Gakmoro: An Application of Physical Secure Computation to Card Game[⋆]

Takaaki Mizuki[1], Tomoki Kuzuma[2], Tomoya Hirano[3], Ririn Oshima[4], and Momofuku Yasuda[4]

[1] Cyberscience Center, Tohoku University, Sendai, Japan
mizuki+lncs[atmark]tohoku.ac.jp
[2] Graduate School of Information Sciences, Tohoku University, Sendai, Japan
[3] School of Engineering, Tohoku University, Sendai, Japan
[4] Faculty of Agriculture, Tohoku University, Sendai, Japan

Abstract. We have created a new card game, Gakmoro, where two players each have seven cards numbered from 1 to 7. In each round, they secretly choose one to three cards to compete based on their total value; the first player to win two rounds wins the game. The unique feature of Gakmoro is that the cards chosen by the players must remain secret, and only the winner of each round is revealed. This secrecy is expected to add depth to the game by emphasizing the strategy of reading an opponent's hand. As designed, the game requires a dealer to secretly calculate the sum of the cards and announce only the winner. In this paper, instead of relying on a human dealer, we use card-based cryptography to virtually fulfill the dealer's role. In other words, we design an 'unconventional' computation protocol that allows players to securely determine the winner without a dealer using a physical deck of cards.

Keywords: Card-based cryptography · Secure computation · Playing cards · Card games.

1 Introduction

We introduce a new card game, *Gakmoro*, designed by students (the third, fourth, and fifth authors) and a teaching assistant (the second author) during an "Introduction to Academic Learning" class at Tohoku University. Its name, Gakmoro, is derived from the Japanese name of the class, "Gakumonron Enshu."

[⋆] This paper appears in Proceedings of UCNC 2025. This version of the contribution has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record will be available online soon. Use of this Accepted Version is subject to the publisher ' s Accepted Manuscript terms of use: https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms.

1.1 Gakmoro's Rules

The game, Gakmoro, involves two players, Alice and Bob, together with a dealer. First, each of the two players has seven cards with numbers from 1 to 7. For example, Alice holds seven cards $\boxed{1\spadesuit}\boxed{2\spadesuit}\boxed{3\spadesuit}\boxed{4\spadesuit}\boxed{5\spadesuit}\boxed{6\spadesuit}\boxed{7\spadesuit}$ in her hand, and Bob holds seven cards $\boxed{1\diamondsuit}\boxed{2\diamondsuit}\boxed{3\diamondsuit}\boxed{4\diamondsuit}\boxed{5\diamondsuit}\boxed{6\diamondsuit}\boxed{7\diamondsuit}$ in his hand. The game consists of up to three rounds. The first player to win two rounds wins the game. Each round proceeds as follows.

1. Each player selects one to three cards from his or her hand and submits them secretly to the dealer. (Neither the number of cards nor the card values are revealed to the opponent.)
2. The dealer calculates the sum of the card values submitted by each player. The player with the higher total wins the round. The dealer announces only the winner's name or "a tie" (while keeping the sum, the number of cards, and the card values secret).

Of course, once a card is submitted to the dealer, it cannot be used in subsequent rounds. As mentioned, the first player to achieve two wins is the winner of the game.

1.2 Example of Game Progression

Here is an example of how a game of Gakmoro might unfold.

In the first round, Alice secretly plays $\boxed{6\spadesuit}$ and $\boxed{7\spadesuit}$ (totaling 13), while Bob secretly plays $\boxed{1\diamondsuit}$ and $\boxed{2\diamondsuit}$ (totaling 3). Alice's total is higher, so she wins the round. The dealer only announces, "Alice wins the round."

In the second round, Alice plays $\boxed{1\spadesuit}$ and $\boxed{2\spadesuit}$ (totaling 3), aiming to save her higher cards for a potential third round. Bob, anticipating this, plays his strong cards: $\boxed{5\diamondsuit}$, $\boxed{6\diamondsuit}$, and $\boxed{7\diamondsuit}$ (totaling 18). Bob's total is higher, so he wins the round. Only "Bob wins the round" is announced.

In the third round, they play their remaining cards. Alice plays $\boxed{3\spadesuit}$, $\boxed{4\spadesuit}$, and $\boxed{5\spadesuit}$ (totaling 12). Bob plays $\boxed{3\diamondsuit}$ and $\boxed{4\diamondsuit}$ (totaling 7). Alice's total is higher, so she wins the round and, consequently, she is the winner of the game.

1.3 Playing Without a Dealer

The dealer's role in Gakmoro is crucial for maintaining secrecy and ensuring fair play. The main functions are:

1. Ensuring that each player submits between one and three cards.
2. Correctly adding the values of the submitted cards.
3. Announcing only the winner, not the sums or the number of cards played.

Without a dealer, the game loses its core element of hidden information, which is central to strategic play. The excitement and strategic depth come from the uncertainty of what the opponent has played.

Thus, the role of a dealer is indispensable. What if there are only two players, making it difficult to find a dealer? Are there any solutions for playing Gakmoro with just two people?

1.4 Application of Secure Computations

To the questions raised above, one might think of using *secure computation* [51] to fulfill the dealer's role in Gakmoro. Secure computation is a cryptographic technique that produces the output of a predetermined function while keeping the input information secret. Typically, secure computation protocols are implemented on computers over networks.

However, Gakmoro is envisioned as a casual card game played with physical cards, so relying on computers or networks is not ideal, and we solicit 'unconventional' computing that relies only on daily objects. Therefore, we will use *card-based cryptography* [15, 27], which enables secure computation using a physical deck of cards.

1.5 Contribution of This Paper

In this paper, we aim to eliminate the need for a human dealer in Gakmoro by combining existing techniques from card-based cryptography and designing a new protocol.

