# Efficient AND Protocols Resistant to Maliciously Revealing a Single Card*

Koichi Koizumi[1], Minato Abe[1], Eikoh Chida[2], and Takaaki Mizuki[3]

[1] National Institute of Technology, Fukushima College, Iwaki, Japan
[2] National Institute of Technology, Ichinoseki College, Ichinoseki, Japan
[3] Cyberscience Center, Tohoku University, Sendai, Japan

**Abstract.** In card-based cryptography for performing secure computation, after each player places an input commitment consisting of two face-down cards, all players cooperate to manipulate a sequence of cards according to a protocol. In the presence of a malicious player who does not follow the protocol, prior work has considered the active card-revealing attack and defined the $t$-secureness as the ability to prevent information about the input from being leaked even if at most $t$ cards are turned over illegally. In this paper, we first propose an efficient 1-secure AND protocol: our proposed protocol uses only eight cards and one shuffle, whereas the existing protocol requires 16 cards and eight shuffles. Our 1-secure AND protocol is quite simple and easy to implement. We next design a committed-format 1-secure AND protocol by adding four more cards; a committed-format protocol produces its output in the same format as its inputs.

**Keywords:** Card-based cryptography · Secure computation · Card-revealing attack.

## 1 Introduction

Secure computations, which involve performing computations while keeping the inputs secret, have been extensively studied and developed (e.g. [4, 32, 46]). Many secure computation protocols have been devised in the field of card-based cryptography [11, 21, 26, 38]; such a card-based protocol uses a deck of cards to physically perform a secure computation. Most of the existing card-based protocols assume that all players are semi-honest. Therefore, if a malicious player cheats without following the protocol, the confidentiality of the input, i.e., the security of such protocols, is generally compromised. There have been several studies that address such issues [12, 16, 25, 42]); among them, this paper focuses on the "active card-revealing attack" formulated by Takashima et al. [42].

* This paper appears in Proceedings of ICTAC 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 is available online at: https://doi.org/10.1007/978-3-032-11176-0_21. Use of this Accepted Version is subject to the publishers Accepted Manuscript terms of use: https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms.

### 1.1 Active Card-Revealing Attack

Takashima et al. [42] introduced an active attack assumption, considering a situation in which an attacker is willing to turn over cards illegally and does not hesitate even if detected, which they call the *active card-revealing attack*.

Card-based protocols that implement secure computations typically use black ♣ and red ♥ cards, and handle bit values according to the following encoding rule:

$$\boxed{\clubsuit}\boxed{\heartsuit} = 0, \quad \boxed{\heartsuit}\boxed{\clubsuit} = 1. \tag{1}$$

Two cards placed face down according to this encoding (1) for a given bit $x \in \{0, 1\}$ are called a *commitment* to $x$, and expressed as

$$\underbrace{\boxed{?}\boxed{?}}_{x},$$

where the pattern on the back of every card is assumed to be the same $\boxed{?}$ throughout this paper.

Typically, each player participating in a protocol secretly creates a commitment to their own private bit (unseen by other players). For example, suppose that Alice and Bob have private bits $a \in \{0, 1\}$ and $b \in \{0, 1\}$, respectively, and wish to securely compute their AND value $a \wedge b$. In this case, they first each make input commitments secretly:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\ \underbrace{\boxed{?}\boxed{?}}_{b}.$$

The first card-based protocol in history, namely the *five-card trick* [2] invented by Den Boer, adds one red card ♥ between the two commitments and outputs only the value of $a \wedge b$:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{b} \rightarrow \cdots \rightarrow a \wedge b,$$

although the detailed steps are omitted here.

If an active card-revealing attack were to be launched against this protocol, turning over one of the cards comprising the input commitments would immediately reveal whether one of the private inputs of Alice and Bob is 0 or 1. Therefore, not only the five-card trick, but also any protocol that prepares and places input commitments according to the encoding rule (1) is not secure against illegally revealing a single card.

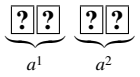
Thus, Takashima et al. [42] defined the "$t$-secureness" as the property that no information about the input is leaked even if at most $t$ cards are turned over illegally, and they constructed protocols that satisfy this requirement. We assume here that once a malicious player illegally turns over some cards, the protocol is stopped (and the players do not execute the remaining steps).

### 1.2 The Existing Protocols

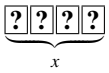
As seen in Section 1.1, if Alice places a commitment

$$\underbrace{\boxed{?}\boxed{?}}_{a}$$

to her own private input $a \in \{0, 1\}$ on the table, it is no longer 1-secure (because revealing one card would immediately leak the value of $a$). To solve this issue, Takashima et al. [42] utilized the idea of *secret sharing* [37]. Specifically, instead of creating a single commitment to $a$, Alice randomly generates two bits $a^1$ and $a^2$ such that $a = a^1 \oplus a^2$ to split $a$ into two "shares," and places these two commitments as Alice's input:




In this case, the value of $a$ is not leaked even if at most one card is turned over. In this paper, these four cards are referred to as a *split commitment* to $a$, and $a^1$ and $a^2$ are referred to as *shares* or *share commitments*. We sometimes denote a split commitment to $x \in \{0, 1\}$ by




Based on the above idea, Takashima et al. [42] proposed a 1-secure AND protocol using 16 cards and 8 shuffles. Note that each of Alice's and Bob's split commitments uses four cards as described above, and hence we require eight additional cards:

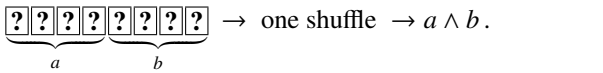

We will introduce this protocol in Section 2.

In addition, for $t \geq 2$, Takashima et al. [42] proposed a $t$-secure AND protocol using $8t + 12$ cards and $2t^2 + 7t + 2$ shuffles.

