A Secure Three-Input AND Protocol with a Standard Deck of Minimal Cards

Hiroto Koyama[1], Daiki Miyahara[2,4](✉), Takaaki Mizuki[3,4], and Hideaki Sone[3]

[1] School of Engineering, Tohoku University, 6-6 Aramaki-Aza-Aoba, Aoba-ku,
Sendai 980-8579, Japan
hiroto.koyama.t4@dc.tohoku.ac.jp

[2] Graduate School of Information Sciences, Tohoku University,
6-3-09 Aramaki-Aza-Aoba, Aoba-ku, Sendai 980–8579, Japan
daiki.miyahara.q4@dc.tohoku.ac.jp

[3] Cyberscience Center, Tohoku University, 6-3 Aramaki-Aza-Aoba, Aoba-ku,
Sendai 980-8578, Japan
mizuki+lncs@tohoku.ac.jp

[4] National Institute of Advanced Industrial Science and Technology (AIST),
2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan

Abstract. Card-based protocols are used to perform cryptographic tasks such as secure multiparty computation using a deck of physical cards. While most of the existing protocols use a two-colored deck consisting of red cards and black cards, Niemi and Renvall in 1999 constructed protocols for securely computing two-input Boolean functions (such as secure logical AND and XOR computations) using a commonly available standard deck of playing cards. Since this initial investigation, two-input protocols with fewer cards and/or shuffles have been designed, and by combining them, one can perform a secure computation of any Boolean circuit. In this paper, we directly construct a simple card-based protocol for the three-input AND computation. Our three-input AND protocol requires fewer cards and shuffles compared to that required when applying any existing two-input AND protocol twice to perform the three-input AND computation. Our protocol is unique in the sense that it is card minimal if we use two cards to encode a single bit.

Keywords: Card-based cryptography · Secure computation · Real-life hands-on cryptography · Logical AND function

1 Introduction

Card-based protocols perform cryptographic tasks such as secure multiparty computation using a deck of physical cards. Most existing studies in this line of research use a two-colored deck consisting of indistinguishable red $\boxed{\heartsuit}$ and black $\boxed{\clubsuit}$ cards, whose backs have the same pattern $\boxed{?}$. The Boolean values are usually encoded as follows:

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

R. Santhanam and D. Musatov (Eds.): CSR 2021, LNCS 12730, pp. 242–256, 2021.
https://doi.org/10.1007/978-3-030-79416-3_14

If two face-down cards represent a bit $x \in \{0,1\}$ according to the above encoding rule, then we call them a *commitment to* x and write it as follows:

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

Previous research has proposed many card-based protocols that are capable of securely computing Boolean functions such as the logical AND function. As mentioned above, most prior studies have used a two-colored deck of cards (e.g., [1,2,4,5,7,16,24]); however, there are a few protocols that use a *standard deck of playing cards*, as introduced below.[1]

1.1 Card-Based Protocols with a Standard Deck of Cards

In 1999, Niemi and Renvall [18] proposed card-based protocols for secure two-input computations (such as computations of the logical AND and XOR) using a commonly available standard deck of playing cards for the first time. Since then, several protocols using such a standard deck have been proposed [6,9,10].

A typical deck of playing cards consists of 52 distinct cards excluding jokers; hence, we assume the following deck of 52 numbered cards:

$$\boxed{1}\,\boxed{2}\,\boxed{3}\,\boxed{4}\,\boxed{5}\,\boxed{6}\,\cdots\,\boxed{51}\,\boxed{52},$$

where all their backs are identical $\boxed{?}$.

Niemi and Renvall [18] considered, based on two cards $\boxed{i}$ and $\boxed{j}$, $1 \leq i < j \leq 52$, an encoding scheme such that

$$\boxed{i}\boxed{j} = 0, \quad \boxed{j}\boxed{i} = 1. \tag{1}$$

That is, two cards can be represented as 0 if the left number is smaller than the right one; otherwise, they are represented as 1. In this way, similar to the two-colored case, we can consider a *commitment to* a bit $x \in \{0,1\}$ with two numbered cards $\boxed{i}$ and $\boxed{j}$, denoted by

$$\underbrace{\boxed{?}\boxed{?}}_{[x]^{\{i,j\}}}.$$

Here, the set $\{i,j\}$ is called a *base* of the commitment. We sometimes write

$$\underbrace{\boxed{?}\boxed{?}}_{[x]}$$

without the description of a base if it is clear from context or there are multiple possibilities for the base. Note that, swapping the two face-down cards of a given

[1] There are other types of cards used for secure computations, such as polarizing cards [29], polygon cards [30], triangle cards [28], and dihedral cards [27].

commitment to $x \in \{0,1\}$ converts it into a commitment to the negation $\bar{x}$ while keeping the value of x secret; thus, a secure NOT computation is easy.