Card-based cryptography typically uses two types of cards, such as $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\cdots$ and $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}\cdots$, whose backs are identical $\boxed{?}\boxed{?}\boxed{?}\cdots$ (cf. [3, 19, 46]). A pair of cards is often used to represent a bit value, i.e., $\boxed{\clubsuit}\boxed{\heartsuit} = 0$, $\boxed{\heartsuit}\boxed{\clubsuit} = 1$. Under such encoding, it is theoretically possible to perform secure computation of addition by representing Alice and Bob's cards in binary numbers, as protocols for secure computations of the half-adder and the full-adder are known [25]. However, it is not user-friendly to handle binary arithmetic in a game intended for both children and adults.

Thus, instead, we will use an encoding that expresses integers by the position of $\boxed{\heartsuit}$, as in the following example (details are given in Section 2.2):

$$\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}.$$

Under such integer encoding, Ruangwises and Itoh [38] have proposed a protocol to realize secure computation of addition (details are given in Section 2.4).

In this paper, we use their protocol to sum the cards submitted by each player. Recalling Gakmoro's rules, besides addition, the other necessary operation is comparison between two integers (to determine the winner). In card-based cryptography, several methods for secure computation of comparison (on integers) are known [21, 31], but in this paper, we propose a new comparison protocol (Section 3). By combining the above, we show that the dealer's role in Gakmoro can be achieved by a card-based protocol (Section 4).

1.6 Related Studies

Card-based cryptography has made remarkable progress in recent years. Most previous studies on card-based cryptography have focused on protocols for secure computations (e.g., [4, 39, 49, 52]) and zero-knowledge proofs (e.g., [17, 37, 48]), with few studies applying card-based cryptography to card games. To the best of the authors' knowledge, the first such attempt was to create virtual players in a famous card game, Old Maid [42], followed by the work for UNO games [40].

Other examples of applications to games (not necessalily card games) include: a protocol that can secretly form multiple player groups for games like Werewolf [8]; a protocol that can generate only solvable problems for games like the 15-puzzle and Rubik's Cube [41]; a protocol for two-player games that allows players to secretly determine who goes first based on their private preferences [44]; and protocols for generating a random derangement to enjoy secret Santa [3,11,29]. In addition, as implied above, numerous card-based zero-knowledge proof protocols exist, primarily conceived for puzzles and games, such as Topswops [18], the 15-puzzle [47], Moon-or-Sun [7], Usowan [23], Sumplete [9], Dosun-Fuwari [14], and Nurimisaki [36].

1.7 Organization of This Paper

The rest of this paper is organized as follows. Section 2 describes the cards to be used, the integer encoding, and the existing protocol for addition. Section 3 provides the subtraction protocol and the method of secure computation of comparison using it. In Section 4, we describe the proposed protocol for playing Gakmoro without a dealer. The implementation method is discussed in Section 5. The conclusion is given in Section 6.

2 Preliminaries

This section provides the necessary preparation for describing and explaining the proposed method. Specifically, we first describe the cards to be used, the encoding of integers, and the shuffling operation, followed by an introduction to the existing protocol for integer addition.

2.1 Two-Color Deck of Cards

As we saw in Section 1, Gakmoro itself is a game played with playing cards. However, since it is not easy to use playing cards to construct secure computation protocols for addition or comparison of the numbers written on the playing cards, we use a *two-color deck* of cards, as is typically used in card-based cryptography.

As briefly introduced in Section 1.5, we use two types of cards, namely, black cards $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\cdots$ and red cards $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}\cdots$, whose backs $\boxed{?}\boxed{?}\boxed{?}\cdots$ are all indistinguishable. In the field of card-based cryptography, many research papers have been published using such a two-color deck of cards, and some practical representative protocols for secure computation of logical products exist, such as the five-card trick [1] and MS-AND protocol [28].

In fact, although it is possible to construct secure computation protocols with playing cards (e.g., [6,16,24,32]), it is generally believed to be more difficult to construct a simple protocol compared to the two-color-deck case (since all playing cards are 'unique,' it is generally more challenging to keep the input secret). It is a future prospect to develop an efficient method that can be implemented on playing cards,[5] which are more readily available.

2.2 Encoding of Integers

Gakmoro requires integers from 1 to 7 as inputs, and their addition and comparison. This paper employs the following encoding of integers using a two-color deck of cards. Suppose that we fix a positive integer m and wish to deal with integers from 0 to m.

Using m $\boxed{\clubsuit}$s and one $\boxed{\heartsuit}$, setting the $(i+1)$-th card to $\boxed{\heartsuit}$ denotes the integer $i \in \{0, 1, \ldots, m\}$:

$$\begin{array}{ccccccccl}
0 & 1 & 2 & & i & i+1 & & m & \\
\boxed{\heartsuit} & \boxed{\clubsuit} & \boxed{\clubsuit} & \cdots & \boxed{\clubsuit} & \boxed{\clubsuit} & \cdots & \boxed{\clubsuit} & = 0 \\
\boxed{\clubsuit} & \boxed{\heartsuit} & \boxed{\clubsuit} & \cdots & \boxed{\clubsuit} & \boxed{\clubsuit} & \cdots & \boxed{\clubsuit} & = 1 \\
 & & & & & & & & \vdots \\
\boxed{\clubsuit} & \boxed{\clubsuit} & \boxed{\clubsuit} & \cdots & \boxed{\heartsuit} & \boxed{\clubsuit} & \cdots & \boxed{\clubsuit} & = i \\
 & & & & & & & & \vdots \\
\boxed{\clubsuit} & \boxed{\clubsuit} & \boxed{\clubsuit} & \cdots & \boxed{\clubsuit} & \boxed{\clubsuit} & \cdots & \boxed{\heartsuit} & = m.
\end{array}$$

In the following, we write $E^{\heartsuit}_{m+1}(i)$ as a bundle of cards representing an integer i placed face down under this encoding.