These existing protocols require multiple runs of the other existing protocols, such as the copy protocol [27] and the AND protocol [25], making them more complex to execute compared to the other AND protocols that do not consider the $t$-secureness (e.g. [2, 20, 27, 40]). Therefore, it is desirable to develop protocols that can be executed more easily. Especially, a simpler 1-secure AND protocol using fewer cards and shuffles is solicited.

### 1.3 Our Contribution

In this paper, we first propose a new 1-secure AND protocol based on a different idea from the existing protocol. Like the existing protocol, our protocol uses eight cards for input split commitments to Alice's private bit $a$ and Bob's private bit $b$, but does not require any additional cards; the number of shuffles is only one:




As will be seen in Section 3, this protocol, which we call Protocol A, is extremely simple, and thus, it is easy to implement.

We next present another protocol, called Protocol B, which produces as output a split commitment to $a \wedge b$, given two split commitments to $a$ and $b$:

Thus, Protocol B requires four additional cards aside from the eight cards for input split commitments. As will be seen in Section 4, it uses two shuffles. Protocol B is classified as a *committed-format* protocol (because it produces a commitment). A committed-format protocol is useful; for example, executing Protocol B repeatedly $n - 1$ times provides us a secure computation of the $n$-variable AND function. Note that neither Protocol A nor the existing protocols given by Takashima et al. [42] are committed-format ones.

Table 1 shows the numbers of required cards and shuffles for the existing 1-secure AND protocol and ours. Both of our protocols, i.e., Protocols A and B, can be executed with fewer cards and fewer shuffles than the existing protocol.

**Table 1.** Comparison of 1-secure AND protocols

| | # of cards | # of shuffles | committed |
|---|---|---|---|
| Takashima et al. [42] | 16 | 8 | no |
| Our Protocol A (§3) | 8 | 1 | no |
| Our Protocol B (§4) | 12 | 2 | yes |

Although we omit the details due to the page limitation, our Protocol B can be extended to a $t$-secure AND protocol for any $t \geq 2$.

### 1.4 Situations of Use of $t$-Secure Protocols

Here, we mention three scenarios where $t$-secure protocols would be useful.

- Suppose that Bob has malicious intent when Alice and Bob execute a protocol; then, he illegally flips at most $t$ cards. In this case, Alice's input is not leaked (thanks to the $t$-secureness). Since the trust between Alice and Bob is lost, the protocol will never be executed by these two again.
- Even if a malicious third party illegally flips at most $t$ cards, neither Alice's nor Bob's input will be leaked. After the third party is dismissed, both the players restart the protocol with new input commitments.
- A simple mistake, in which a player unintentionally flips over a few cards during a card operation[iv], is common, especially among players who are not familiar with card operations. In this case, neither Alice nor Bob's input is leaked (as long as the number of mistakenly opened cards is at most $t$). Each player then makes a new input commitment and restarts the protocol from the beginning.

### 1.5 Related Work

As mentioned above, several studies have explored active attacks or related concepts. Koch and Walzer [12] addressed active attacks on card-based protocols, focusing on the

[iv] Another operative error was discussed in [22].

use of envelopes to prevent malicious actions. Manabe and Ono [16] also employed envelopes to construct protocols resistant to malicious players. Mizuki and Shizuya [25] considered information leakage due to scuff marks on cards and proposed countermeasures. Morooka et al. [28] presented three-player protocols designed to prevent malicious actions by observers.

Card-based cryptography is a dynamic field with a growing body of research. Recent areas of interest include: zero-knowledge proof protocols for Sudoku [44], other puzzles [5,6,19,31,34,36], games [15], and graph problems [43]; private set intersection protocols [3]; private-model protocols using standard decks [10, 17, 29]; applications of 3D printers [8]; shuffle-efficient protocols based on garbled circuit [30, 45]; efficient protocols for symmetric functions [33, 41]; and applications to card games [23, 35, 39].

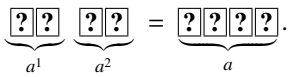
### 1.6 Organization of This Paper

The remainder of this paper is organized as follows. In Section 2, we describe the existing 1-secure AND protocol [42]. Next, in Section 3, we present a new 1-secure AND protocol (i.e., Protocol A), which is much simpler than the existing protocols. Then, in Section 4, we develop a committed-format 1-secure AND protocol (i.e., Protocol B). Finally, in Section 5, we conclude with our results.

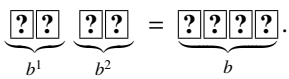
## 2 Existing 1-Secure Protocol

In this section, we introduce the 1-secure AND protocol proposed by Takashima et al. [42].

As noted above, this protocol uses the idea of secret sharing to conceal the value of each player's input bit. That is, instead of directly creating a commitment to $a$ based on the encoding rule (1), Alice splits her private bit as in $a = a^1 \oplus a^2$, and creates a split commitment to $a$:




For example, if $a = 1$, there are two options: $(a^1, a^2) = (1, 0)$ and $(a^1, a^2) = (0, 1)$, from which Alice chooses one uniformly at random. Similarly, Bob creates a split commitment to $b$ satisfying $b = b^1 \oplus b^2$:

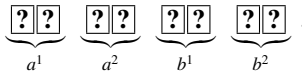



With these split commitments, if one of their cards is illegally turned over, only one of the shares that make up $a$ or $b$ will be known, and hence, the values of $a$ and $b$ themselves cannot be identified from there[v]. Given such two split commitments along with eight additional cards, the existing 1-secure AND protocol [42] proceeds as follows.

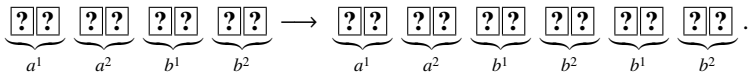
**The Existing 1-Secure AND Protocol [42]**

[v] As assumed, once a card is turned over illegally, the protocol is stopped immediately.

