. To verify that there are coins of distinct weights for each room, P and V use the same method as for Sudoku using the challenge (Sect. 3).
2. To verify that a ◯ on each cell pointed by an arrow is the heaviest, P compares it with each other adjacent ◯ around the arrow: ◯ | ◯. If the coin pointed by the arrow is ever found lighter than another, V rejects P's solution. The coins are moved to their original positions after each comparison.
3. To verify that no identical coins are next to each other, for each of such a pair of coins, P shuffles the two ◯s: [◯◯], and compares them: ◯ | ◯. If the balance shows even, V rejects P's solution. The two coins are no longer used and are removed.

Efficiency: Let n_s denote the number of rooms with size s, $2 \leq s \leq s_0$ for some s_0, c the number of arrows, d the number of cells adjacent to arrow cells, and e the number of bold lines between two adjacent cells in different rooms. This protocol uses $\sum_{s=2}^{s_0} n_s + s_0 + 2e$ coins and performs $2\sum_{s=2}^{s_0}(n_s \sum_{k=2}^{s} k) + e$ shuffles and $\sum_{s=2}^{s_0}(n_s \sum_{k=1}^{s} k) + d - c + e$ comparisons. The number of comparisons for the room verification follows the same approach to Sudoku and is $\sum_{s=2}^{s_0}(n_s \sum_{k=1}^{s} k)$. In the arrow verification, the number of comparisons is $d-e$ because it performs comparison for the cells indicated by the arrows with the other cells. In the arrow verification, because no shuffling is performed, it does not impact the total number of shuffles.

6 Kakuro

Kakuro (or *Kakkuro*) was the most popular logic puzzle in Japanese printed press until 1992, when Sudoku took the top spot. The Kakuro grid has white cells and gray cells separated by diagonal lines into two triangular rooms. In Fig. 6, we give a example of a 6×6 Kakuro grid and its solution. The goal is to place one number in every white cells on the grid according to the following constraints:

1. The number in the upper right corner of the oblique line represents the sum of the numbers entering the consecutive white cells to its right.
2. The number in the lower left corner of the oblique line represents the sum of the numbers entering the consecutive white cells below it.
3. Each connected (*i.e.,* uninterrupted by a gray cell) row or column cannot contain twice the same number.

By using a balance, it is easy to compare the numbers in the cells separated by diagonal lines with the sum of the numbers in the continuously connected cells. We detail our ZKP protocol. It achieves perfect completeness, perfect soundness and is perfectly zero-knowledge. The proofs are given in Appendix D.

Setup: P and V fill the grid in two phases:

1. For each triangular cell, V places a coin ⓑ of the indicated weight.
2. According to its solution, P places two coins ⓐs on each white cell.

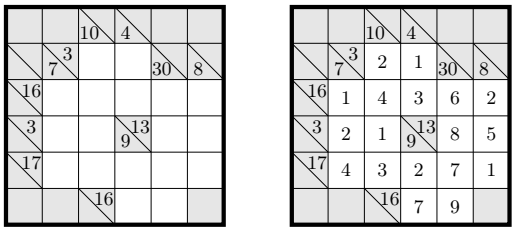

Fig. 6: An example of a Kakuro puzzle and its solution.

Verification: P and V execute the following steps:

1. For each triangular cell with a number, to verify if the weight of the coin on the cell is equal to the sum of the weights of the coins on consecutive white cells from the triangle cell, P and V follow these steps:
   - P compares the $\textcircled{b}$ representing the number on the triangular cell with the $\textcircled{a}$s on the consecutive white cells: $\textcircled{a}\!\!\textcircled{a}\!\!\textcircled{a} \mid \textcircled{b}$.
   - If the comparison does not result in even, then V rejects P's solution.
   - P moves the $\textcircled{a}$s to their original positions.[2]
2. For each uninterrupted row (resp. column), V verifies that a coin placed on each cell is of different weight to the ones placed on the other cells, as follows:
   - P picks one $\textcircled{a}$ placed on each cell and shuffles them: $[\,\textcircled{a}\!\!\textcircled{a}\!\!\textcircled{a}\,]$.
   - For each of all possible pairs of the coins, P compares them: $\textcircled{a} \mid \textcircled{a}$.
   - If even one of the above comparisons results in even, then V rejects P's solution.