Based on this encoding, Niemi and Renvall [18] designed card-based protocols. They were followed by a couple of research groups [6,10], whose advancements will be detailed in the next subsection.

1.2 Existing Protocols

Table 1 presents the existing protocols for the secure computation of the two-input AND function with a standard deck of cards. A protocol that terminates in a finite number of steps is said to be *finite*, while a protocol with a runtime that is finite in expectation is said to be *Las Vegas*; the fourth column of Table 1 indicates this.

Niemi and Renvall [18] proposed a Las Vegas two-input AND protocol using five cards; four cards are used to represent input commitments and the remaining card is an auxiliary one.[2] In other words, given two commitments to bits $a, b \in \{0,1\}$ along with one additional card, their protocol produces a commitment to $a \wedge b$ without leaking (revealing) any information about a and b:

$$\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\ \underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}}\ \boxed{5} \ \rightarrow \ \cdots \ \rightarrow \ \underbrace{\boxed{?}\boxed{?}}_{[a \wedge b]}.$$

In the (original) Niemi–Renvall protocol [18], the base of the output commitment is always $\{1,2\}$ and the expected number of required shuffles is 9.5; however, Koch, Schrempp, and Kirsten [6] demonstrated that the expected number of shuffles can be reduced to 7.5 if we allow the base of the output commitment to be either $\{1,4\}$, $\{1,5\}$, or $\{4,5\}$. In this paper, we refer this modified version as to the *Niemi–Renvall protocol*. As will be seen in Sect. 2.4, this protocol uses only a simple shuffle called a *random cut* (which is described in Sect. 2.2).

Later, Mizuki [10] proposed a finite two-input AND protocol with eight cards.[3] This protocol uses only a shuffle action called a *random bisection cut* (which is described in Sect. 2.3), and the number of required shuffles is four.

In 2021, Koch, Schrempp, and Kirsten [6] proposed a Las Vegas two-input AND protocol with four cards. This protocol uses only a random cut, and its expected number of shuffles is six.

1.3 Contribution

Let us consider how to securely compute the three-input AND function:

$$\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\ \underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}}\ \underbrace{\boxed{?}\boxed{?}}_{[c]^{\{5,6\}}} \cdots \ \rightarrow \ \cdots \ \rightarrow \ \underbrace{\boxed{?}\boxed{?}}_{[a \wedge b \wedge c]}.$$

[2] Niemi and Renvall [18] also provided Las Vegas protocols for the secure computations of XOR and copy.

[3] Mizuki [10] also presented finite XOR and copy protocols.

Table 1. The existing two-input AND protocols with a standard deck of cards

	# of cards	# of shuffles	finite?
Niemi & Renvall [18]	5	7.5 (exp.)	
Mizuki [10]	8	4	✓
Koch & Schrempp & Kirsten [6]	4	6 (exp.)	

Table 2. Three-input AND computations with a standard deck of cards

	# of cards	# of shuffles	finite?
Mizuki [10] (twice)	10	8	✓
Koch & Schrempp & Kirsten [6] (twice)	6	12 (exp.)	
Ours	**6**	**8.5 (exp.)**	

Computation of the three-input AND function can be performed by repeatedly applying one of the three existing protocols [6,10,18] twice: First, one must obtain a commitment to $a \wedge b$ by applying a two-input protocol, and then apply the protocol to the commitments to $a \wedge b$ and c. To perform this, the numbers of required cards and shuffles by Mizuki's protocol [10] or Koch et al.'s protocol [6] are listed in Table 2.

In this study, we solicit a more efficient computation of the three-input AND. The main contribution of this paper is the novelty of our protocol that directly performs a secure three-input AND computation in a single application. Our protocol is partly based on Niemi and Renvall's two-input AND protocol [18]. Our protocol uses only six cards, which are necessary for three input commitments; therefore, it uses the same number of cards as two runs of Koch et al.'s protocol [6]. The expected number of shuffles required for our protocol is 8.5, which is 3.5 fewer than the previous best six-card computation (see Table 2).

