# Zero-knowledge Proof Protocol for Cryptarithmetic Using Dihedral Cards[⋆]

Raimu Isuzugawa[1], Daiki Miyahara[1,2], and Takaaki Mizuki[1,2]

[1] Tohoku University, Sendai, Japan
raimu.isuzugawa.q6[atmark]dc.tohoku.ac.jp,
mizuki+lncs[atmark]tohoku.ac.jp
[2] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

**Abstract.** Cryptarithmetic, also known as Verbal Arithmetic or Word Addition, is a popular pencil puzzle in which the aim is to deduce which letter corresponds to which numeral, given a mathematical equation in which each numeral (from 0 to 9) has been replaced with a unique letter. The most famous instance of this puzzle is probably "SEND + MORE = MONEY", whose solution is "9567 + 1085 = 10652", i.e., S = 9, E = 5, N = 6, D = 7, M = 1, O = 0, R = 8, and Y = 2. In this study, we construct a physical zero-knowledge proof protocol for a Cryptarithmetic puzzle: That is, our protocol enables a prover who knows a solution to the puzzle to convince a verifier that he/she knows the solution without revealing any information about it. The proposed protocol uses a physical deck of "dihedral cards," which were developed by Shinagawa in 2019.

**Keywords:** Cryptarithmetic · Dihedral Cards · Physical Zero-knowledge Proof · Card-based Cryptography

## 1 Introduction

*Cryptarithmetic*, also known as Verbal Arithmetic or Word Addition, is a famous pencil puzzle: given an equation, such as "SEND + MORE = MONEY" (Fig. 1(a)), where each numeral from 0 to 9 has been replaced with a unique letter, one has to guess which letter corresponds to which numeral. The solution to the aforementioned example is presented in Fig. 1(b): That is, the correspondences are S $\mapsto$ 9, E $\mapsto$ 5, N $\mapsto$ 6, D $\mapsto$ 7, M $\mapsto$ 1, O $\mapsto$ 0, R $\mapsto$ 8, and Y $\mapsto$ 2.

The rules for solving Cryptarithmetic puzzles are as follows.

1. Any letter (at every position) corresponds to the same numeral, and different letters correspond to different numerals.
2. The most significant digit (letter) must not correspond to 0.
3. After all letters are replaced with their numerals, the resulting equation must be mathematically correct.

In this paper, we shall construct a *zero-knowledge proof* protocol for Cryptarithmetic. We begin by explaining what zero-knowledge proof protocols for pencil puzzles are.

[⋆] This paper appears in Proceedings of UCNC 2021. The final authenticated version is available online at https://doi.org/10.1007/978-3-030-87993-8_4.

$$\begin{array}{r} \mathrm{S\ E\ N\ D} \\ +\ \mathrm{M\ O\ R\ E} \\ \hline \mathrm{M\ O\ N\ E\ Y} \end{array}$$

(a)

$$\begin{array}{r} 9\ 5\ 6\ 7 \\ +\ 1\ 0\ 8\ 5 \\ \hline 1\ 0\ 6\ 5\ 2 \end{array}$$

(b)

Fig. 1: A puzzle instance of Cryptarithmetic and its solution

## 1.1 Zero-knowledge Proofs for Puzzles

Consider a situation in which there are two players: a prover $P$ and a verifier $V$; the prover $P$ knows a solution $w$ to an instance $x$ of a puzzle (such as Cryptarithmetic and Sudoku[i]), whereas the verifier $V$ does not know any solution to $x$. The verifier $V$ spent considerable time attempting to find a solution to $x$, but $V$ was unable to find it. Thus, $V$ becomes skeptical and asks $P$ to prove that there is a solution to $x$. However, if $P$ only shows the solution $w$ to $V$, the puzzle instance $x$ will not be worth solving. A *zero-knowledge proof* protocol, whose concept was first conceived in [6], can solve this dilemma: It enables $P$ to convince $V$ of the existence of $w$ without revealing any information about $w$ (that only $P$ knows), satisfying the following three properties.

**Completeness.** If $P$ knows $w$, then $V$ is convinced of the existence of $w$.
**Extractability.** If $P$ does not know $w$, then $V$ is not convinced.
**Zero-knowledge.** $V$ does not obtain any information about $w$.

In 2007, Gradwohl et al. [7, 8] first constructed zero-knowledge proof protocols for Sudoku using physical daily-use objects such as a deck of playing cards. Since then, many zero-knowledge proof protocols for pencil puzzles using a deck of physical cards, which we call *card-based ZKP protocols*, have been proposed, such as those for Akari [1], Hashiwokakero [27], Hitori [21], Juosan [14], Kakuro [1, 15], KenKen [1], Makaro [2], Masyu [10], Nonogram [3,22], Norinori [4], Numberlink [24,25], Nurikabe [21], Ripple Effect [26], Slitherlink [10, 11], Sudoku [23, 28, 29], and Takuzu [1, 14]. These physical zero-knowledge proof protocols do not depend on computers or programs; hence, it is relatively easy for lay people to perform zero-knowledge proof and/or to understand its concept.

## 1.2 Our Contribution

It should be noted that all the pencil puzzles listed above are played with a rectangular grid (consisting of many cells): That is, all the existing card-based ZKP protocols have been designed to manipulate a grid with numbers and/or symbols. By contrast, Cryptarithmetic, for which this study shall design a zero-knowledge proof protocol, is played not with a grid but with an equation, as already seen in Fig. 1. Therefore, another technique or treatment is required to construct a card-based ZKP protocol for Cryptarithmetic. Furthermore, while most of the existing protocols use a deck of

[i] Sudoku is the most famous pencil puzzle, which has been published by NIKOLI Co., Ltd. (`https://www.nikoli.co.jp/en/`)

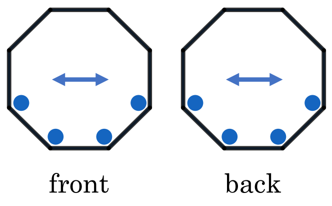

Fig. 2: Dihedral card of $2m$-sided polygon for $m = 4$

physical cards consisting of black and red cards $\boxed{\clubsuit}\boxed{\clubsuit}\cdots\boxed{\heartsuit}\boxed{\heartsuit}\cdots$ and/or number cards $\boxed{1}\boxed{2}\boxed{3}\cdots$, these cards are not suitable for computing an arithmetic addition; in particular, computing a carry causes a heavy load (e.g., [17]). Therefore, we need to consider other types of physical cards.