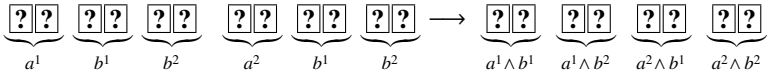
1. Place the two input split commitments, i.e., four share commitments, as follows:




2. Duplicate each of the share commitments to $b^1$ and $b^2$ using the existing copy protocol [27] (which requires four free cards ♣♥♣♥):




3. Obtain commitments to $a^1 \wedge b^1$ and $a^1 \wedge b^2$ from commitments to $a^1$, $b^1$, and $b^2$ using the existing AND protocol [25] (which requires four free cards ♣♥♣♥); similarly, obtain commitments to $a^2 \wedge b^1$ and $a^2 \wedge b^2$:




4. Use the 4-bit XOR sub-protocol (see [42] for details) to compute $(a^1 \wedge b^1) \oplus (a^1 \wedge b^2) \oplus (a^2 \wedge b^1) \oplus (a^2 \wedge b^2)$, which is equal to $a \wedge b$.

This is the 1-secure AND protocol proposed by Takashima et al. [42]. This protocol requires eight additional cards in addition to eight cards for input split commitments. Therefore, the total number of required cards is 16. The number of required shuffles is eight although the details are omitted (see [42]).

In the next section, we present a very simple 1-secure AND protocol that can be executed with only eight cards and one shuffle.

## 3 Our Simple 1-Secure AND Protocol

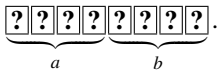
In this section, we propose a simple 1-secure AND protocol that does not require any additional cards and uses only one shuffle (namely, Protocol A shown in Table 1).

In Section 3.1, we discuss the idea behind our protocol. We then describe the protocol in Section 3.2 and its pseudocode in Section 3.3. In Section 3.4, we show the correctness and security of our protocol.

### 3.1 Idea

In this subsection, we briefly explain the idea behind our protocol.

Assume that we have two split commitments:

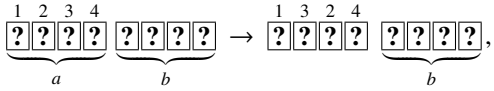



Remember that the four cards of each split commitment satisfy the following patterns, depending on its value:

| 0 | 1 |
|---|---|
| ♣♥♣♥ or ♥♣♥♣ | ♣♥♥♣ or ♥♣♣♥ |

For example, if $a = 0$, the four cards are ♣♥♣♥ or ♥♣♥♣ (each occurs with a probability of $1/2$).

Let us exchange the positions of the second and third cards of the split commitment to $a$:




where we attach numbers (from 1 to 4) to the four cards for convenience sake. Then, the four cards (related to $a$) become as follows:

| $a = 0$ | $a = 1$ |
|---|---|
| ♣♥♣♥ → ♣♣♥♥ | ♣♥♥♣ → ♣♥♥♣ |
| ♥♣♥♣ → ♥♥♣♣ | ♥♣♣♥ → ♥♣♣♥ |

Now let us compare the four-card sequence related to $a$ with the four-card sequence of the split commitment to $b$; especially, focus on the Hamming distance (which can be naturally defined, based on {♣, ♥}). For instance, the Hamming distance between ♣♣♥♥ ($a = 0$) and ♣♥♣♥ ($b = 0$) is 2. Somewhat surprisingly, when $a \wedge b = 0$, the Hamming distance is always 2. Furthermore, when $a \wedge b = 1$, the Hamming distance is either 0 or 4. The following table enumerates the Hamming distances for all cases:

| $b$ \ $a$ | ♣♣♥♥ | ♥♥♣♣ | ♣♥♥♣ | ♥♣♣♥ |
|---|---|---|---|---|
| ♣♥♣♥ | 2 | 2 | 2 | 2 |
| ♥♣♥♣ | 2 | 2 | 2 | 2 |
| ♣♥♥♣ | 2 | 2 | 0 | 4 |
| ♥♣♣♥ | 2 | 2 | 4 | 0 |

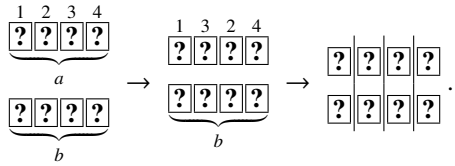
Thus, the Hamming distance between these two four-card sequences tells us the value of $a \wedge b$. Since we want to know only the Hamming distance, we will apply a shuffle while keeping their distance unchanged, as will be seen in the next subsection.

### 3.2 Description of Protocol

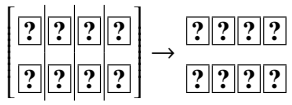
In this subsection, we give a complete description of our 1-secure AND protocol.

Given split commitments to $a$ and $b$, our protocol proceeds as follows.

1. After placing the input split commitments as below, swap the second and third cards of the split commitment to $a$ and make four two-card piles as follows:



2. Apply a *pile-scramble shuffle* [7]:



which permutes the four two-card piles uniformly at random, resulting in one of the 4! possibilities with a probability of 1/4!. Note that the Hamming distance between the upper four-card sequence and the lower four-card sequence has been unchanged. (This shuffle can be implemented, for example, by placing each pile of cards in an envelope and stirring the four envelopes uniformly at random.)
3. Turn over all the cards, and let HD be the Hamming distance between the upper and lower four-card sequences. If $\mathtt{HD} = 0$ or $\mathtt{HD} = 4$, then $a \wedge b = 1$; if $\mathtt{HD} = 2$, then $a \wedge b = 0$.

This is our 1-secure AND protocol. It uses only eight cards and one pile-scramble shuffle.

Our 1-secure AND protocol works correctly and securely because the Hamming distance surely reveals the value of $a \wedge b$ and the pile-scramble shuffle erases any information more than the value of $a \wedge b$. Moreover, the active card-revealing attack can be done at Step 1 or 2, but revealing a single card does not leak any information about $a$ or $b$. A more formal treatment will be given in the following subsections.