Overall, our proposed three-input AND protocol is "card-minimal" in the sense that only six cards are necessary under the encoding rule (1). Additionally, our protocol is simple because it uses only random cuts and random bisection cuts as shuffles.

1.4 Outline

This paper is organized as follows: Sect. 2 describes basic terminology in card-based cryptography and introduces the Niemi–Renvall protocol [18]. In Sect. 3, we describe our three-input AND protocol. We conclude this paper in Sect. 4.

2 Preliminaries

In this section, we introduce the actions involved in card-based protocols as well as practical shuffles, specifically the random cut and the random bisection cut, which will be used in our proposed protocol. Additionally, we explain the two-input AND protocol constructed by Niemi and Renvall [18].

2.1 Actions

In card-based protocols, as they have been formalized in [7,13,15], there are the following three main actions to be applied to a sequence of n cards.

Rearrangement. Apply some permutation $\pi \in S_n$ to a sequence of n cards, where S_n denotes the symmetric group of degree n. We write this action as (perm, π):

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

Turn. Turn over the t-th card (from the left) for $t \in \{1, \ldots, n\}$ in a sequence to check the number of the card. We write this action as $(\mathsf{turn}, \{t\})$:

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

In this example, the numbered card displaying a 1 was revealed.

Shuffle. Apply a permutation $\pi \in \mathit{\Pi}$ chosen uniformly randomly from a permutation set $\mathit{\Pi} \subseteq S_n$. We write this action as $(\mathsf{shuf}, \mathit{\Pi})$:

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

Note that it is not possible for an observer to know which permutation in $\mathit{\Pi}$ was applied.

2.2 Random Cut

A random cut is a shuffling action (denoted by $\langle\cdot\rangle$) that shifts a sequence of cards cyclically and randomly. If a random cut is applied to a sequence of n cards, then the resulting sequence becomes one of the following n sequences, each of which occurs with a probability of $1/n$:

$$\left\langle \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\cdots\overset{n-1}{\boxed{?}}\overset{n}{\boxed{?}} \right\rangle \rightarrow \begin{cases} \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\cdots\overset{n-1}{\boxed{?}}\overset{n}{\boxed{?}}, \\ \overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\cdots\overset{n}{\boxed{?}}\overset{1}{\boxed{?}}, \\ \qquad\vdots \\ \overset{n-1}{\boxed{?}}\overset{n}{\boxed{?}}\overset{1}{\boxed{?}}\cdots\overset{n-3}{\boxed{?}}\overset{n-2}{\boxed{?}}, \\ \overset{n}{\boxed{?}}\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\cdots\overset{n-2}{\boxed{?}}\overset{n-1}{\boxed{?}}\,. \end{cases}$$

This random cut can be written, using a cyclic permutation $\sigma = (1\,2\,3\,\cdots\,n)$, as

$$(\mathsf{shuf}, \{\mathsf{id}, \sigma, \sigma^2, \ldots, \sigma^{n-1}\}),$$

where id denotes the identity permutation. Hereinafter, we use $\mathsf{RC}_{1,2,\ldots,n}$ to represent $\{\mathsf{id}, \sigma, \sigma^2, \ldots, \sigma^{n-1}\}$.

A random cut can be easily performed by human hands, and a secure implementation called the Hindu cut is well known [31,32].

2.3 Random Bisection Cut

A random bisection cut is a shuffling action invented by Mizuki and Sone [16] in 2009. This shuffle (denoted by $[\,\cdot\,|\,\cdot\,]$) bisects a sequence of $2n$ cards and randomly swaps the two halves; the resulting sequence becomes one of the following two sequences:

$$\left[\overset{1}{\boxed{?}}\cdots\overset{n}{\boxed{?}}\,\middle|\,\overset{n+1}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}\right] \to \begin{cases} \overset{1}{\boxed{?}}\cdots\overset{n}{\boxed{?}}\overset{n+1}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}, \\ \overset{n+1}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}\overset{1}{\boxed{?}}\cdots\overset{n}{\boxed{?}}. \end{cases}$$

That is, the resulting sequence either remains the same as the original one or becomes a sequence where the two halves are swapped with a probability of 1/2. This random bisection cut can be written as follows:

$$(\mathsf{shuf}, \{\mathsf{id}, (1\ n{+}1)(2\ n{+}2)\cdots(n\ 2n)\}).$$

Secure implementations using familiar tools were shown in [31,32]. The random bisection cut has brought many efficient protocols (e.g., [11,12,14,19–21]).