In this paper, we construct a zero-knowledge proof protocol for Cryptarithmetic using *dihedral cards*, as illustrated in Fig. 2; these novel cards were proposed by Shinagawa in 2019 [30, 31][ii]. That is, using our proposed protocol, a prover $P$ who knows the solution to a given Cryptarithmetic puzzle can convince a verifier $V$ that $P$ knows the solution without revealing any information about it. As will be seen in Sects. 2 and 3, the dihedral cards are suitable for Cryptarithmetic because they can efficiently compute an arithmetic addition with a carry (compared with other regular polygon cards [32], or a normal deck of cards as mentioned above). After we present our protocol in Sect. 3, we evaluate its performance and demonstrate its correctness in Sect. 4. The paper is concluded in Sect. 5.

Although Cryptarithmetic puzzles can have multiple solutions or can be an equation whose left-hand side has more than two terms (for addition) [33–35], we focus on Cryptarithmetic puzzles with the addition of exactly two terms such that there is a unique solution (as shown in Fig. 1) throughout this paper. (We will revisit this point in Sect. 5.)

Note that if we allow the base, denoted by $k$, in arithmetic to be arbitrary (aside from $k = 10$, i.e., decimal arithmetic), the decision problem for Cryptarithmetic becomes $\mathsf{NP}$-complete [5] (where $k$ is not fixed). The computational complexity of Cryptarithmetic has been studied, including methods for efficiently deriving solutions using genetic algorithm [13] and automata that accept Cryptarithmetic problems for bases $k \leq 7$ [19]. If we fix $k$, say $k = 10$ as in this paper, then we can find a solution to a given Cryptarithmetic problem (if any) by enumerating at most 10! assignments of numerals to the given letters. However, pencil puzzles such as Cryptarithmetic are usually solved with a pen and sheets of paper; hence, without the aid of a computer, 10! possibilities cannot be enumerated by hands; thus, it is worthwhile to perform zero-knowledge proofs even for puzzles in $\mathsf{P}$.

[ii] Figs. 2 to 11 were created based on the figures presented in [31].

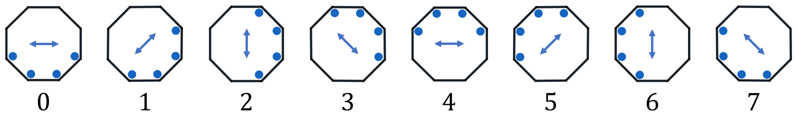

Fig. 3: How to represent integers using a dihedral card of $m = 4$

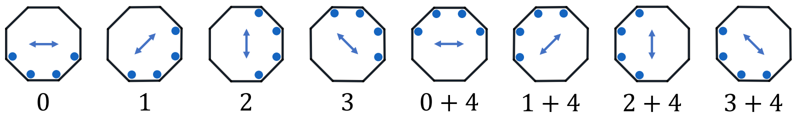

Fig. 4: How to represent integers using a dihedral card considering a carry

## 2 Preliminaries: Dihedral Cards [30,31]

In this section, we describe the dihedral cards proposed by Shinagawa [30,31], operations on them, and some sub-protocols that will be used when constructing our protocol in Sect. 3.

### 2.1 Dihedral Cards

A *dihedral card* is a regular polygon card marked with *invisible ink*. We use dihedral cards, each of which is a regular $2m$-sided polygon, to conduct computations on a finite field $\mathbb{Z}/\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$ for a positive integer $m$. Fig. 2 shows an example of a dihedral card of a $2m$-sided polygon for $m = 4$, i.e., a regular octagon card. The blue arrows and dots in Fig. 2 have been drawn using invisible ink. Specifically, a symmetric bidirectional arrow is marked in the center, and $m$ dots corresponding to $m$ consecutive vertices (among $2m$ ones) are marked. The same pattern is drawn on the back of the card, i.e., every vertex marked with a dot in the front also has a dot in the back.

The physical property of invisible ink guarantees that blue arrows and dots are invisible to the naked eye, but they can be made visible by illuminating with black light. Therefore, markings on a dihedral card can only be confirmed when illuminated with black light.

Because dihedral cards have the shape of a regular $2m$-sided polygon, we can simply represent every integer from 0 to $2m - 1$ depending on the angle at which the cards are placed, as shown in Fig. 3. However, in our proposed protocol, the values from $m$ to $2m - 1$ are treated as the values from 0 to $m - 1$ with a carry, as shown in Fig. 4. Given a $2m$-sided polygon card having a value $x \in \mathbb{Z}/\mathbb{Z}_{2m}$, if we rotate it by $c\pi/m$ degrees for an integer $c$, its value is changed from $x$ to $x + c \bmod 2m$; we refer to this action as "rotating a card by a degree $c$."

In addition to rotation, a dihedral card can be transformed by flipping it face up or down based on a certain axis of rotation. In this study, we use three axes, i.e., three flipping methods: Flip the card vertically, as shown in Fig. 5; diagonally, as shown in Fig. 6; and horizontally, as shown in Fig. 7.