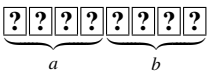
### 3.3 Pseudocode

In this subsection, we describe our protocol more formally, based on the computational model of card-based protocols [24].

In general, a protocol is supposed to achieve a desired functionality by permuting, shuffling, and/or turning over cards. We explain these three actions briefly: $(\mathsf{perm}, \pi)$ permutes the sequence of cards according to a given permutation $\pi$; given a set of permutations $\Pi$, $(\mathsf{shuf}, \Pi)$ chooses $\pi \in \Pi$ uniformly at random and applies $\pi$ to the sequence of cards; and $(\mathsf{turn}, T)$ turns over every $t$-th card with $t \in T$, given a set $T$ of indices.

The following is a pseudocode of our 1-secure AND protocol, where $\mathrm{PSS}_{(4,2)}$ represents the permutation set corresponding to the pile-scramble shuffle for four two-card piles.

**Input:**




1: $(\mathsf{perm}, (2\,3))$
2: $(\mathsf{perm}, (2\,3\,5)\,(4\,7\,6))$
3: $(\mathsf{shuf}, \mathrm{PSS}_{(4,2)})$
4: $(\mathsf{turn}, \{1, 2, 3, 4, 5, 6, 7, 8\})$
5: **if** $\mathtt{HD} = 2$ **then**
6: $\quad(\mathsf{result}, "a \wedge b = 0")$
7: **else**
8: $\quad(\mathsf{result}, "a \wedge b = 1")$
9: **end if**

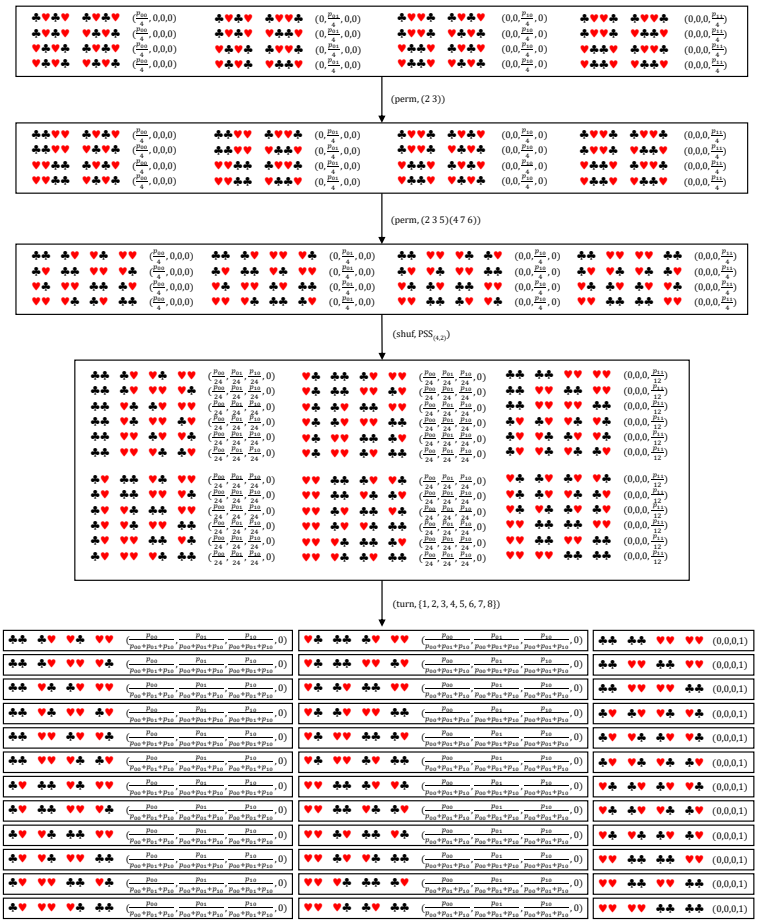
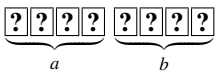



**Fig. 1.** KWH-tree of non-committed-format 1-secure AND protocol

### 3.4 Correctness and Security

In this subsection, we use the modified version [22] of the *KWH-tree* [13] to show the correctness and security of our protocol more formally. Figure 1 is such a KWH-tree for our protocol, which we explain, as below.

Let $p_{ij}$ for every $(i, j) \in \{0, 1\}^2$ denote a probability that input $(a, b)$ (of Alice and Bob) is equal to $(i, j)$. For example, if $(a, b) = (0, 0)$, the input sequence




is one of the following four possibilities:

♣♥♣♥ ♣♥♣♥, ♣♥♣♥ ♥♣♥♣, ♥♣♥♣ ♣♥♣♥, ♥♣♥♣ ♥♣♥♣;

each of these occurs with a probability of $\frac{p_{00}}{4}$ (before the protocol). Thus, we write this (partial) 'status' as

$$\clubsuit\heartsuit\clubsuit\heartsuit\ \clubsuit\heartsuit\clubsuit\heartsuit\ (p_{00}/4,0,0,0) \qquad \clubsuit\heartsuit\clubsuit\heartsuit\ \heartsuit\clubsuit\heartsuit\clubsuit\ (p_{00}/4,0,0,0)$$
$$\heartsuit\clubsuit\heartsuit\clubsuit\ \clubsuit\heartsuit\clubsuit\heartsuit\ (p_{00}/4,0,0,0) \qquad \heartsuit\clubsuit\heartsuit\clubsuit\ \heartsuit\clubsuit\heartsuit\clubsuit\ (p_{00}/4,0,0,0),$$

where a four-tuple $(q_{00}, q_{01}, q_{10}, q_{11})$ along with a card sequence means that the probability that $(a, b) = (i, j)$ and the card sequence occurs is $q_{ij}$ for every $(i, j) \in \{0, 1\}^2$. Considering also the remaining three cases $(a, b) = (0, 1), (1, 0), (1, 1)$, we obtain the topmost box in Fig. 1 as the initial (full) *status*.