2.4 The Niemi–Renvall Protocol

Given two commitments to $a, b \in \{0,1\}$ along with an additional card, the Niemi–Renvall protocol [18] outputs a commitment to $a \wedge b$. The procedure is as follows:

1. Place the two input commitments and an additional card $\boxed{5}$ and turn it over:

$$\boxed{5}\;\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\;\underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}} \quad\to\quad \boxed{?}\;\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\;\underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}}.$$

2. Rearrange the third and fourth cards:

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{5}{\boxed{?}} \quad\to\quad \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{4}{\boxed{?}}\overset{3}{\boxed{?}}\overset{5}{\boxed{?}}.$$

Now, let us consider the order of $\boxed{5}$, $\boxed{1}$, and $\boxed{4}$ in the rearranged sequence; the order is $\boxed{5} \to \boxed{4} \to \boxed{1}$ (apart from cyclic rotation) if and only if $(a, b) = (1, 1)$, i.e., $a \wedge b = 1$ (whereas the order is $\boxed{5} \to \boxed{1} \to \boxed{4}$ if and only if $a \wedge b = 0$). Therefore, we try to remove $\boxed{2}$ and $\boxed{3}$ in Steps 3 and 4.

Table 3. The sequence after removing $\boxed{2}$ and $\boxed{3}$

(a, b)	$a \wedge b$	Sequence after Removing $\boxed{2}$ and $\boxed{3}$
$(0, 0)$	0	$\boxed{1}\boxed{4}\boxed{5}$ or $\boxed{4}\boxed{5}\boxed{1}$ or $\boxed{5}\boxed{1}\boxed{4}$
$(0, 1)$	0	$\boxed{1}\boxed{4}\boxed{5}$ or $\boxed{4}\boxed{5}\boxed{1}$ or $\boxed{5}\boxed{1}\boxed{4}$
$(1, 0)$	0	$\boxed{1}\boxed{4}\boxed{5}$ or $\boxed{4}\boxed{5}\boxed{1}$ or $\boxed{5}\boxed{1}\boxed{4}$
$(1, 1)$	1	$\boxed{1}\boxed{5}\boxed{4}$ or $\boxed{4}\boxed{1}\boxed{5}$ or $\boxed{5}\boxed{4}\boxed{1}$

3. Apply a random cut to the sequence of all cards:

$$\langle \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?} \rangle \rightarrow \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}.$$

4. Turn over the first card; if it is $\boxed{2}$ or $\boxed{3}$, then remove it. Otherwise, turn it over. If there is still a $\boxed{2}$ or $\boxed{3}$ in the sequence, then return to Step 3.
5. The resulting sequence after removing $\boxed{2}$ and $\boxed{3}$ becomes one as presented in Table 3. Apply a random cut to the sequence and then reveal the first card to obtain the output commitment. (This step was developed by Koch et al. [6].)

$$\boxed{?}\boxed{?}\boxed{?} \xrightarrow{(\mathsf{turn},\{1\})} \begin{cases} \boxed{1}\ \underbrace{\boxed{?}\boxed{?}}_{[a\wedge b]^{\{4,5\}}}, \\ \boxed{4}\ \underbrace{\boxed{?}\boxed{?}}_{[\overline{a\wedge b}]^{\{1,5\}}}, \\ \boxed{5}\ \underbrace{\boxed{?}\boxed{?}}_{[a\wedge b]^{\{1,4\}}}. \end{cases}$$

If the first card is $\boxed{4}$, then we obtain a commitment to the negation of $a \wedge b$. In this case, we apply the NOT computation to obtain a commitment to $a \wedge b$.

Correctness of this protocol is clear from the above description. Regarding security, since a random cut is applied to the sequence in Step 3, the revealed card in Step 4 is chosen randomly from the sequence. Therefore, no information about the input is leaked. For example, if the number of cards in Step 3 is five, the revealed card in Step 4 should be one in $\{\boxed{1}, \boxed{2}, \boxed{3}, \boxed{4}, \boxed{5}\}$ with an equal probability regardless of the input values. In the same manner, no information about the input is leaked when the first card is revealed in Step 5. To summarize, this protocol is correct and secure.