This encoding was first introduced in Crépeau and Kilian's secure derangement generation [3], and then has been used in Bultel et al.'s zero-knowledge proof protocol [2], Ruangwises and Itoh's protocols [38], and so on.

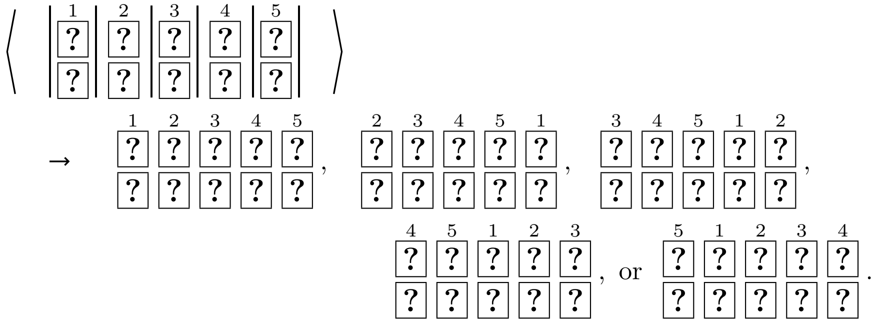
2.3 Pile-Shifting Shuffle

The shuffling action used in our proposed method is the *pile-shifting shuffle* [33,43][6]. This is an operation that considers a bundle of cards of the same size as a *pile* and shuffles the piles in a cyclic manner. The number of shifts is random, and no one knows the number of shifts. This shuffling can be implemented securely by putting each pile of cards in a sleeve (or a card case [13]) and using a technique called Hindu cut [50].

[5] The "partial opening" action [10, 22] on playing cards is considered a promising technique.

[6] Note that this is different from the pile-scramble shuffle [11].

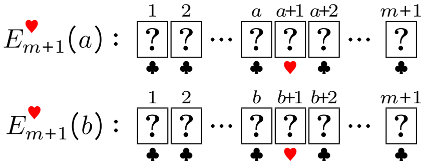
As an example, applying a pile-shifting shuffle to the following five (two-card) piles yields one of the following five sequences each with a probability of $1/5$:



2.4 Addition Protocol

Ruangwises and Itoh [38] constructed a protocol that, given two integers based on the encoding given in Section 2.2, securely computes their sum. Since this existing protocol is used in this paper, the protocol procedure is described below.

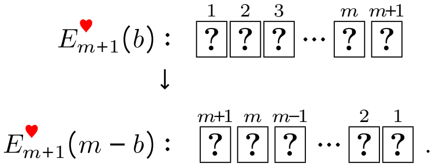
Let m be fixed, $a, b \in \{0, 1, \ldots, m\}$, and suppose that bundles of cards corresponding to integers a and b are given face down, that is,



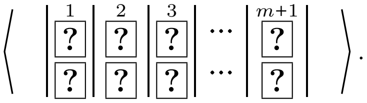
is the input (a total of $2m + 2$ cards). Recall that the positions of $\boxed{\heartsuit}$ determine the values of a and b.

The addition protocol [38] takes $E^{\heartsuit}_{m+1}(a)$ and $E^{\heartsuit}_{m+1}(b)$ as input and works as follows.

1. Rearrange the cards in $E^{\heartsuit}_{m+1}(b)$ such that the left and right sides are reversed. This yields $E^{\heartsuit}_{m+1}(m - b)$:



2. $E^{\heartsuit}_{m+1}(a)$ and $E^{\heartsuit}_{m+1}(m - b)$ are arranged in two rows, with one card on top and one card on the bottom, and a pile-shifting shuffle is applied to the piles:



In this case, a random shift $r \in \{0, 1, \dots, m\}$ arises, so that we obtain $E^{\heartsuit}_{m+1}(a + r \bmod m + 1)$ and $E^{\heartsuit}_{m+1}(m - b + r \bmod m + 1)$.

3. Turn over all the cards in the bottom row, i.e., $E^{\heartsuit}_{m+1}(m - b + r \bmod m + 1)$, and shift the top and bottom cards in piles until $\boxed{\heartsuit}$ is at the right edge:

$$\begin{array}{c}
\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}\\
\boxed{\clubsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}\boxed{\clubsuit}\\
\downarrow\\
\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}\\
\boxed{\clubsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}\boxed{\heartsuit}.
\end{array}$$

This results in the upper row being shifted by

$$m - (m - b + r \bmod m + 1) = b - r \bmod m + 1$$

positions to the right. Therefore, the resulting bundle from $E^{\heartsuit}_{m+1}(a + r \bmod m + 1)$ after this shift will be $E^{\heartsuit}_{m+1}((a + r) + (b - r) \bmod m + 1)$. Thus, $E^{\heartsuit}_{m+1}(a + b \bmod m + 1)$ is obtained.

The above is the existing addition protocol, which can generate the addition result $E^{\heartsuit}_{m+1}(a + b \bmod m + 1)$ from $E^{\heartsuit}_{m+1}(a)$ and $E^{\heartsuit}_{m+1}(b)$ without leaking their values.

3 Subtraction Protocol and Comparison Protocol

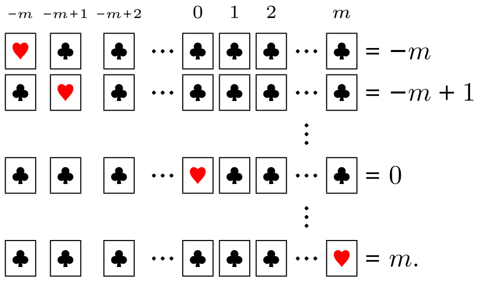
In this section, we construct a card-based protocol for secure comparison between integers in order to perform the role of Gakmoro's dealer.