Remember that the value of a dihedral card can be revealed by illuminating with black light. Next, we describe methods for partially revealing the values of a dihedral

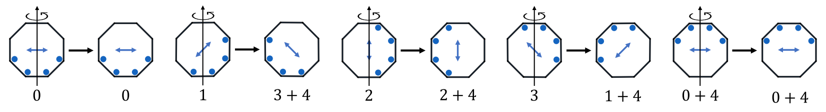

Fig. 5: Flipping a card vertically

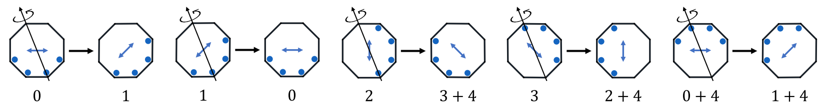

Fig. 6: Flipping a card diagonally

card. That is, assume that, given a $2m$-sided polygon card with a value $x \in \mathbb{Z}/\mathbb{Z}_{2m}$, we want to obtain the one-bit value $p(x \geq m)$ and/or the $m$-valued $x \bmod m$. Here, we define $p(x \geq m) = 1$ if $x \geq m$, and $p(x \geq m) = 0$ if $x < m$. We call $p(x \geq m)$ the *sign* of the card, and $x \bmod m$ the mod-$m$ *value* of the card. This can be achieved by covering areas that are not related to the value to be observed; Fig. 8 and Fig. 9 illustrate how to reveal the sign and the mod-$m$ value of a card, respectively, using covers of special shapes. As can be verified in Fig. 4, the sign of a card determines whether or not the digit has a carry, and a dot mark at the point where the black light is irradiated, as shown in Fig. 8, corresponds to this sign. On the other hand, the mod-$m$ value can be obtained by looking only at the arrow in the center illuminated by the black light because its direction reveals the mod-$m$ value only (without its possible carry). In Fig. 8 and Fig. 9, the sign $p(x \geq m)$ is 0, and the mod-$m$ value is 1.

## 2.2 Shuffle Operations on Dihedral Cards

Here, considering a sequence of dihedral cards, we describe two shuffle operations. Let a positive integer $m$ be fixed, and denote by $[\![x]\!]$ a $2m$-sided polygon card with a value of $x \in \mathbb{Z}/\mathbb{Z}_{2m}$. For a positive integer $i$, we define $[i] = \{1, 2, \ldots, i\}$. Given a sequence of $\ell$ regular $2m$-sided dihedral cards $([\![x_1]\!], [\![x_2]\!], \ldots, [\![x_\ell]\!])$, we consider two types of shuffle, as follows.

**Rotation shuffle** A *rotation shuffle* with a set $T \subseteq [\ell]$ rotates all cards whose positions are in $T$ by a uniformly distributed random number $r \in \mathbb{Z}/\mathbb{Z}_{2m}$; that is, it shuffles all cards specified by $T$ together. In this case, the sequence of the cards $([\![x_1]\!], [\![x_2]\!], \ldots, [\![x_\ell]\!])$ becomes $x_i \to x_i + r \pmod{2m}$ when $i \in T$, and $x_i \to x_i$ when $i \notin T$.

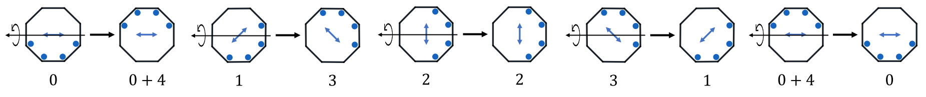

Fig. 7: Flipping a card horizontally

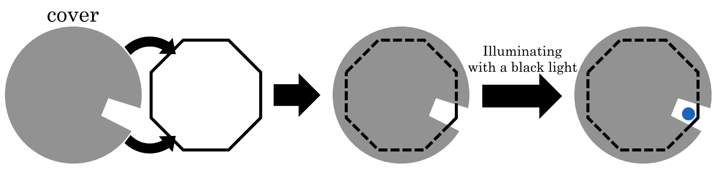

Fig. 8: Opening the sign

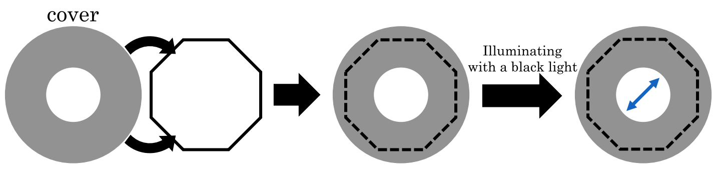

Fig. 9: Opening the mod-$m$ value

**Two-sided rotation shuffle** A *two-sided rotation shuffle* is an operation that randomly rotates all cards whose positions are in $T \subseteq [\ell]$ by $\pi$. In this case, the card sequence $([[x_1]], [[x_2]], \ldots, [[x_\ell]])$ becomes $x_i \to x_i + rm \pmod{2m}$ when $i \in T$, and $x_i \to x_i$ when $i \notin T$, where $r$ is a random bit $r \in \{0, 1\}$. This shuffle can be implemented using two clips to fix and rotate the cards, as illustrated in Fig. 10.

### 2.3 Protocols with Dihedral Cards

In this subsection, we briefly introduce several basic protocols for computations working on dihedral cards [30, 31]; refer to [31] for details.

**2.3.1 Initialization Protocol** The initialization protocol takes as input a card $[[x]]$ such that $x \in \mathbb{Z}/\mathbb{Z}_{2m}$ and initializes its value to 0: $[[x]] \Rightarrow [[0]]$. It proceeds as follows.

1. Apply a rotation shuffle to the card.
2. Illuminate the whole card with black light and let the opened value be $x' \in \mathbb{Z}/\mathbb{Z}_{2m}$.
3. Rotate the card by a degree $-x'$.

This protocol requires one shuffle.