3 Our Three-Input AND Protocol

In this section, we present a three-input AND protocol that requires no additional card, i.e., is card-minimal and uses fewer shuffles as compared to the applications of previous protocols:

$$\underbrace{\boxed{?}\,\boxed{?}}_{[a]^{\{1,2\}}}\;\underbrace{\boxed{?}\,\boxed{?}}_{[b]^{\{3,4\}}}\;\underbrace{\boxed{?}\,\boxed{?}}_{[c]^{\{5,6\}}} \quad\rightarrow\quad \cdots \quad\rightarrow\quad \underbrace{\boxed{?}\,\boxed{?}}_{[a\wedge b\wedge c]}.$$

We begin by describing the idea behind our proposed protocol.

3.1 Idea

Observe the following simple fact about $a \wedge b \wedge c$:

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

In other words, to construct a three-input AND protocol, it suffices to simulate a two-input AND protocol if $c = 1$ and to always output 0 if $c = 0$. For this, we borrow the idea behind the Niemi–Renvall protocol [18] introduced in Sect. 2.4.

Remember that their protocol swaps the third and fourth cards in Step 2; this action reverses the order of $\boxed{1}$ and $\boxed{4}$ if and only if $(a, b) = (1, 1)$, i.e., the output is 1. That is, if we skip this step and perform the remaining steps, then the output should be always 0. From this observation, it suffices to swap the third and fourth cards if and only if $c = 1$ and then perform the remaining steps of the protocol. Therefore, in the next subsection, we will describe how to swap two cards according to the value of c (without knowing it).

3.2 Swapping by Value of Commitment

For a sequence of two cards along with a commitment to $c \in \{0, 1\}$ whose base is $\{i, j\}$, we want to swap the cards if $c = 1$ and we want to keep them unchanged if $c = 0$, without leaking the value of c:

$$\overset{1}{\boxed{?}}\,\overset{2}{\boxed{?}}\;\underbrace{\boxed{?}\,\boxed{?}}_{[c]^{\{i,j\}}} \quad\rightarrow\quad \begin{cases} \overset{1}{\boxed{?}}\,\overset{2}{\boxed{?}} & \text{if } c = 0, \\ \overset{2}{\boxed{?}}\,\overset{1}{\boxed{?}} & \text{if } c = 1. \end{cases}$$

We call this the *swap operation* by the commitment to c. The swap operation proceeds as follows (whose procedure is similar to the Mizuki XOR protocol [10]).

1. Assume a target sequence of two cards and a commitment to c (with $i < j$):

$$\boxed{?}\,\boxed{?}\,\underbrace{\boxed{?}\,\boxed{?}}_{[c]^{\{i,j\}}}.$$

2. Rearrange the second and third cards:

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}} \rightarrow \overset{1}{\boxed{?}}\overset{3}{\boxed{?}}\overset{2}{\boxed{?}}\overset{4}{\boxed{?}}\,.$$

3. Apply a random bisection cut to the sequence of all cards:

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

4. Rearrange the second and third cards:

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}} \rightarrow \overset{1}{\boxed{?}}\overset{3}{\boxed{?}}\overset{2}{\boxed{?}}\overset{4}{\boxed{?}}\,.$$

Observe that, depending on the value of c, the current sequence satisfies the followings:

$$c = 0 \Rightarrow \begin{cases} \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\ \boxed{i}\boxed{j} & \text{(Prob. of } 1/2\text{)},\\ \overset{2}{\boxed{?}}\overset{1}{\boxed{?}}\ \boxed{j}\boxed{i} & \text{(Prob. of } 1/2\text{)}. \end{cases}$$

$$c = 1 \Rightarrow \begin{cases} \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\ \boxed{j}\boxed{i} & \text{(Prob. of } 1/2\text{)},\\ \overset{2}{\boxed{?}}\overset{1}{\boxed{?}}\ \boxed{i}\boxed{j} & \text{(Prob. of } 1/2\text{)}. \end{cases}$$

5. Turn over the third and fourth cards; $\boxed{i}\boxed{j}$ or $\boxed{j}\boxed{i}$ should appear with a probability of 1/2.
   (a) If $\boxed{i}\boxed{j}$ appear, then output the first and second cards.
   (b) If $\boxed{j}\boxed{i}$ appear, then swap the first and second cards and then output them.

3.3 Description of Our Protocol

We now present our three-input AND protocol.