The protocol design policy is that, when integers a and b are to be compared, the subtraction $a - b$ is obtained, and the sign (positive, negative, or zero) of this value is used to determine whether $a > b$, $a < b$, or $a = b$. For this reason, we first construct a subtraction protocol in Section 3.1, referring to the addition protocol introduced in Section 2.4. Next, in Section 3.2, we propose a comparison protocol by making use of the subtraction protocol.

3.1 Subtraction Protocol

The subtraction protocol proposed here takes $E^{\heartsuit}_{m+1}(a)$ and $E^{\heartsuit}_{m+1}(b)$ as input and aims to output the difference $a - b$ (note that this is not $a - b \bmod m + 1$). Therefore, since we need to deal with negative integers, we extend the encoding a little. Fix a positive integer m and represent the integers from $-m$ to m using

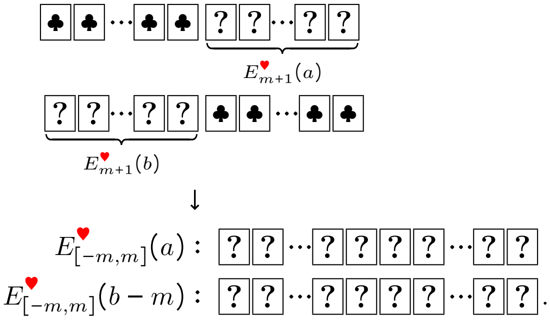
$2m+1$ cards as follows:



Under such an encoding, when an integer i such that $-m \le i \le m$ is represented by $2m+1$ cards placed face down, we write $E^{\heartsuit}_{[-m,m]}(i)$ for the bundle of cards. (A similar encoding is used in the literature [9].)

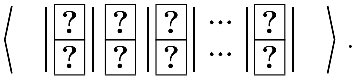
In the following, given $E^{\heartsuit}_{m+1}(a)$ and $E^{\heartsuit}_{m+1}(b)$ along with $2m$ additional black cards, we show a protocol for generating $E^{\heartsuit}_{[-m,m]}(a-b)$.

1. Place $2m$ $\boxed{\clubsuit}$s face down as follows; this yields $E^{\heartsuit}_{[-m,m]}(a)$ and $E^{\heartsuit}_{[-m,m]}(b-m)$:


$E^{\heartsuit}_{m+1}(a)$
$E^{\heartsuit}_{m+1}(b)$
$E^{\heartsuit}_{[-m,m]}(a)$:
$E^{\heartsuit}_{[-m,m]}(b-m)$:

Note that if we shift the lower row together with the upper row so that $\boxed{\heartsuit}$ in the lower row is at the left end, the whole is shifted b positions to the left and the value of the upper row becomes $a-b$, yielding $E^{\heartsuit}_{[-m,m]}(a-b)$. Since we cannot just reveal the lower row, we will apply a shuffle in the next step.

2. Apply a pile-shifting shuffle to the two rows:



3. Turn over all the cards in the bottom row and shift the cards in the upper and lower cards in piles until $\boxed{\heartsuit}$ is at the left edge:

$$\begin{array}{c}
\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}\\
\boxed{\clubsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}\boxed{\clubsuit}\\
\downarrow\\
\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}\\
\boxed{\heartsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\cdots\boxed{\clubsuit}\boxed{\clubsuit}\ .
\end{array}$$

In this case, the upper row becomes $E^{\heartsuit}_{[-m,m]}(a-b)$.

3.2 Secure Computation of Comparison

Suppose that we want to compare $E^{\heartsuit}_{m+1}(a)$ and $E^{\heartsuit}_{m+1}(b)$. With the subtraction protocol constructed in Section 3.1, we can obtain

$$E^{\heartsuit}_{[-m,m]}(a-b):\ \boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?},$$

and hence, if we reveal this bundle of cards to disclose its value, we can of course determine whether $a > b$, $a < b$, or $a = b$. However, this should be prohibited for Gakmoro, because if the value of $a - b$ is disclosed, the opponent's value can be calculated.

Therefore, shuffling is performed for the following two locations in order to obtain only the result of a comparison:

$$\underbrace{\overset{-m}{\boxed{?}}\ \overset{-m+1}{\boxed{?}}\ \cdots\overset{-1}{\boxed{?}}}_{\text{shuffling}}\ \overset{0}{\boxed{?}}\ \underbrace{\overset{1}{\boxed{?}}\ \cdots\ \overset{m-1}{\boxed{?}}\ \overset{m}{\boxed{?}}}_{\text{shuffling}}\ .$$

Then, all cards are turned over, and the position of $\boxed{\heartsuit}$, which is present only once, yields the result of the comparison:

- If $\boxed{\heartsuit}$ appears in the center, then $a = b$;
- If $\boxed{\heartsuit}$ appears on the left, then $a < b$; and
- If $\boxed{\heartsuit}$ appears to the right, then $a > b$.

This is our comparison protocol using the subtraction protocol.

The proposed method is slightly similar to Nuida's "Tug-of-War" technique [34] and also slightly similar to the additive protocol using counters and dummies by Hatsugai et al. [9]. Card-based arithmetic operations were discussed in [35], and those for other kinds of cards appear in [12, 45].

4 Gakmoro with Secure Computation

In this section, we describe a card-based protocol for playing Gakmoro with only two players (Alice and Bob) by combining the protocols described and constructed thus far.