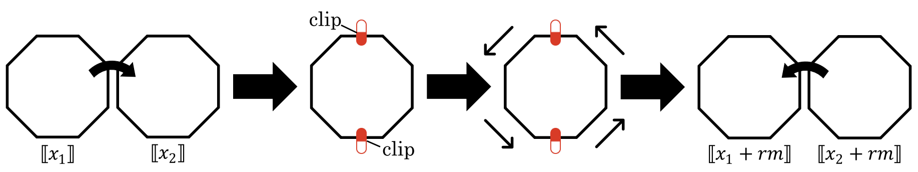

Fig. 10: Implementation of two-sided rotation shuffle ($\ell = 2$)

**2.3.2 Addition Protocol** The addition protocol takes as input two cards $[\![x_1]\!], [\![x_2]\!]$ such that $x_1, x_2 \in \mathbb{Z}/\mathbb{Z}_{2m}$, and outputs the arithmetic addition of the two (and $[\![0]\!]$):

$$([\![x_1]\!], [\![x_2]\!]) \Rightarrow ([\![0]\!], [\![x_1 + x_2 \bmod 2m]\!]).$$

It proceeds as follows.

1. Flip the left card vertically, as shown in Fig. 5, to obtain $[\![-x_1]\!]$.
2. Apply a rotation shuffle to the sequence of two cards.
3. Illuminate the entire left card with black light. Let the opened value be $x' \in \mathbb{Z}/\mathbb{Z}_{2m}$.
4. Rotate the sequence of cards by a degree $-x'$.

This protocol also requires one shuffle.

**2.3.3 Sign Normalization Protocol** The sign normalization protocol takes as input a card $[\![x]\!]$ such that $x \in \mathbb{Z}/\mathbb{Z}_{2m}$ and changes its value to $x \bmod m$:

$$[\![x]\!] \Rightarrow [\![x \bmod m]\!].$$

It proceeds as follows.

1. Apply a two-sided rotation shuffle to the card.
2. Reveal the sign of the card (using the method illustrated in Fig. 8). Let $s' \in \{0, 1\}$ be the sign of the card.
3. Rotate the card by a degree $s'm$.

This protocol uses one shuffle.

**2.3.4 Sign-to-Value Protocol** The sign-to-value protocol takes as input a card $[\![x]\!]$ such that $x \in \mathbb{Z}/\mathbb{Z}_{2m}$ (along with card $[\![0]\!]$), and outputs the sign of the card (and $[\![0]\!]$):

$$([\![x]\!], [\![0]\!]) \Rightarrow ([\![p(x \geq m)]\!], [\![0]\!]).$$

It proceeds as follows.

1. Apply a two-sided rotation shuffle to the sequence of two cards.
2. Reveal the sign of the left card. Let $s_1 \in \{0, 1\}$ be the revealed sign.
3. Rotate the right card by a degree $s_1 m$.
4. Apply the initialization protocol to the left card. We now have $([\![0]\!], [\![p(x \geq m) \cdot m]\!])$.
5. Consider a diagonal axis (as in Fig. 6) for the left card and a horizontal axis (as shown in Fig. 7) for the right card. Then, the cards are randomly flipped together based on these axes; to achieve this, after adjusting the degree of the left card, fix the two cards together with two plates, as illustrated in Fig. 11, and repeatedly rotate them quickly.
6. Reveal the sign of the right card. Let $s_2 \in \{0, 1\}$ be the revealed sign.
   (a) If $s_2 = 0$, output the current sequence.
   (b) If $s_2 = 1$, rotate the right card by a degree $m$, and then, flip the left card diagonally (as shown in Fig. 6).

Three shuffles are required for the sign-to-value protocol.

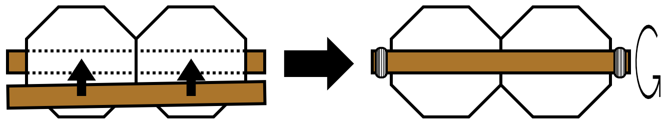
Fig. 11: Implementation of uniform random flipping

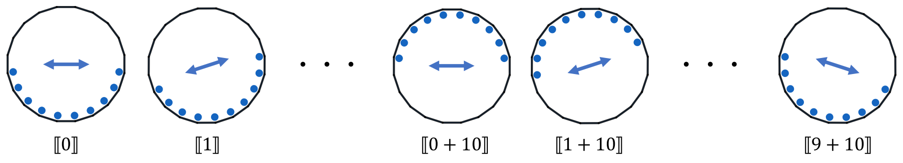

Fig. 12: Dihedral cards of regular 20-sided polygons

## 3 Zero-knowledge Proof Protocol for Cryptarithmetic

In this section, we construct a card-based zero-knowledge proof protocol for Cryptarithmetic using dihedral cards, utilizing the basic sub-protocols introduced in the previous section.

First, in Sect. 3.1, we propose a copy protocol for use in our proposed protocol. This copy protocol creates multiple dihedral cards with the same value from one dihedral card without revealing any information about its value. We then describe the procedure for our proposed protocol in Sect. 3.2. To deal with decimal arithmetic in Cryptarithmetic, we set $m = 10$, i.e., our protocol uses dihedral cards, each of whose shape is a regular 20-sided polygon, as illustrated in Fig. 12.

In Sect. 4, we will evaluate the numbers of cards and shuffles required for executing our proposed protocol and prove that our protocol satisfies the three properties of the zero-knowledge proof.

### 3.1 How to Duplicate Commitment

Let us call a dihedral card with a value $x \in \mathbb{Z}/\mathbb{Z}_{2m}$ a *commitment* to $x$. We present a copy protocol that duplicates a given commitment. As can be observed in Sect. 3.2, we use this copy protocol to duplicate a commitment to every $i$, $0 \leq i \leq 9$, when setting up our proposed protocol.

Given a commitment $[[x]]$, our copy protocol making $\ell$ ($\geq 2$) copied commitments proceeds as follows.

1. Place a sequence of $\ell$ dihedral cards $[[0]]$, all having a value of 0, next to the given commitment to be copied.
2. Apply the addition protocol to the sequence of cards so that the value of the given commitment is added to all the $\ell$ cards, resulting in a sequence of $\ell$ commitments $([[x]], \ldots, [[x]])$. (Note that the addition protocol presented in Sect. 2.3.2 takes only two cards as input, but one can easily extend it by rotating the $\ell + 1$ cards together.)