The initial status (and succeeding statuses) are transformed into another status by an action, such as $(\mathsf{perm}, (2\,3))$, $(\mathsf{perm}, (2\,3\,5)\,(4\,7\,6))$, and $(\mathsf{shuf}, \mathrm{PSS}_{(4,2)})$, as shown in Fig. 1. In particular, the final action $(\mathsf{turn}, \{1, 2, 3, 4, 5, 6, 7, 8\})$ results in 36 "leaf" statuses.

Note that in each of the topmost four statuses depicted in Fig. 1, the (coordinate-wise) sum of all tuples is equal to $(p_{00}, p_{01}, p_{10}, p_{11})$; this guarantees that no information about the input $(a, b)$ will be leaked. Regarding the 36 leaf statuses, each of them has only one element, which is either $(\frac{p_{00}}{p_{00}+p_{01}+p_{10}}, \frac{p_{01}}{p_{00}+p_{01}+p_{10}}, \frac{p_{10}}{p_{00}+p_{01}+p_{10}}, 0)$ or $(0, 0, 0, 1)$; this guarantees the correctness and also implies that any information other than the value of $a \wedge b$ will not be leaked.

Now, let us consider an active card-revealing attack. Assume for example that the leftmost card is illegally turned over after $(\mathsf{perm}, (2\,3))$, i.e., apply $(\mathsf{turn}, \{1\})$ instead of $(\mathsf{perm}, (2\,3\,5)\,(4\,7\,6))$. Then, we have the following status:

$$\clubsuit\clubsuit\heartsuit\heartsuit\ \clubsuit\heartsuit\clubsuit\heartsuit\ (p_{00}/2,0,0,0) \qquad \clubsuit\heartsuit\heartsuit\clubsuit\ \clubsuit\heartsuit\clubsuit\heartsuit\ (0,0,p_{10}/2,0)$$
$$\clubsuit\clubsuit\heartsuit\heartsuit\ \heartsuit\clubsuit\heartsuit\clubsuit\ (p_{00}/2,0,0,0) \qquad \clubsuit\heartsuit\heartsuit\clubsuit\ \heartsuit\clubsuit\heartsuit\clubsuit\ (0,0,p_{10}/2,0)$$
$$\clubsuit\clubsuit\heartsuit\heartsuit\ \clubsuit\heartsuit\heartsuit\clubsuit\ (0,p_{01}/2,0,0) \qquad \clubsuit\heartsuit\heartsuit\clubsuit\ \clubsuit\heartsuit\heartsuit\clubsuit\ (0,0,0,p_{11}/2)$$
$$\clubsuit\clubsuit\heartsuit\heartsuit\ \heartsuit\clubsuit\clubsuit\heartsuit\ (0,p_{01}/2,0,0) \qquad \clubsuit\heartsuit\heartsuit\clubsuit\ \heartsuit\clubsuit\clubsuit\heartsuit\ (0,0,0,p_{11}/2).$$

Since the (coordinate-wise) sum of all tuples is still equal to $(p_{00}, p_{01}, p_{10}, p_{11})$, no information about the input $(a, b)$ is leaked (and the protocol aborts here). This is true for the other statuses (or the final statuses reveal all the cards), and hence, we can confirm that our protocol is 1-secure.

## 4 Our Committed-Format 1-Secure AND Protocol

In this section, we focus on committed-format protocols, which mean that their input and output formats are consistent. The existing protocol explained in Section 2 and our simple 1-secure AND protocol presented in Section 3 are not committed-format ones. Here, we propose a committed-format 1-secure AND protocol that can be executed with four additional cards (namely, Protocol B in Table 1).

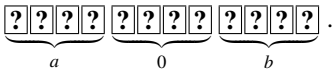
In Section 4.1, we provide the idea behind the protocol. In Section 4.2, we describe the randomization sub-protocol used in the proposed protocol. The details of the protocol and its pseudocode are described in Sections 4.3 and 4.4, respectively. We then confirm its correctness and security in Section 4.5.

### 4.1 Idea

First of all, we borrow the idea behind the Mizuki–Sone AND protocol [27], which is based on the following equation:

$$a \wedge b = \begin{cases} 0 & \text{if } a = 0, \\ b & \text{if } a = 1. \end{cases} \tag{2}$$

Thus, we prepare a split commitment to 0 in addition to input split commitments to $a$ and $b$:

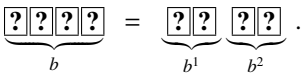



As Eq. (2) implies, depending on the value of $a$, either 0 or $b$ should be the output of an AND protocol. Since directly turning over the split commitment to $a$ would leak its value, it is also necessary to perform randomization as in the Mizuki–Sone AND protocol. This will be detailed in Section 4.3.

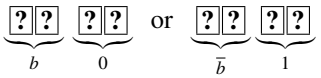
Additionally, since either the split commitment to 0 or the split commitment to $b$ will be the output, it is necessary to perform randomization to erase prior information about the orders of the cards in these split commitments. This will be introduced in the next subsection.

### 4.2 Randomization Sub-Protocol

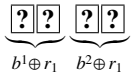
As explained in the previous subsection, either a split commitment to 0 or $b$ will be the output of our protocol. Consider a split commitment to $b$ placed by Bob:



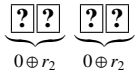

Since Bob knows both shares, $b^1$ and $b^2$, of the split commitment to $b$, he knows whether it is




To eliminate such Bob's prior knowledge, we want to add a random bit $r_1$ as follows:




Similarly, for a split commitment to 0, we want to add a random bit $r_2$:

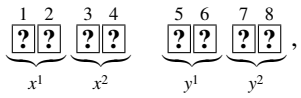



In our protocol, since only one of the split commitment to $b$ and the split commitment to 0 is used as output, it is not necessary to use both random bits $r_1$ and $r_2$ simultaneously. Therefore, it suffices to add a common random bit $r$ to both 0 and $b$.