Remember that in Gakmoro, each player holds seven cards and submits one to three cards at every round. Let us add two cards of value 0 to each player's hand. Then, we can assume that each player, holding nine cards (namely, those of 0, 0, 1, 2, 3, 4, 5, 6, and 7) at the beginning, submits exactly three cards at every round (because a card of value 0 can serve as a dummy card). Thus, in the sequel, we assume this version of Gakmoro.

Our protocol for playing Gakmoro with secure computations proceeds as follows.

1. For each of Alice and Bob, create nine bundles of cards corresponding to 0, 0, 1, 2, 3, 4, 5, 6, and 7:
$$E_8^{\heartsuit}(0), E_8^{\heartsuit}(0), E_8^{\heartsuit}(1), E_8^{\heartsuit}(2), E_8^{\heartsuit}(3), E_8^{\heartsuit}(4), E_8^{\heartsuit}(5), E_8^{\heartsuit}(6), E_8^{\heartsuit}(7)\,.$$
These nine bundles are regarded as his or her hand.
2. Each player chooses three bundles from their hand and places them face down on the table.
3. Let $E_8^{\heartsuit}(a_1), E_8^{\heartsuit}(a_2), E_8^{\heartsuit}(a_3)$ be the three bundles placed by Alice. The addition protocol described in Section 2.4 is applied to these to compute the total value. In this case, the modulus for addition needs to be expanded. Specifically, when $E_8^{\heartsuit}(a_1)$ and $E_8^{\heartsuit}(a_2)$ are first added, the maximum possible sum is $7+6=13$. Therefore, additional cards are prepared, and $E_{14}^{\heartsuit}(a_1)$ and $E_{14}^{\heartsuit}(a_2)$ are used as input to obtain the result of addition $E_{14}^{\heartsuit}(a_1+a_2)$. Next, we want to add $E_{14}^{\heartsuit}(a_1+a_2)$ and $E_8^{\heartsuit}(a_3)$. Considering that the maximum possible total value is $7+6+5=18$, we can use $E_{19}^{\heartsuit}(a_1+a_2)$ and $E_{19}^{\heartsuit}(a_3)$ as input to obtain the result of addition $E_{19}^{\heartsuit}(a_1+a_2+a_3)$. In the same way, we obtain Bob's sum $E_{19}^{\heartsuit}(b_1+b_2+b_3)$.
4. Perform a comparison of the players' sums, $E_{19}^{\heartsuit}(a_1+a_2+a_3)$ and $E_{19}^{\heartsuit}(b_1+b_2+b_3)$, using the comparison protocol described in Section 3.2, to obtain only the winner's name (or a draw). After obtaining the result of a win or loss, the face-up cards are used as additional cards for the next and subsequent rounds.
5. Step 2 to Step 4 are repeated for up to three rounds, with the first player to win two rounds being declared the winner.

5 Implementation Considerations

In this section, we discuss points to note, implementation methods, and physical space requirements when actually using the methods presented in Section 4 in Gakmoro.

5.1 Notes on Input and How to Implement

We proposed a method to enjoy Gakmoro with only two players in Section 4. The protocol for outputting the win/loss at each round conforms to the standard computational model for card-based protocols [26],[7] where both players input $E_8^{\heartsuit}(a_1), E_8^{\heartsuit}(a_2), E_8^{\heartsuit}(a_3)$ and $E_8^{\heartsuit}(b_1), E_8^{\heartsuit}(b_2), E_8^{\heartsuit}(b_3)$. After inputting, the protocol can be executed by anyone, and the correct winner is always output.

However, when actually applying and implementing this in Gakmoro, care must be taken as to how each player prepares an input. For example, if we simply give each player 63 $\boxed{\clubsuit}$s and 9 $\boxed{\heartsuit}$s to make two $E_8^{\heartsuit}(0)$ bundles and $E_8^{\heartsuit}(1)$ through $E_8^{\heartsuit}(7)$ bundles, a malicious player might create all nine bundles as $E_8^{\heartsuit}(7)$[8].

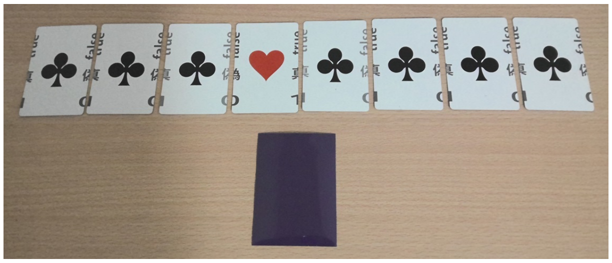


Fig. 1. Preparation for making $E_8^{\heartsuit}(3)$.

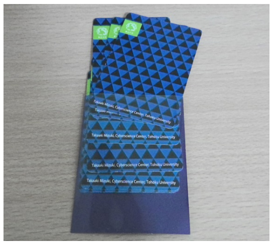


Fig. 2. The cards are placed in a sleeve.

The following implementation method is possible. First, Alice and Bob cooperatively and publicly make two $E_8^{\heartsuit}(0)$ and $E_8^{\heartsuit}(1)$ through $E_8^{\heartsuit}(7)$. Each bundle of cards is then placed in a sleeve, which is often used in trading card games. Figure 1 shows a photograph of the making of $E_8^{\heartsuit}(3)$ and the sleeve. Here, a sleeve with a transparent front and an opaque backside is used. Figure 2 shows how eight cards are placed in a sleeve.

After placing two $E_8^{\heartsuit}(0)$ and one each of $E_8^{\heartsuit}(1)$ through $E_8^{\heartsuit}(7)$ in their respective sleeves, place a sticky note on the transparent surface of each sleeve so that the integer values inside can be easily identified. Such a picture is shown in Figure 3.

Two sets of sleeves made in this way are prepared, one for Alice and one for Bob. The back of the sleeve is opaque, so each player holds the sleeve in his or her hand with the back facing the other player and the sticky note side facing him or her, and make up the hand in units of sleeves, similar to how one

[7] There is another computational model (cf. [20, 30]).