Efficiency: Let t denote the number of triangular cells, n_ℓ the number of uninterrupted rows and columns of length ℓ, $2 \leq \ell \leq \ell_0$ for some ℓ_0, and w the number of white cells. This protocol uses a total of $t + 2w$ coins and performs t shuffles because in step 2, it applies a shuffle for each of uninterrupted rows and columns. The number of comparisons is $t + \sum_{\ell=2}^{\ell_0} \left(n_\ell \binom{\ell}{2} \right)$ because in step 2, it performs comparisons for all possible pairs of cells within each of uninterrupted rows and columns (and in step 1, a comparison is needed for each of triangular cells).

7 Concluding Remarks

In this paper, we constructed ZKP protocols using a balance scale for five pencil puzzles. We demonstrated the security of our proposed solutions, showing that they are perfectly complete, sound, and zero-knowledge. As a future work, we aim to explore other similar games. Additionally, we would like to investigate improvements that allow for the execution of the protocol with fewer coins and steps for the puzzles presented in this paper.

[2] For this, P and V should memorize the order of $\textcircled{a}$s when they are placed on the balance.

An analogous verification was considered in [7], where one confirms whether two cups contain the same number of marbles, say $X = Y$ or not. Because our model employs a balance to confirm which is heavier, say $X \geq Y$ or not, we considered an entirely different mechanism to construct a ZKP protocol. As can be observed from our ZKP protocols, Sudoku ZKP can be conducted only based on verifying $X = Y$ because it involves repeating the verification of whether the coin $\textcircled{a}$ placed in each cell is equal to the coin $\textcircled{b}$ or not.

References

1. Abe, Y., Iwamoto, M., Ohta, K.: Efficient private PEZ protocols for symmetric functions. In: Hofheinz, D., Rosen, A. (eds.) Theory of Cryptography. LNCS, vol. 11891, pp. 372–392. Springer, Cham (2019)
2. Balogh, J., Csirik, J.A., Ishai, Y., Kushilevitz, E.: Private computation using a PEZ dispenser. Theor. Comput. Sci. **306**(1), 69–84 (2003)
3. Ben-Or, M., Goldreich, O., Goldwasser, S., Håstad, J., Kilian, J., Micali, S., Rogaway, P.: Everything provable is provable in zero-knowledge. In: CRYPTO 1988. LNCS, vol. 403, pp. 37–56. Springer, New York (1990)
4. Bultel, X., Dreier, J., Dumas, J., Lafourcade, P.: Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen. In: Demaine, E.D., Grandoni, F. (eds.) Fun with Algorithms. LIPIcs, vol. 49, pp. 8:1–8:20. Schloss Dagstuhl (2016)
5. Bultel, X., Dreier, J., Dumas, J., Lafourcade, P., Miyahara, D., Mizuki, T., Nagao, A., Sasaki, T., Shinagawa, K., Sone, H.: Physical zero-knowledge proof for Makaro. In: Izumi, T., Kuznetsov, P. (eds.) SSS 2018. LNCS, vol. 11201, pp. 111–125. Springer (2018)
6. Dreier, J., Jonker, H., Lafourcade, P.: Secure auctions without cryptography. In: Ferro, A., Luccio, F., Widmayer, P. (eds.) Fun with Algorithms. LNCS, vol. 8496, pp. 158–170. Springer, Cham (2014)
7. Glaser, A., Barak, B., Goldston, R.J.: A zero-knowledge protocol for nuclear warhead verification. Nature **510**, 497–502 (2014)
8. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. Journal of Cryptology **9**(3), 167–189 (1991)
9. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: Sedgewick, R. (ed.) STOC 1985. pp. 291–304. ACM (1985)
10. Gradwohl, R., Naor, M., Pinkas, B., Rothblum, G.N.: Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. Theory of Computing Systems **44**(2), 245–268 (2009)
11. Hand, S., Koch, A., Lafourcade, P., Miyahara, D., Robert, L.: Check alternating patterns: A physical zero-knowledge proof for Moon-or-Sun. In: Shikata, J., Kuzuno, H. (eds.) IWSEC 2023. LNCS, vol. 14128, pp. 255–272. Springer (2023)
12. Hatsugai, K., Asano, K., Abe, Y.: A physical zero-knowledge proof for Sumplete, a puzzle generated by ChatGPT. In: Wu, W., Tong, G. (eds.) Computing and Combinatorics. LNCS, vol. 14422, pp. 398–410. Springer, Cham (2024)
13. Hearn, R.A., Demaine, E.D.: Games, Puzzles, and Computation. CRC Press (2009)
14. Iwamoto, C., Haruishi, M., Ibusuki, T.: Herugolf and Makaro are NP-complete. In: Ito, H., Leonardi, S., Pagli, L., Prencipe, G. (eds.) Fun with Algorithms. LIPIcs, vol. 100, pp. 24:1–24:11. Schloss Dagstuhl, Dagstuhl (2018)