In this protocol, the given commitment is duplicated by adding its value to the desired number of dihedral cards that we want to obtain. Thus, it requires $\ell$ dihedral cards as well as a given commitment, and requires only one shuffle.

### 3.2 Procedure

In this subsection, we describe the procedure for our proposed protocol. Given a Cryptarithmetic problem, our protocol enables a prover $P$ who knows the solution to the problem to convince a verifier $V$ that $P$ knows the solution without revealing any information about the solution. It consists of four phases: Setup, Adding least significant digits (half adder), Adding higher digits (full adder), and Verification.

*Setup.* In this phase, dihedral cards corresponding to the solution are created.

1. Prepare a commitment to $i$ for every $i$, $0 \le i \le 9$, i.e., $[\![0]\!], \ldots, [\![9]\!]$. The values of the commitments should be disclosed so that $V$ can be convinced that every commitment corresponds to a distinct integer.
2. Prepare *symbolic cards* corresponding to the letters appearing in the puzzle instance, as illustrated in Fig. 13; this example corresponds to the puzzle shown in Fig. 1, i.e., we have eight cards with a letter S, E, N, D, M, O, R, or Y on their front, along with two *dummy* cards with blank surfaces, where all 10 cards have indistinguishable backs. The letters on the front can be numbers (as in the case of playing cards), but for the sake of clarity, we use symbolic cards that have the same letters that appeared in the puzzle instance. If the number of letters appearing in the puzzle is less than 10, dummy cards are used for the missing letters, as in the example above.
3. Remember that only the prover $P$ knows the solution, i.e., the one-to-one correspondence between numerals and letters. The prover $P$ takes all the symbolic cards in his/her hand, and places them face-down below the 10 commitments (which were prepared in the first step), such that each pair of a commitment and a symbolic card follows the one-to-one correspondence, as illustrated in Fig. 14, where a dummy card is placed if there is no letter corresponding to that numeral. Because commitments to 3 and 4 do not appear in the example solution, dummy cards with blank surfaces are placed below them. Note that the face-down 10 symbolic cards have been placed secretly by $P$ without $V$ knowing their order.

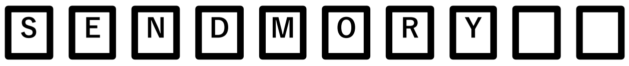

Fig. 13: Examples of symbolic cards

4. Fix the 10 pairs of commitments and symbolic cards (or dummy cards) using envelopes or clips and then shuffle them by hand. This shuffling operation is called a *pile-scramble shuffle* [9]. The shuffle is performed by $P$ and/or $V$; they can repeat shuffling until both of them are satisfied.

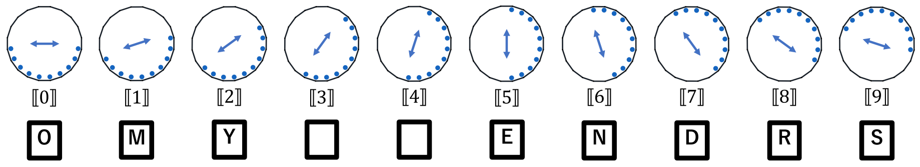

Fig. 14: Example of correspondence between 10 dihedral cards and symbolic cards; *P* places symbolic cards face-down

5. Turn over all the symbolic cards to see the mapping from letters to commitments. If the revealed card is a dummy card, the corresponding commitment can be discarded after applying a rotation shuffle.
6. Remember that a puzzle instance has an equation where the left-hand side is an addition of two sequences of letters; without loss of generality, we assume that the number of letters in the second term on the left-hand side is greater than or equal to that in the first term. Repeatedly execute the copy protocol presented in Sect. 3.1 to ensure that we have a sufficient number of duplicated commitments to accomplish the following: (i) for every letter in the first term on the left-hand side of addition (in the puzzle instance), place one commitment corresponding to that letter; (ii) for every letter in the second term, place two commitments corresponding to that letter; and (iii) for every letter on the right-hand side, place one commitment corresponding to that letter. (The two commitments in (ii) will be used to obtain commitments to both an addition result and a carry.)

Let us illustrate how to arrange commitments in Step 6 by considering the puzzle problem shown in Fig. 1 as an example; note that "SEND" is the first term on the left-hand side, "MORE" is the second term, and "MONEY" is the right-hand side. Because letters M, O, R, and E (which constitute the second term) appear twice, twice, once, and thrice, respectively, we apply the copy protocol to obtain three commitments corresponding to “M,” three commitments corresponding to “O,” two commitments corresponding to “R,” and four commitments corresponding to “E.” For the other letters, i.e., S, N, D, and Y, we obtain as many commitments as appeared in the problem. After copying, we place the obtained commitments at the corresponding positions on the board, as illustrated in Fig. 15. Note that there should be two commitments corresponding to each of letters M, O, R, and E in the figure. Recall that the blue arrows and dots in Fig. 15 were drawn with invisible ink, and that *V* does not know any values of the cards.

*Addition of least significant digits (half adder).* In this phase, we compute the addition of the two commitments corresponding to the least significant digits placed in the Setup phase and output commitments to the result of addition and carry (to the higher digits), i.e., we perform the half adder.

1. Utilizing the addition protocol introduced in Sect. 2.3.2, add the value of the commitment corresponding to the least significant digit in the first term (on the left-hand side) to the two commitments corresponding to the least significant digit in the second term. (Recall that two commitments were placed for every letter in the second

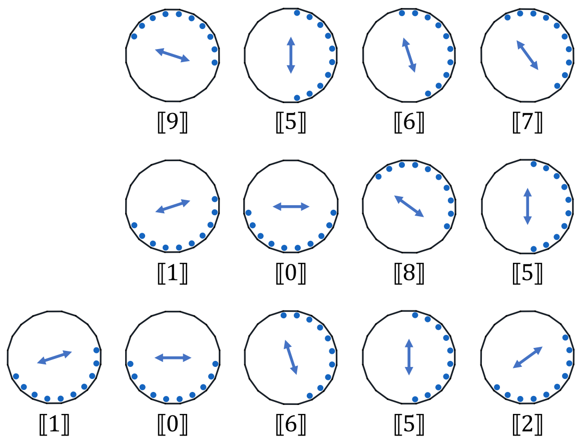

Fig. 15: Example of commitment placement

term.) Therefore, $P$ and $V$ obtain two identical commitments corresponding to the result of the addition.