[8] This is unlikely to happen in a normal secure computation where players input their own secrets and try to obtain a meaningful output.

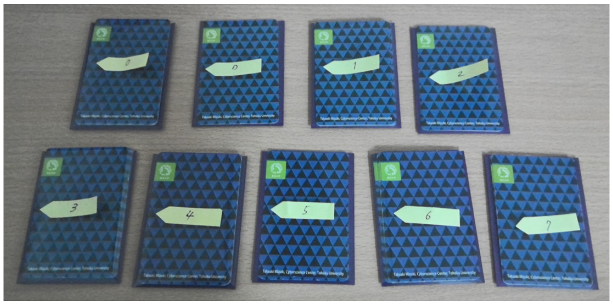


Fig. 3. Sleeves with numbers.

plays Gakmoro with regular playing cards. In this case, each player only needs to choose three sleeves for each round (while possibly shuffling the sleeves).

When each player submits a sleeve from his or her hand, he or she peels off the sticky note before placing it on the table (of course, the peeled-off sticky note should not be visible to the opponent). Then, the player simply removes the cards face down from the sleeve, and the protocol is executed.

5.2 Space Requirements in Implementation

This subsection addresses practical (physical) space requirements encountered during implementation.

Recall that when each player performs the addition protocol on the three sleeves, they first apply the addition protocol to two bundles of cards, then apply it again to the result and the remaining bundle. This process is performed twice. As described in Section 2.4, the first addition uses $14 \times 2 = 28$ cards and the second addition uses $19 \times 2 = 38$ cards. In the comparison protocol, $(19 + 18) \times 2 = 74$ cards will be used.

To perform these operations, we require enough space to place 74 cards total, arranged in a 2-row by 37-column grid. This space requirement could be a challenge in implementation. Furthermore, for the game to run smoothly, it is desirable to have space for six cards vertically, allowing players to arrange the bundles of cards they submitted vertically.

6 Conclusion

In this paper, we proposed a new card game called "Gakmoro" that typically requires a dealer. We then presented a method that utilizes secure computation to allow two players to enjoy this game without a dealer.

We have demonstrated that leveraging secure computations can increase the flexibility of game play. If secure computation can be applied not only to Gakmoro but also to various other card games, it is expected that a wider range of people will be exposed to the fascinating world of cryptography. Furthermore, such applications could be useful for education in terms of gamification (cf. [5]).

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 JP24K02938 and JP23H00479.

References

1. Boer, B.D.: More efficient match-making and satisfiability the five card trick. In: Quisquater, J.J., Vandewalle, J. (eds.) Advances in Cryptology – EUROCRYPT' 89. LNCS, vol. 434, pp. 208–217. Springer, Heidelberg (1990), https://doi.org/10.1007/3-540-46885-4_23
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: Izumi, T., Kuznetsov, P. (eds.) Stabilization, Safety, and Security of Distributed Systems. LNCS, vol. 11201, pp. 111–125. Springer (2018), https://doi.org/10.1007/978-3-030-03232-6_8
3. Crépeau, C., Kilian, J.: Discreet solitary games. In: Stinson, D.R. (ed.) Advances in Cryptology—CRYPTO' 93. LNCS, vol. 773, pp. 319–330. Springer, Berlin, Heidelberg (1994), https://doi.org/10.1007/3-540-48329-2_27
4. Doi, A., Ono, T., Abe, Y., Nakai, T., Shinagawa, K., Watanabe, Y., Nuida, K., Iwamoto, M.: Card-based protocols for private set intersection and union. New Gener. Comput. **42**, 359–380 (2024), https://doi.org/10.1007/s00354-024-00268-z
5. Gong, X., Xu, W., Yu, S., Ma, J., Qiao, A.: Enhancing computational thinking and spatial reasoning skills in gamification programming learning: A comparative study of tangible, block and paper-and-pencil tools. British Journal of Educational Technology **56**(1), 80–102 (2025). https://doi.org/10.1111/bjet.13482
6. Haga, R., Hayashi, Y., Miyahara, D., Mizuki, T.: Card-minimal protocols for three-input functions with standard playing cards. In: Batina, L., Daemen, J. (eds.) AFRICACRYPT 2022. LNCS, vol. 13503, pp. 448–468. Springer, Cham (2022), https://doi.org/10.1007/978-3-031-17433-9_19
7. Hand, S., Koch, A., Lafourcade, P., Miyahara, D., Robert, L.: Efficient card-based ZKP for single loop condition and its application to Moon-or-Sun. New Gener. Comput. **42**, 449–477 (2024), https://doi.org/10.1007/s00354-024-00274-1
8. Hashimoto, Y., Shinagawa, K., Nuida, K., Inamura, M., Hanaoka, G.: Secure grouping protocol using a deck of cards. IEICE Trans. Fundam. **E101.A**(9), 1512–1524 (2018), https://doi.org/10.1587/transfun.E101.A.1512
9. Hatsugai, K., Ruangwises, S., Asano, K., Abe, Y.: NP-completeness and physical zero-knowledge proofs for Sumplete, a puzzle generated by ChatGPT. New Gener. Comput. **42**, 429–448 (2024), https://doi.org/10.1007/s00354-024-00267-0
10. Honda, Y., Shinagawa, K.: Efficient card-based protocols with a standard deck of playing cards using partial opening. In: Minematsu, K., Mimura, M. (eds.) Advances in Information and Computer Security. LNCS, vol. 14977, pp. 85–100. Springer, Singapore (2024), https://doi.org/10.1007/978-981-97-7737-2_5
11. 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