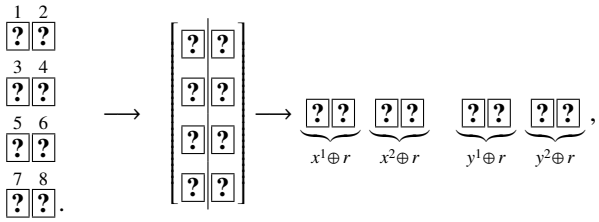
The following randomization sub-protocol achieves this: given two split commitments to $x$ and $y$, it adds a common random bit $r$.

**Randomization Sub-Protocol**

1. Given two split commitments to $x = x^1 \oplus x^2$ and $y = y^1 \oplus y^2$:




   place each share from top to bottom as follows, and apply a pile-scramble shuffle (i.e., randomize only the column positions while maintaining each of the two vertical columns on the left and right):

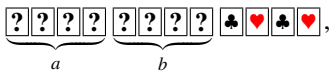



   where $r$ is a uniformly distributed random bit, generated by the shuffle.

### 4.3 Description of Protocol

In this subsection, we present our committed-format 1-secure AND protocol.

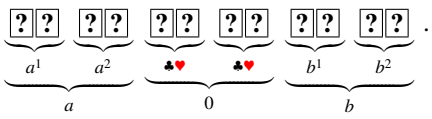
Given input split commitments to $a$ and $b$ along with four additional cards




our protocol proceeds as follows.

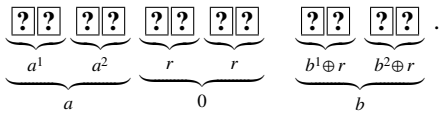
**Committed-Format 1-Secure AND Protocol**

1. Place the input split commitments, and make a split commitment to 0, as follows:



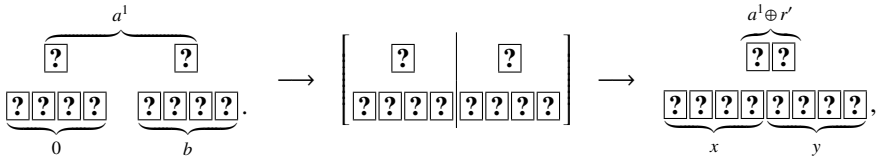

   Note that if $a^1 \oplus a^2 = a = 0$, then the four cards in the middle have the same value as $a \wedge b\,(= 0 \wedge b = 0)$. If $a^1 \oplus a^2 = a = 1$, then the rightmost four cards have the same value as $a \wedge b\,(= 1 \wedge b = b)$.

2. Apply the randomization sub-protocol explained in Section 4.2 to the split commitments to 0 and $b$; then we obtain




From now on, we apply the idea behind the Mizuki–Sone AND protocol [27] to the three split commitments to $a$, 0, and $b$.

3. Place the two cards of the share $a^1$ as shown below, i.e., the left card is placed above the split commitment to 0, and the right card is placed above the split commitment to $b$, as follows, and apply a pile-scramble shuffle:




where

$$(x, y) = \begin{cases} (0, b) & \text{if } r' = 0, \\ (b, 0) & \text{if } r' = 1. \end{cases}$$

4. Turn over the commitments to $a^2$ and $a^1 \oplus r'$ to compute $a^2 \oplus (a^1 \oplus r') = a \oplus r'$.
   - If $a \oplus r' = 0$, then $x$ is a split commitment to $a \wedge b$.
   - If $a \oplus r' = 1$, then $y$ is a split commitment to $a \wedge b$.

This is our committed-format 1-secure AND protocol. It uses 12 cards and two pile-scramble shuffles. Since this is a committed-format protocol, it is useful; for example, a secure AND computation with more than two inputs can also be realized.

### 4.4 Pseudocode

The following is a pseudocode of our committed-format 1-secure AND protocol, where the result action specifies the positions of the output split commitment.

**Input:**


1: (shuf, {id, (5 6)(7 8)(9 10)(11 12)})
2: (shuf, {id, (1 2)(5 9)(6 10)(7 11)(8 12)})
3: (turn, {1, 2, 3, 4})
4: **if** ♣♥♣♥ or ♥♣♥♣ appears **then**
5: (result, (5, 6, 7, 8))
6: **else**
7: (result, (9, 10, 11, 12))
8: **end if**