2. Apply the sign-to-value protocol introduced in Sect. 2.3.4 to one of the two commitments obtained in the previous step to obtain a commitment to the carry to the higher digits.
3. Apply the sign normalization protocol introduced in Sect. 2.3.3 to the other commitment, i.e., the one that was not used in the previous step. (Thanks to this step, any commitment has a value between 0 and $m - 1$.)

We do not explicitly use the carry protocol proposed by Shinagawa [30,31] to obtain a commitment corresponding to a carry of addition, but we do employ the same idea behind the protocol, i.e., combining the addition and sign-to-value protocols.

*Addition of higher digits (full adder).* In this phase, we compute the addition of commitments placed on higher digits in the Setup phase and a commitment to a carry from the lower digits so that we obtain commitments to the addition result and a carry, i.e., we perform the full adder. For every pair of higher digits (on the left-hand side of the equation), we perform the following one by one. That is, execute Steps 1 to 4 from the second lowest digit until the addition of the most significant digit is completed.

1. Utilizing the addition protocol, add the commitment $[[x]]$ in the first term (on the right-hand side) to the two commitments $[[y]], [[y]]$ in the second term to obtain two commitments to the result of addition $[[x + y]], [[x + y]]$.
2. Similarly, add the commitment to the carry $[[c]]$ to the two commitments obtained in Step 1 to obtain two commitments to the result of addition with carry $[[x + y + c]], [[x + y + c]]$.
3. Apply the sign normalization protocol to one of the two commitments obtained in Step 2.
4. Apply the sign-to-value protocol to the other commitment to obtain a commitment corresponding to a carry to the higher digit.

*Verification.* In this phase, we verify whether the rules of Cryptarithmetic are satisfied using the commitments placed in the Setup phase and those obtained in the addition phases.

1. The verifier $V$ checks that every most significant digit is not equal to 0 by partially illuminating the commitment with black light, as shown in Sect. 2.1. This is possible because of the application of the sign normalization protocol in the addition phase. If $V$ finds a value of 0 (for at least one of them), then $V$ rejects it.
2. The verifier $V$ checks that the result of addition is equal to the left-hand side: Apply a rotation shuffle to every pair of commitments that should correspond to the same letter, i.e., a commitment placed in the Setup phase and the one obtained in the addition phase, and then reveal their values. If $V$ finds a pair with different values, then $V$ rejects it.

## 4 Evaluation

In this section, we demonstrate that our protocol constructed in Sect. 3 works. Specifically, in Sect. 4.1, we count the numbers of cards and shuffles required for our proposed protocol. In Sect. 4.2, we show that our protocol is surely a zero-knowledge proof protocol.

### 4.1 Numbers of Cards and Shuffles

In this subsection, we evaluate the performance of our proposed protocol, i.e., we count the numbers of required cards and shuffles.

Because the number of symbolic cards is fixed, i.e., it is always 10, we consider only the number of dihedral cards. Regarding the number of shuffles, we consider the worst case. For simplicity, a puzzle instance is assumed to be an equation in which the two terms on the left-hand side have the same number of letters and the right-hand side has one more than that, e.g., the puzzle shown in Fig. 1. We denote the number of letters in the first term on the left-hand side by $d$. Note that the second term also has $d$ letters and the right-hand side has $d + 1$ letters; hence, the total number of letters in the puzzle instance is $3d + 1$.

Let us count the number of required dihedral cards. After the setup phase, there are $4d + 1$ commitments (cards), and to produce such copied commitments, one more card is required during the final copy. Therefore, the protocol uses $4d + 2$ cards.

Next, let us count the number of shuffles. For the setup, in addition to the pile-scramble shuffle, the initialization protocol and the copy protocol use a shuffle and they are executed at most 10 times in total. Thus, the total number of shuffles in the setup is 11. To add the least significant digit, the addition, sign-to-value, and sign normalization protocols are performed once each. Thus, the total number of shuffles here is 5. For the addition of the higher digits, the addition is performed twice, sign-to-value and sign normalization protocols are performed once per digit, and the process is repeated for the number of digits to be added. Thus, the total number of shuffles here is $6(d - 1)$. For verification, the rotation shuffle is executed for the number of digits of the addition

Tab. 1: Number of cards and shuffles in the proposed protocol

| Number of cards | Number of shuffles |
|---|---|
| $4d + 2$ | $7d + 11$ |

result. Thus, the total number of shuffles in the verification is $d + 1$. Therefore, the total number of shuffles in the entire protocol is at most $7d + 11$.

Table 1 shows the number of cards and the number of shuffles derived from the foregoing discussion.

### 4.2 Proof

In this subsection, we verify that the proposed protocol described in Sect. 3.2 satisfies the zero-knowledge proof property.

**Completeness.** The verifier $V$ verifies that the values of the 10 dihedral cards are all different, from 0 to 9, in Step 1 of the Setup phase. Because the same commitment is assigned to the same letter and different commitments are assigned to different letters, $V$ is convinced that the same numeral is assigned to the same letter and different numerals are assigned to different letters. Furthermore, if the prover $P$ knows the solution, $P$ places the symbolic card corresponding to each letter of the puzzle instance that satisfies the rules; hence, $V$ is convinced that the solution that $P$ knows satisfies the rule in the Verification phase.