12. Isuzugawa, R., Miyahara, D., Mizuki, T.: Zero-knowledge proof protocol for Cryptarithmetic using dihedral cards. In: Kostitsyna, I., Orponen, P. (eds.) Unconventional Computation and Natural Computation. LNCS, vol. 12984, pp. 51–67. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-87993-8_4
13. Ito, Y., Shikata, H., Suganuma, T., Mizuki, T.: Card-based cryptography meets 3D printer. In: Cho, D.J., Kim, J. (eds.) Unconventional Computation and Natural Computation. LNCS, vol. 14776, pp. 74–88. Springer, Cham (2024), https://doi.org/10.1007/978-3-031-63742-1_6
14. Iwamoto, C., Ohara, K.: Card-based zero-knowledge proof for Dosun-Fuwari. IEICE Trans. Fundam. (2025), https://doi.org/10.1587/transfun.2024DML0001
15. Koch, A.: The landscape of security from physical assumptions. In: IEEE Information Theory Workshop. pp. 1–6. IEEE, NY (2021), https://doi.org/10.1109/ITW48936.2021.9611501
16. Koch, A., Schrempp, M., Kirsten, M.: Card-based cryptography meets formal verification. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology–ASIACRYPT 2019. LNCS, vol. 11921, pp. 488–517. Springer, Cham (2019), https://doi.org/10.1007/978-3-030-34578-5_18
17. Komano, Y., Mizuki, T.: Card-based zero-knowledge proof protocol for pancake sorting. In: Bella, G., Doinea, M., Janicke, H. (eds.) Innovative Security Solutions for Information Technology and Communications. LNCS, vol. 13809, pp. 222–239. Springer, Cham (2023), https://doi.org/10.1007/978-3-031-32636-3_13
18. Komano, Y., Mizuki, T.: Physical zero-knowledge proof protocols for Topswops and Botdrops. New Gener. Comput. **42**, 399–428 (2024), https://doi.org/10.1007/s00354-024-00272-3
19. Koyama, H., Miyahara, D., Mizuki, T., Sone, H.: A secure three-input AND protocol with a standard deck of minimal cards. In: Santhanam, R., Musatov, D. (eds.) Computer Science – Theory and Applications. LNCS, vol. 12730, pp. 242–256. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-79416-3_14
20. Manabe, Y., Ono, H.: Card-based cryptographic protocols with a standard deck of cards using private operations. New Gener. Comput. **42**, 305–329 (2024), https://doi.org/10.1007/s00354-024-00257-2
21. Miyahara, D., Hayashi, Y., Mizuki, T., Sone, H.: Practical card-based implementations of Yao's millionaire protocol. Theor. Comput. Sci. **803**, 207–221 (2020), https://doi.org/10.1016/j.tcs.2019.11.005
22. Miyahara, D., Mizuki, T.: Secure computations through checking suits of playing cards. In: Li, M., Sun, X. (eds.) Frontiers in Algorithmics. LNCS, vol. 13461, pp. 110–128. Springer, Cham (2023), https://doi.org/10.1007/978-3-031-20796-9_9
23. Miyahara, D., Robert, L., Lafourcade, P., Mizuki, T.: ZKP protocols for Usowan, Herugolf, and Five Cells. Tsinghua Science and Technology **29**(6), 1651–1666 (2024), https://doi.org/10.26599/TST.2023.9010153
24. Mizuki, T.: Efficient and secure multiparty computations using a standard deck of playing cards. In: Foresti, S., Persiano, G. (eds.) Cryptology and Network Security. LNCS, vol. 10052, pp. 484–499. Springer, Cham (2016), https://doi.org/10.1007/978-3-319-48965-0_29
25. 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

26. Mizuki, T., Shizuya, H.: A formalization of card-based cryptographic protocols via abstract machine. Int. J. Inf. Secur. **13**(1), 15–23 (2014), https://doi.org/10.1007/s10207-013-0219-4
27. Mizuki, T., Shizuya, H.: Computational model of card-based cryptographic protocols and its applications. IEICE Trans. Fundam. **E100.A**(1), 3–11 (2017), https://doi.org/10.1587/transfun.E100.A.3
28. Mizuki, T., Sone, H.: Six-card secure AND and four-card secure XOR. In: Deng, X., Hopcroft, J.E., Xue, J. (eds.) Frontiers in Algorithmics. LNCS, vol. 5598, pp. 358–369. Springer, Berlin, Heidelberg (2009), https://doi.org/10.1007/978-3-642-02270-8_36
29. Murata, S., Miyahara, D., Mizuki, T., Sone, H.: Efficient generation of a card-based uniformly distributed random derangement. In: Uehara, R., Hong, S.H., Nandy, S.C. (eds.) WALCOM: Algorithms and Computation. LNCS, vol. 12635, pp. 78–89. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-68211-8_7
30. Nakai, T., Iwanari, K., Ono, T., Abe, Y., Watanabe, Y., Iwamoto, M.: Card-based cryptography with a standard deck of cards, revisited: Efficient protocols in the private model. New Gener. Comput. **42**, 345–358 (2024), https://doi.org/10.1007/s00354-024-00269-y
31. Nakai, T., Tokushige, Y., Misawa, Y., Iwamoto, M., Ohta, K.: Efficient card-based cryptographic protocols for millionaires' problem utilizing private permutations. In: Foresti, S., Persiano, G. (eds.) Cryptology and Network Security. LNCS, vol. 10052, pp. 500–517. Springer, Cham (2016), https://doi.org/10.1007/978-3-319-48965-0_30
32. Niemi, V., Renvall, A.: Solitaire zero-knowledge. Fundam. Inf. **38**(1,2), 181–188 (1999), https://doi.org/10.3233/FI-1999-381214
33. Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: Pile-shifting scramble for card-based protocols. IEICE Trans. Fundam. **101**(9), 1494–1502 (2018), https://doi.org/10.1587/transfun.E101.A.1494
34. Nuida, K.: Efficient card-based Millionaires' protocols via non-binary input encoding. In: Shikata, J., Kuzuno, H. (eds.) Advances in Information and Computer Security. LNCS, vol. 14128, pp. 237–254. Springer, Cham (2023), https://doi.org/10.1007/978-3-031-41326-1_13
35. Odaka, S., Komano, Y.: Card-based arithmetic operations and application to statistical data aggregation. In: Innovative Security Solutions for Information Technology and Communications (SecITC 2024). LNCS, vol. 15595. Springer, Cham (2025, to appear)
36. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Physical ZKP protocols for Nurimisaki and Kurodoko. Theor. Comput. Sci. **972**, 114071 (2023), https://doi.org/10.1016/j.tcs.2023.114071
37. Ruangwises, S., Itoh, T.: Physical ZKP for connected spanning subgraph: Applications to Bridges puzzle and other problems. In: Kostitsyna, I., Orponen, P. (eds.) Unconventional Computation and Natural Computation. pp. 149–163. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-87993-8_10
38. Ruangwises, S., Itoh, T.: Securely computing the n-variable equality function with $2n$ cards. Theor. Comput. Sci. **887**, 99–110 (2021), https://doi.org/10.1016/j.tcs.2021.07.007
39. Ruangwises, S., Ono, T., Abe, Y., Hatsugai, K., Iwamoto, M.: Card-based overwriting protocol for equality function and applications. In: Cho, D.J., Kim, J. (eds.) Unconventional Computation and Natural Computation. LNCS, vol. 14776, pp. 18–27. Springer, Cham (2024), https://doi.org/10.1007/978-3-031-63742-1_2