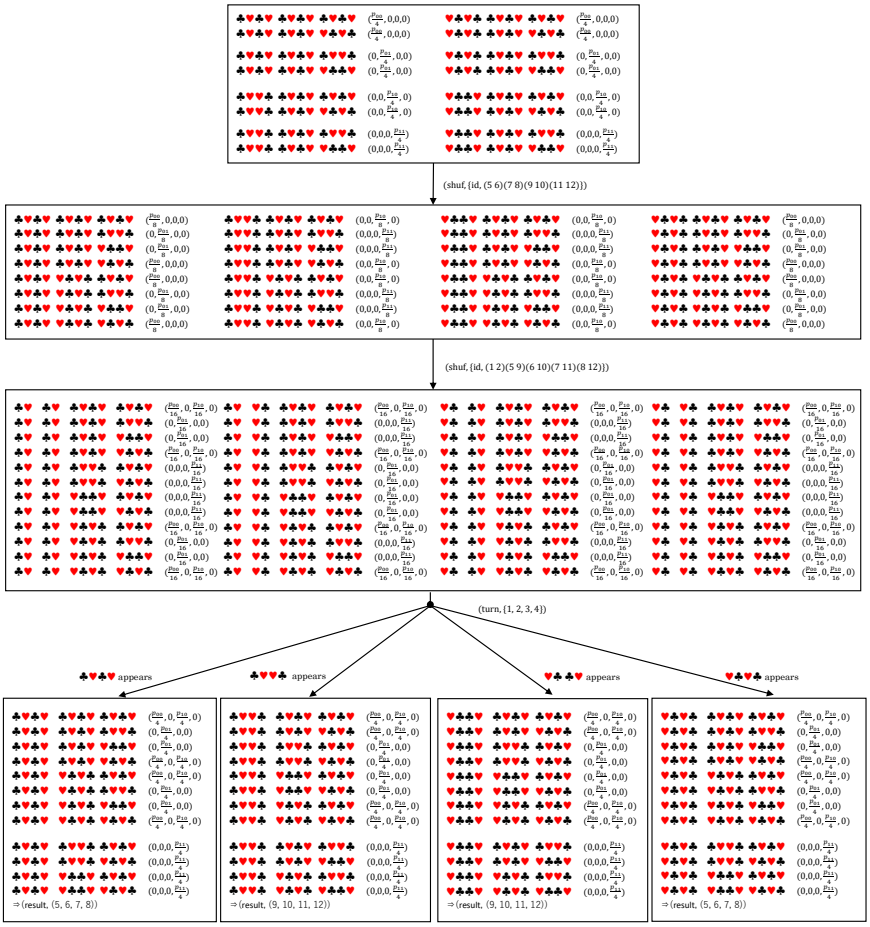



**Fig. 2.** KWH tree of committed-format 1-secure AND protocol

### 4.5 Correctness and Security

Basically, the correctness and security of our protocol come from the Mizuki–Sone AND protocol [27]. More formally, we depict the KWH-tree in Fig. 2.

There are seven statuses in the KWH-tree. Similar to our non-committed-format protocol, i.e., Protocol A, the topmost status consists of 16 elements. The status is changed by the actions. In each status, the sum of all tuples is equal to $(p_{00}, p_{01}, p_{10}, p_{11})$, which ensures that no information about the inputs $a$ and $b$ is leaked. In the four final statuses, we can confirm the correctness (i.e., the value of $a \wedge b$ is surely computed). Furthermore, one can confirm that if any illegal $(\mathsf{turn}, \{i\})$ is applied anywhere, it results in a status whose sum is $(p_{00}, p_{01}, p_{10}, p_{11})$, implying the 1-secureness.

## 5 Conclusion

In this paper, we designed a simple 1-secure AND protocol using a novel approach compared to the existing protocols. Our proposed protocol uses only eight cards, i.e., it does not require any additional cards beyond the input split commitments. The number of shuffles is only one. We believe that our protocol is easy to implement, as illustrated in Fig. 3 with real physical cards. Furthermore, we also proposed a committed-format 1-secure AND protocol that can be executed with four additional cards, bringing the total number of cards to 12. The number of shuffles is two.


**Fig. 3.** Execution of our protocol.

As shown in Table 1, both of the proposed protocols can be executed with fewer cards and fewer shuffles than the existing protocols, making them more practical and easy to understand. Although this paper focuses on designing only 1-secure AND protocols, the assumption of $t = 1$ may suffice to protect against casual attacks or operational errors. In addition, our 1-secure committed-format AND protocol can be extended to a $t$-secure protocol for any $t \geq 2$.

Our contributions provide significant improvements in the efficiency and practicality of card-based protocols. Future work could explore further optimizations, expansions to other computations (beyond the AND function), and applications of these protocols in various secure computation scenarios, as well as investigate their robustness against other types of attacks. Since our protocols use pile-scramble shuffles, constructing protocols using the random cut ( [1,2]), which is an easier shuffing operation, would be a desired direction for future work. Expanding our ideas to devise protocols using other familiar tools such as coins [14,18] and a balance scale [9] presents an interesting challenge for future work.

**Acknowledgments.** We thank the anonymous reviewers, whose comments have helped us improve the presentation of the paper. We thank Kazumasa Shinagawa for his helpful comments on the sub-protocol described in Section 4.2. This work was supported by JSPS KAKENHI Grant Numbers JP23H00479 and JP24K02938.

## References

1. Abe, Y., Mizuki, T., Sone, H.: Committed-format AND protocol using only random cuts. Nat. Comput. **20**(4), 639–645 (2021), https://doi.org/10.1007/s11047-021-09862-2

2. 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
3. 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
4. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. pp. 169–178. STOC '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1536414.1536440, https://doi.org/10.1145/1536414.1536440
5. 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
6. 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
7. 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
8. Ito, Y., Shikata, H., Suganuma, T., Mizuki, T.: Card-based cryptography meets 3d printer. In: Da-Jung Cho, J.K. (ed.) 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
9. Kaneko, S., Lafourcade, P., Mallordy, L.B., Miyahara, D., Puys, M., Sakiyama, K.: Balance-based ZKP protocols for pencil-and-paper puzzles. In: Mouha, N., Nikiforakis, N. (eds.) Information Security. pp. 211–231. LNCS, Springer, Cham (2025), https://doi.org/10.1007/978-3-031-75757-0_11
10. Kobayashi, N., Manabe, Y.: Card-based cryptographic protocols for three-input functions with a standard deck of cards using private operations. In: Garcia-Alfaro, J., Barker, K., Navarro-Arribas, G., Pérez-Solà, C., Delgado-Segura, S., Katsikas, S., Cuppens, F., Lambrinoudakis, C., Cuppens-Boulahia, N., Pawlicki, M., Choraś, M. (eds.) Computer Security. ESORICS 2024 International Workshops. pp. 94–111. LNCS, Springer, Cham (2025)
11. 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
12. Koch, A., Walzer, S.: Foundations for actively secure card-based cryptography. In: Farach-Colton, M., Prencipe, G., Uehara, R. (eds.) Fun with Algorithms. LIPIcs, vol. 157, pp. 17:1–17:23. Schloss Dagstuhl, Dagstuhl, Germany (2020), https://doi.org/10.4230/LIPIcs.FUN.2021.17
13. Koch, A., Walzer, S., Härtel, K.: Card-based cryptographic protocols using a minimal number of cards. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology—ASIACRYPT 2015. LNCS, vol. 9452, pp. 783–807. Springer, Berlin, Heidelberg (2015), https://doi.org/10.1007/978-3-662-48797-6_32
14. Komano, Y., Mizuki, T.: Coin-based secure computations. Int. J. Inf. Secur. **21**, 833–846 (2022), https://doi.org/10.1007/s10207-022-00585-8
15. 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