15. Komano, Y., Mizuki, T.: Coin-based secure computations. Int. J. Inf. Secur. **21**, 833–846 (2022)
16. Lloyd, H., Crossley, M., Sinclair, M., Amos, M.: J-pop: Japanese puzzles as optimization problems. IEEE Transactions on Games **14**(3), 391–402 (2021)
17. Miyahara, D., Komano, Y., Mizuki, T., Sone, H.: Cooking cryptographers: Secure multiparty computation based on balls and bags. In: IEEE Computer Security Foundations Symposium. pp. 1–16. IEEE, NY (2021)
18. Miyahara, D., Robert, L., Lafourcade, P., Takeshige, S., Mizuki, T., Shinagawa, K., Nagao, A., Sone, H.: Card-based ZKP protocols for Takuzu and Juosan. In: Farach-Colton, M., Prencipe, G., Uehara, R. (eds.) Fun with Algorithms. LIPIcs, vol. 157, pp. 20:1–20:21. Schloss Dagstuhl (2021)
19. Miyahara, D., Sasaki, T., Mizuki, T., Sone, H.: Card-based physical zero-knowledge proof for Kakuro. IEICE Trans. Fundamentals **102**(9), 1072–1078 (2019)
20. Mizuki, T., Kugimoto, Y., Sone, H.: Secure multiparty computations using a dial lock. In: Cai, J., Cooper, S.B., Zhu, H. (eds.) Theory and Applications of Models of Computation. LNCS, vol. 4484, pp. 499–510. Springer (2007)
21. Moran, T., Naor, M.: Basing cryptographic protocols on tamper-evident seals. Theor. Comput. Sci. **411**(10), 1283–1310 (2010)
22. Murata, S., Miyahara, D., Mizuki, T., Sone, H.: Public-PEZ cryptography. In: Susilo, W., Deng, R.H., Guo, F., Li, Y., Intan, R. (eds.) Information Security. LNCS, vol. 12472, pp. 59–74. Springer, Cham (2020)
23. Quisquater, J.J., Quisquater, M., Quisquater, M., Quisquater, M., Guillou, L., Guillou, M.A., Guillou, G., Guillou, A., Guillou, G., Guillou, S.: How to explain zero-knowledge protocols to your children. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 628–631. Springer, New York (1990)
24. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Card-based ZKP for connectivity: Applications to Nurikabe, Hitori, and Heyawake. New Gener. Comput. **40**(1), 149–171 (2022)
25. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Hide a liar: Card-based ZKP protocol for Usowan. In: Du, D., Du, D., Wu, C., Xu, D. (eds.) Theory and Applications of Models of Computation. LNCS, vol. 13571, pp. 201–217. Springer (2022)
26. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Physical ZKP protocols for Nurimisaki and Kurodoko. Theor. Comput. Sci. **972**, 114071 (2023)
27. Ruangwises, S.: Two standard decks of playing cards are sufficient for a ZKP for Sudoku. New Gener. Comput. **40**(1), 49–65 (2022)
28. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for Ripple Effect. Theor. Comput. Sci. **895**, 115–123 (2021)
29. Sasaki, T., Miyahara, D., Mizuki, T., Sone, H.: Efficient card-based zero-knowledge proof for Sudoku. Theor. Comput. Sci. **839**, 135–142 (2020)
30. Shinagawa, K., Mizuki, T., Schuldt, J.C.N., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Secure computation protocols using polarizing cards. IEICE Trans. Fundamentals **99-A**, 1122–1131 (2016)
31. Tanaka, K., Mizuki, T.: Two uno decks efficiently perform zero-knowledge proof for Sudoku. In: Fernau, H., Jansen, K. (eds.) Fundamentals of Computation Theory. LNCS, vol. 14292, pp. 406–420. Springer, Cham (2023)
32. Uehara, R.: Computational complexity of puzzles and related topics. Interdisciplinary Information Sciences **29**(2), 119–140 (2023)
33. Yato, T., Seta, T.: Complexity and completeness of finding another solution and its application to puzzles. IEICE Trans. Fundamentals **86**(5), 1052–1060 (2003)