40. Ruangwises, S., Shinagawa, K.: Simulating virtual players for UNO without computers. arXiv preprint (2025). https://doi.org/10.48550/arXiv.2502.05987, https://arxiv.org/abs/2502.05987
41. Shinagawa, K., Kanai, K., Miyamoto, K., Nuida, K.: How to covertly and uniformly scramble the 15 puzzle and rubik's cube. In: Broder, A.Z., Tamir, T. (eds.) Fun with Algorithms. LIPIcs, vol. 291, pp. 30:1–30:15. Schloss Dagstuhl, Dagstuhl, Germany (2024), https://doi.org/10.4230/LIPIcs.FUN.2024.30
42. Shinagawa, K., Miyahara, D., Mizuki, T.: How to play Old Maid with virtual players. Theory of Computing Systems **69**(1) (2025), https://doi.org/10.1007/s00224-024-10203-w
43. 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. **E100.A**(9), 1900–1909 (2017), https://doi.org/10.1587/transfun.E100.A.1900
44. Shinoda, Y., Miyahara, D., Shinagawa, K., Mizuki, T., Sone, H.: Card-based covert lottery. In: Maimut, D., Oprina, A.G., Sauveron, D. (eds.) Innovative Security Solutions for Information Technology and Communications. LNCS, vol. 12596, pp. 257–270. Springer, Cham (2021), https://doi.org/10.1007/978-3-030-69255-1_17
45. Takahashi, Y., Shinagawa, K.: Extended addition protocol and efficient voting protocols using regular polygon cards. New Gener. Comput. **42**, 479–496 (2024), https://doi.org/10.1007/s00354-024-00275-0
46. Takahashi, Y., Shinagawa, K., Shikata, H., Mizuki, T.: Efficient card-based protocols for symmetric functions using four-colored decks. In: ACM ASIA Public-Key Cryptography Workshop. pp. 1–10. ACM, New York (2024), https://doi.org/10.1145/3659467.3659902
47. Tamura, Y., Suzuki, A., Mizuki, T.: Card-based zero-knowledge proof protocols for the 15-puzzle and the token swapping problem. In: ACM ASIA Public-Key Cryptography Workshop. pp. 11–22. ACM, New York (2024), https://doi.org/10.1145/3659467.3659905
48. Tanaka, K., Sasaki, S., Shinagawa, K., Mizuki, T.: Only two shuffles perform card-based zero-knowledge proof for Sudoku of any size. In: 2025 Symposium on Simplicity in Algorithms (SOSA). pp. 94–107. SIAM (2025), https://doi.org/10.1137/1.9781611978315.7
49. Tozawa, K., Morita, H., Mizuki, T.: Single-shuffle card-based protocol with eight cards per gate. In: Genova, D., Kari, J. (eds.) Unconventional Computation and Natural Computation. LNCS, vol. 14003, pp. 171–185. Springer, Cham (2023), https://doi.org/10.1007/978-3-031-34034-5_12
50. Ueda, I., Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: How to implement a random bisection cut. In: Martín-Vide, C., Mizuki, T., Vega-Rodríguez, M.A. (eds.) Theory and Practice of Natural Computing. LNCS, vol. 10071, pp. 58–69. Springer, Cham (2016), https://doi.org/10.1007/978-3-319-49001-4_5
51. Yao, A.C.: Protocols for secure computations. In: Foundations of Computer Science. pp. 160–164. IEEE Computer Society, Washington, DC, USA (1982), https://doi.org/10.1109/SFCS.1982.88
52. Yoshida, T., Tanaka, K., Nakabayashi, K., Chida, E., Mizuki, T.: Upper bounds on the number of shuffles for two-helping-card multi-input AND protocols. In: Deng, J., Kolesnikov, V., Schwarzmann, A.A. (eds.) Cryptology and Network Security. LNCS, vol. 14342, pp. 211–231. Springer, Singapore (2023), https://doi.org/10.1007/978-981-99-7563-1_10