**Extractability.** As mentioned above, the value of any commitment in the Setup phase is between 0 and 9. If $P$ gives a false input, i.e., a numeral that does not match the solution for a certain letter, $V$ will not be convinced because the solution is assumed to be unique in this study and will output an addition result that is different from the solution, meaning that the value disclosed in Step 2 of the Verification phase will be different.

**Zero-Knowledge.** Because the commitments corresponding to the letters are prepared using symbolic cards so that $V$ does not know the correspondence, $V$ cannot see which numeral corresponds to which letter in the Setup phase. In the Addition phases, the values of both input and output are not disclosed, so that no information about the solution is leaked to $V$. In the Verification phase, the values of the commitments are disclosed in Step 2, but $V$ cannot know the information about the original value due to the shuffle operation.

## 5 Conclusion

In this paper, we proposed a card-based ZKP protocol for Cryptarithmetic using dihedral cards. Our protocol was obtained by constructing copy, half-adder, and full-adder protocols working on dihedral cards with the help of existing basic sub-protocols.

Our future work includes improving the efficiency of the protocol. For example, as shown in Fig. 1, when the number of letters on the right-hand side is larger than that of each of the two terms on the left-hand side, the numeral corresponding to the most significant digit on the right-hand side is automatically determined to be 1. Therefore, the number of shuffles can be reduced by disclosing the commitment to confirm whether it is 1, instead of a rotation shuffle. It would also be interesting to measure the execution time of our protocol in more detail, e.g., [16], or to adopt the "private permutation" model, which allows players' private actions, e.g., [12, 18, 20, 36].

In actual examples [33–35], there are many Cryptarithmetic problems involving the addition of two inputs as well as the addition of three or more inputs. Therefore, one of the tasks is to consider the application to the addition of three or more inputs.

## Acknowledgements

We thank the anonymous referees, whose comments have helped us improve the presentation of the paper. We would like to thank Hideaki Sone for his cooperation in preparing a Japanese draft version at an earlier stage of this work. This work was supported in part by JSPS KAKENHI Grant Numbers JP19J21153 and JP21K11881.

## References

1. Bultel, X., Dreier, J., Dumas, J.G., Lafourcade, P.: Physical zero-knowledge proofs for Akari, Takuzu, Kakuro and KenKen. In: Demaine, E.D., Grandoni, F. (eds.) Fun with Algorithms. Leibniz International Proceedings in Informatics (LIPIcs), vol. 49, pp. 8:1–8:20. Schloss Dagstuhl, Dagstuhl, Germany (2016), https://doi.org/10.4230/LIPIcs.FUN.2016.8
2. Bultel, X., Dreier, J., Dumas, J., Lafourcade, P., Miyahara, D., Mizuki, T., Nagao, A., Sasaki, T., Shinagawa, K., Sone, H.: Physical zero-knowledge proof for Makaro. In: Stabilization, Safety, and Security of Distributed Systems. LNCS, vol. 11201, pp. 111–125 (2018), https://doi.org/10.1007/978-3-030-03232-6_8
3. Chien, Y.F., Hon, W.K.: Cryptographic and physical zero-knowledge proof: From Sudoku to Nonogram. In: Boldi, P., Gargano, L. (eds.) Fun with Algorithms. LNCS, vol. 6099, pp. 102–112. Springer, Berlin, Heidelberg (2010), https://doi.org/10.1007/978-3-642-13122-6_12
4. Dumas, J.G., Lafourcade, P., Miyahara, D., Mizuki, T., Sasaki, T., Sone, H.: Interactive physical zero-knowledge proof for Norinori. In: Du, D.Z., Duan, Z., Tian, C. (eds.) Computing and Combinatorics. LNCS, vol. 11653, pp. 166–177. Springer, Cham (2019), https://doi.org/10.1007/978-3-030-26176-4_14
5. Epstein, D.: On the NP-completeness of cryptarithms. ACM SIGACT News **18**(3), 38–40 (1987), https://doi.org/10.1145/24658.24662
6. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989), https://doi.org/10.1137/0218012
7. Gradwohl, R., Naor, M., Pinkas, B., Rothblum, G.N.: Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. In: Crescenzi, P., Prencipe, G., Pucci, G. (eds.) Fun with Algorithms. LNCS, vol. 4475, pp. 166–182. Springer, Berlin, Heidelberg (2007), https://doi.org/10.1007/978-3-540-72914-3_16
8. 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), https://doi.org/10.1007/s00224-008-9119-9