A Security of Sudoku

We prove the three properties of ZKP for our proposed protocol. A proof for the completeness is omitted because it is clear from the protocol description.

Lemma 1 (Perfect Completeness – Sudoku). *If P provides a correct solution of a given Sudoku grid, V is always convinced.*

Lemma 2 (Perfect Soundness – Sudoku). *If P does not provide a correct solution of a given Sudoku grid, V always rejects P's solution.*

Proof. Without loss of generality, assume that P gives an incorrect solution for a row, *i.e.*, there are two or more coins of the same weight $k \in \{1, \ldots, 9\}$ among nine coins in the same row. The first time that P compares such a coin of weight k with the challenge in step 2, P will remove a coin $\textcircled{b}$ of weight k from the challenge. Then, when P compares the coin of weight k with the challenge, no comparison should result in even, because P has already removed the coin $\textcircled{b}$ of weight k from the challenge, and a coin $\textcircled{b}$ of weight k no longer exist in the challenge. Therefore, V can always rejects an incorrect solution.

Lemma 3 (Perfect Zero-Knowledge – Sudoku). *The verifier V is not given any information other than that the prover P can solve a given Sudoku grid.*

Proof. Note that V must not know the weight of even one coin placed on a cell; otherwise, V knows a number filled with the corresponding cell in the solution. Since the weight of a coin is indistinguishable from its appearance, once P places a coin on a cell, its weight cannot be known unless V picks it. Note that our protocol lets P handle the coins when they need to be moved or touched.

In step 2b, V cannot know the weight of the commitment from comparisons, because two coins compared were shuffled in step 2a, and hence, the result of each comparison should be either left or right with a probability of $1/2$, which is independent to the solution. Moreover, when the result is even, it leaks no information on the solution because the challenge was shuffled beforehand. Therefore, V cannot learn anything throughout the whole process, except whether the solution is valid or not.

B Security of Futoshiki

We prove the security of the Futoshiki protocol. A proof for completeness is omitted because it is clear from the protocol description.

Lemma 4 (Perfect Completeness – Futoshiki). *If P knows a solution of a given Futoshiki grid, he can always convince V.*

Lemma 5 (Perfect Soundness – Futoshiki). *If P does not provide a correct solution of a given Futoshiki grid, V always rejects P.*

Proof. When P gives an incorrect solution, the following two situations are possible:

- A row (resp. column) contains the same number at least twice. In this case, V will reject P's solution in the same way as in Sudoku (see Lemma 2).
- A pair of numbers does not verify the inequality sign between them. In this case, when V observes the result of the comparison, it will notice that the inequality is not satisfied and V will reject P's solution.

Therefore, V will always reject an invalid solution.

Lemma 6 (Perfect Zero-Knowledge – Futoshiki). *The verifier V is not given any information other than that the prover P can solve the Futoshiki grid.*

Proof. We show that no information has been leaked other than that the prover P can solve the Futoshiki grid in both of the verification phases:

- In step 1, V cannot learn anything on the numbers P placed on each cell for the same reason as in the Sudoku ZKP protocol (Lemma 3).
- In step 2, when V checks whether the numbers satisfy the inequality rule, as V does not touch the coins but only observes the result of the scale, V can only learn which coin is heavier (the coins are visually indistinguishable). Hence V still does not learn anything on P's solution except that it is correct.