16. Manabe, Y., Ono, H.: Card-based cryptographic protocols with malicious players using private operations. New Gener. Comput. **40**, 67–93 (2022), https://doi.org/10.1007/s00354-021-00148-w
17. 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
18. Minamikawa, Y., Shinagawa, K.: Coin-based cryptographic protocols without hand operations. IEICE Trans. Fundamentals **E107.A**(8), 1178–1185 (2024), https://doi.org/10.1587/transfun.2023EAP1082
19. 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
20. Mizuki, T.: Card-based protocols for securely computing the conjunction of multiple variables. Theor. Comput. Sci. **622**(C), 34–44 (2016), https://doi.org/10.1016/j.tcs.2016.01.039
21. Mizuki, T.: Preface: Special issue on card-based cryptography 3. New Gener. Comput. **42**, 303–304 (2024), https://doi.org/10.1007/s00354-024-00280-3
22. Mizuki, T., Komano, Y.: Information leakage due to operative errors in card-based protocols. Inf. Comput. **285**, 104910 (2022), https://doi.org/10.1016/j.ic.2022.104910
23. Mizuki, T., Kuzuma, T., Hirano, T., Oshima, R., Yasuda, M.: Gakmoro: An application of physical secure computation to card game. In: Unconventional Computation and Natural Computation. LNCS, Springer, Cham (2025, to appear)
24. 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
25. Mizuki, T., Shizuya, H.: Practical card-based cryptography. In: Ferro, A., Luccio, F., Widmayer, P. (eds.) Fun with Algorithms. LNCS, vol. 8496, pp. 313–324. Springer, Cham (2014), https://doi.org/10.1007/978-3-319-07890-8_27
26. 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
27. 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
28. Morooka, T., Manabe, Y., Shinagawa, K.: Malicious player card-based cryptographic protocols with a standard deck of cards using private operations. In: Meng, W., Yan, Z., Piuri, V. (eds.) Information Security Practice and Experience. Lecture Notes in Computer Science, vol. 14341, pp. 332–346. Springer (2023), https://doi.org/10.1007/978-981-99-7032-2_20
29. 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
30. Ono, T., Shinagawa, K., Nakai, T., Watanabe, Y., Iwamoto, M.: Single-shuffle card-based protocols with six cards per gate. In: Seo, H., Kim, S. (eds.) Information Security and Cryptology. LNCS, vol. 14562, pp. 157–169. Springer, Singapore (2024), https://doi.org/10.1007/978-981-97-1238-0_9
31. Otsuji, T., Fulla, P., Fukunaga, T.: NP-Completeness and physical zero-knowledge proof of Hotaru Beam. In: Chen, Y., Gao, X., Sun, X., Zhang, A. (eds.) Computing and

Combinatorics. pp. 239–251. Springer, Singapore (2024), https://doi.org/10.1007/978-981-96-1090-7_20
32. Parter, M.: Secure Computation Meets Distributed Universal Optimality . In: 2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS). pp. 2336–2368. IEEE Computer Society, Los Alamitos, CA, USA (Nov 2023). https://doi.org/10.1109/FOCS57990.2023.00144, https://doi.ieeecomputersociety.org/10.1109/FOCS57990.2023.00144
33. Ruangwises, S.: The landscape of computing symmetric n-variable functions with 2n cards. In: Ábrahám, E., Dubslaff, C., Tarifa, S.L.T. (eds.) Theoretical Aspects of Computing – ICTAC 2023. LNCS, vol. 14446, pp. 74–82. Springer, Cham (2023), https://doi.org/10.1007/978-3-031-47963-2_6
34. Ruangwises, S., Iwamoto, M.: Printing protocol: Physical ZKPs for decomposition puzzles. New Gener. Comput. **42**, 331–343 (2024), https://doi.org/10.1007/s00354-024-00266-1
35. Ruangwises, S., Shinagawa, K.: Simulating virtual players for UNO without computers. In: Unconventional Computation and Natural Computation. LNCS, Springer, Cham (2025, to appear)
36. Sasaki, S., Shinagawa, K.: Physical zero-knowledge proof for Sukoro. New Gener. Comput. **42**, 381–398 (2024), https://doi.org/10.1007/s00354-024-00271-4
37. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (Nov 1979). https://doi.org/10.1145/359168.359176, https://doi.org/10.1145/359168.359176
38. Shinagawa, K.: On the Construction of Easy to Perform Card-Based Protocols. Ph.D. thesis, Tokyo Institute of Technology (2020), https://t2r2.star.titech.ac.jp/cgi-bin/publicationinfo.cgi?q_publication_content_number=CTT100817272
39. 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
40. Stiglic, A.: Computations with a deck of cards. Theor. Comput. Sci. **259**(1–2), 671–678 (2001), https://doi.org/10.1016/S0304-3975(00)00409-6
41. 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
42. Takashima, K., Miyahara, D., Mizuki, T., Sone, H.: Actively revealing card attack on card-based protocols. Nat. Comput. **21**(4), 615–628 (2021), https://doi.org/10.1007/s11047-020-09838-8
43. 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
44. 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
45. Tozawa, K., Morita, H., Mizuki, T.: Single-shuffle card-based protocol with eight cards per gate and its extensions. Natural Computing **24**(1), 131–147 (2025), https://doi.org/10.1007/s11047-024-10006-5
46. Yao, A.C.: Protocols for secure computations. In: 23rd Annual Symposium on Foundations of Computer Science (FOCS 1982). pp. 160–164 (1982). https://doi.org/10.1109/SFCS.1982.38