9. Ishikawa, R., Chida, E., Mizuki, T.: Efficient card-based protocols for generating a hidden random permutation without fixed points. In: Calude, C.S., Dinneen, M.J. (eds.) Unconventional Computation and Natural Computation. LNCS, vol. 9252, pp. 215–226. Springer, Cham (2015), https://doi.org/10.1007/978-3-319-21819-9_16
10. Lafourcade, P., Miyahara, D., Mizuki, T., Robert, L., Sasaki, T., Sone, H.: How to construct physical zero-knowledge proofs for puzzles with a "single loop" condition. Theor. Comput. Sci. (2021, in press), https://doi.org/10.1016/j.tcs.2021.07.019
11. Lafourcade, P., Miyahara, D., Mizuki, T., Sasaki, T., Sone, H.: A physical ZKP for Slitherlink: How to perform physical topology-preserving computation. In: Heng, S.H., Lopez, J. (eds.) Information Security Practice and Experience. LNCS, vol. 11879, pp. 135–151. Springer, Cham (2019), https://doi.org/10.1007/978-3-030-34339-2_8
12. Manabe, Y., Ono, H.: Card-based cryptographic protocols for three-input functions using private operations. In: Combinatorial Algorithms. LNCS, vol. 12757, pp. 469–484. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-79987-8_33
13. Minhaz, A., Singh, A.V.: Solution of a classical cryptarithmetic problem by using parallel genetic algorithm. In: Reliability, Infocom Technologies and Optimization. pp. 1–5. IEEE (2014), https://doi.org/10.1109/ICRITO.2014.7014715
14. 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. Leibniz International Proceedings in Informatics (LIPIcs), vol. 157, pp. 20:1–20:21. Schloss Dagstuhl, Dagstuhl, Germany (2020), https://doi.org/10.4230/LIPIcs.FUN.2021.20
15. Miyahara, D., Sasaki, T., Mizuki, T., Sone, H.: Card-based physical zero-knowledge proof for Kakuro. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **102**(9), 1072–1078 (2019), https://doi.org/10.1587/transfun.E102.A.1072
16. Miyahara, D., Ueda, I., Hayashi, Y.i., Mizuki, T., Sone, H.: Analyzing execution time of card-based protocols. In: Stepney, S., Verlan, S. (eds.) Unconventional Computation and Natural Computation. LNCS, vol. 10867, pp. 145–158. Springer, Cham (2018), https://doi.org/10.1007/978-3-319-92435-9_11
17. Mizuki, T., Asiedu, I.K., Sone, H.: Voting with a logarithmic number of cards. In: Mauri, G., Dennunzio, A., Manzoni, L., Porreca, A.E. (eds.) Unconventional Computation and Natural Computation. LNCS, vol. 7956, pp. 162–173. Springer, Berlin, Heidelberg (2013), https://doi.org/10.1007/978-3-642-39074-6_16
18. Nakai, T., Misawa, Y., Tokushige, Y., Iwamoto, M., Ohta, K.: How to solve millionaires' problem with two kinds of cards. New Gener. Comput. **39**(1), 73–96 (2021), https://doi.org/10.1007/s00354-020-00118-8
19. Nozaki, Y., Hendrian, D., Yoshinaka, R., Shinohara, A.: Enumeration of cryptarithms using deterministic finite automata. In: Câmpeanu, C. (ed.) Implementation and Application of Automata. LNCS, vol. 10977, pp. 286–298. Springer, Cham (2018), https://doi.org/10.1007/978-3-319-94812-6_24
20. Ono, H., Manabe, Y.: Card-based cryptographic logical computations using private operations. New Gener. Comput. **39**(1), 19–40 (2021), https://doi.org/10.1007/s00354-020-00113-z
21. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Interactive physical zkp for connectivity: Applications to Nurikabe and Hitori. In: De Mol, L., Weiermann, A., Manea, F., Fernández-Duque, D. (eds.) Connecting with Computability. LNCS, vol. 12813, pp. 373–384. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-80049-9_37
22. Ruangwises, S.: An improved physical ZKP for Nonogram (2021), https://arxiv.org/abs/2106.14020
23. Ruangwises, S.: Two standard decks of playing cards are sufficient for a ZKP for Sudoku (2021), https://arxiv.org/abs/2106.13646

24. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for Numberlink. In: Farach-Colton, M., Prencipe, G., Uehara, R. (eds.) Fun with Algorithms. Leibniz International Proceedings in Informatics (LIPIcs), vol. 157, pp. 22:1–22:11. Schloss Dagstuhl, Dagstuhl, Germany (2020), https://doi.org/10.4230/LIPIcs.FUN.2021.22
25. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for Numberlink puzzle and k vertex-disjoint paths problem. New Gener. Comput. **39**(1), 3–17 (2021), https://doi.org/10.1007/s00354-020-00114-y
26. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for Ripple Effect. In: Hong, S., Nandy, S., Uehara, R. (eds.) WALCOM: Algorithms and Computation. LNCS, vol. 12635, pp. 296–307. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-68211-8_24
27. Ruangwises, S., Itoh, T.: Physical ZKP for connected spanning subgraph: Applications to Bridges Puzzle and other problems. In: Unconventional Computation and Natural Computation. LNCS, Springer, Cham (2021, to appear)
28. Sasaki, T., Miyahara, D., Mizuki, T., Sone, H.: Efficient card-based zero-knowledge proof for Sudoku. Theor. Comput. Sci. **839**, 135–142 (2020), https://doi.org/10.1016/j.tcs.2020.05.036
29. Sasaki, T., Mizuki, T., Sone, H.: Card-based zero-knowledge proof for Sudoku. In: Ito, H., Leonardi, S., Pagli, L., Prencipe, G. (eds.) Fun with Algorithms. Leibniz International Proceedings in Informatics (LIPIcs), vol. 100, pp. 29:1–29:10. Schloss Dagstuhl, Dagstuhl, Germany (2018), https://doi.org/10.4230/LIPIcs.FUN.2018.29
30. Shinagawa, K.: Card-based cryptography with invisible ink. In: Gopal, T., Watada, J. (eds.) Theory and Applications of Models of Computation. LNCS, vol. 11436, pp. 566–577. Springer, Cham (2019), https://doi.org/10.1007/978-3-030-14812-6_35
31. Shinagawa, K.: Card-based cryptography with dihedral symmetry. New Gener. Comput. **39**(1), 41–71 (2021), https://doi.org/10.1007/s00354-020-00117-9
32. Shinagawa, K., Mizuki, T., Schuldt, J., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Card-based protocols using regular polygon cards. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E100.A**(9), 1900–1909 (2017), https://doi.org/10.1587/transfun.E100.A.1900
33. Tamura, N.: Cryptarithmetic puzzle solver. https://tamura70.gitlab.io/web-puzzle/cryptarithm/, (Accessed on 03/21/2021)
34. Torsten, S.: Alphametics and cryptarithms. https://www.math.uni-bielefeld.de/~sillke/PUZZLES/ALPHAMETIC/, (Accessed on 03/21/2021)
35. Truman, C.: Alphametic puzzles. http://www.tkcs-collins.com/truman/alphamet/alphamet.shtml, (Accessed on 03/21/2021)
36. Yasunaga, K.: Practical card-based protocol for three-input majority. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E103.A**(11), 1296–1298 (2020), https://doi.org/10.1587/transfun.2020EAL2025