Therefore, V cannot learn anything throughout the whole process, except whether the solution is valid or not.

C Security of Makaro

We prove the security of the Makaro protocol. A proof for completeness is omitted because it is clear from the protocol description.

Lemma 7 (Perfect Completeness – Makaro). *If P knows a solution of a given Makaro grid, he can always convince V.*

Lemma 8 (Perfect Soundness – Makaro). *If P does not provide a collect solution of a given Makaro grid, V always rejects P.*

Proof. When P gives an incorrect solution, the following three situations are possible.

- A room of size s contains twice the same number, or a number not in $\{1, \ldots, s\}$. In this case, V will reject P's solution as in Sudoku (see Lemma 2).
- The number in the cell pointed to by the arrow is not the highest. In this case, a comparison using the balance reveals that the coin in that cell is not the heaviest, and V rejects P's solution.
- Adjacent cells contains the same number. In this case, the balance will be even when comparing the coins in these cells, and V will reject P's solution.

Therefore, when P does not give the correct answer, V will always reject.

Lemma 9 (Perfect Zero-Knowledge – Makaro). *The verifier V is not given any information other than that the prover P can solve the Makaro grid.*

Proof. We show that no information has been leaked other than that the prover P can solve the Makaro grid through the following three checks in the verification phase:

- In step 1, V does not learn anything except that each room of size s contains one and only one number i for all $i \in \{1, \ldots, s\}$ for the exact same reason as for Sudoku (see Lemma 3).
- In step 2, when comparing the coins around the arrow cell, V does not learn anything except for which is the heaviest because the coins are visually indistinguishable and V never touches them and only observes the balance results.
- In step 3, when V checks that no adjacent cells contain the same number, as the coins are shuffled before the comparison and never re-used after, V cannot learn anything except whether they are of different weight.

Therefore, V cannot learn anything throughout the whole process, except whether the solution is valid or not.

D Security of Kakuro

We prove the security of the Kakuro protocol. A proof for completeness is omitted because it is clear from the protocol description.

Lemma 10 (Perfect Completeness – Kakuro). *If P knows a solution of a given Kakuro grid, he can always convince V.*

Lemma 11 (Perfect Soundness – Kakuro). *If P does not provide a correct solution of a given Kakuro grid, V always rejects P's solution.*

Proof. When P gives an incorrect solution, the following two situations are possible.

- A number in a triangular cell and the sum of the subsequent numbers from that triangle cell are not equal. In this case, the weight of the coin ⓑ representing the triangular cell number and the sum of the weights of the coins ⓐ in the consecutive white cells from that triangle cell are not equal, causing the balance to be unbalanced. Hence, V will reject P's solution.
- The same number is included twice in a block formed by consecutive white cells either vertically or horizontally. In this case, during the comparison of the coins ⓐ in the white cells of the block, the weights of two coins are equal, resulting in a balanced scale. Hence, V will reject P's solution.

Therefore, when P does not give the correct answer, V will always reject its solution.

Lemma 12 (Perfect Zero-Knowledge – Kakuro). *The verifier V cannot learn any information other than that the prover P can solve the Kakuro grid.*

Proof. We show that no information has been leaked other than that the prover P can solve the Kakuro grid through each step of the verification phase:

- In step 1, V checks that the number in the triangular cell is equal to the sum of the subsequent numbers. The coins ⓐs from consecutive white cells are stacked before placing them on the scale. Hence V does not learn anything on their individual weight (they are visually indistinguishable), except that the sum of their weight is the same as the weight of the corresponding coin ⓑ.
- In step 2, V ensures that uninterrupted rows and columns do not contain twice the same number. The coins ⓐs are shuffled, and each pair of coins is compared. Since the initial positions of each coin cannot be identified, V cannot determine the numbers on the white cells.

Hence, V cannot learn anything on P's solution throughout the whole protocol.