1. Assume three input commitments to $a, b, c \in \{0, 1\}$:

$$\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\ \underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}}\ \underbrace{\boxed{?}\boxed{?}}_{[c]^{\{5,6\}}}\,.$$

2. Apply the swap operation by the commitment to c shown in Sect. 3.2 to the second and third cards, as follows.

(a) Rearrange the sequence:

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{5}{\boxed{?}}\overset{6}{\boxed{?}} \;\rightarrow\; \overset{1}{\boxed{?}}\overset{4}{\boxed{?}}\overset{2}{\boxed{?}}\overset{5}{\boxed{?}}\overset{3}{\boxed{?}}\overset{6}{\boxed{?}}\,.$$

(b) Apply a random bisection cut to the last four cards:

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

(c) Apply the inverse rearrangement of Step 2(a):

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{5}{\boxed{?}}\overset{6}{\boxed{?}} \;\rightarrow\; \overset{1}{\boxed{?}}\overset{3}{\boxed{?}}\overset{5}{\boxed{?}}\overset{2}{\boxed{?}}\overset{4}{\boxed{?}}\overset{6}{\boxed{?}}\,.$$

(d) Turn over the fifth and sixth cards: If $\boxed{5}\boxed{6}$ appear, then do nothing. If $\boxed{6}\boxed{5}$ appear, then swap the second and third cards.

3. Rearrange the sequence so that the first card becomes $\boxed{5}$ appeared in the previous step and then turn it over. Subsequently, apply Steps 3, 4, and 5 of the Niemi–Renvall protocol [18] to the first five cards.

3.4 Correctness and Security

Here, we prove the correctness and security of our three-input AND protocol. Observe that, in the sequence after Step 2 as listed in Table 4, the order of $\boxed{1}\boxed{4}$ is $\boxed{4} \rightarrow \boxed{1}$ if and only if $a \wedge b \wedge c = 1$ (whereas it is $\boxed{1} \rightarrow \boxed{4}$ if and only if $a \wedge b \wedge c = 0$). From this, it suffices to remove $\boxed{2}$ and $\boxed{3}$ in the same manner as the Niemi–Renvall protocol [18]. Therefore, our protocol can always output a commitment to $a \wedge b \wedge c$ by applying Steps 3, 4, and 5 of the Niemi–Renvall protocol [18].

More formally, we prove this by using the *KWH-tree* [7] as in Figs. 1 and 2. The KWH-tree is a tree-like diagram that shows the transitions of possible sequences of cards along with their respective polynomials in a box, where actions to be applied to the sequence are appended to an edge. In the figure, the probability of $(a, b, c) = (x, y, z)$ is denoted by X_{xyz}. A polynomial annotating a sequence in a box such as $1/2X_{000}$ represents the conditional probability that the current sequence is the one next to the polynomial, given what can be observed so far on the table.

If the sum of all the polynomials in each box is equal to

$$\sum_{x,y,z\in\{0,1\}} X_{xyz},$$

then it is guaranteed that no information about the input is leaked. The KWH-tree of our protocol for Steps 1 and 2 is shown in Fig. 1. The KWH-tree for Step 3 is shown in Fig. 2. From the figures, it can be easily confirmed that the aforementioned condition is satisfied in each box, i.e., our protocol is secure.

Table 4. The possible sequences after Step 2 in our protocol

$(a,\, b,\, c)$	$a \wedge b \wedge c$	Sequence After Step 2
$(0, 0, 0)$	0	$\boxed{1}\,\boxed{2}\,\boxed{3}\,\boxed{4}$
$(0, 0, 1)$	0	$\boxed{1}\,\boxed{3}\,\boxed{2}\,\boxed{4}$
$(0, 1, 0)$	0	$\boxed{1}\,\boxed{2}\,\boxed{4}\,\boxed{3}$
$(1, 0, 0)$	0	$\boxed{2}\,\boxed{1}\,\boxed{3}\,\boxed{4}$
$(0, 1, 1)$	0	$\boxed{1}\,\boxed{4}\,\boxed{2}\,\boxed{3}$
$(1, 0, 1)$	0	$\boxed{2}\,\boxed{3}\,\boxed{1}\,\boxed{4}$
$(1, 1, 0)$	0	$\boxed{2}\,\boxed{1}\,\boxed{4}\,\boxed{3}$
$(1, 1, 1)$	1	$\boxed{2}\,\boxed{4}\,\boxed{1}\,\boxed{3}$

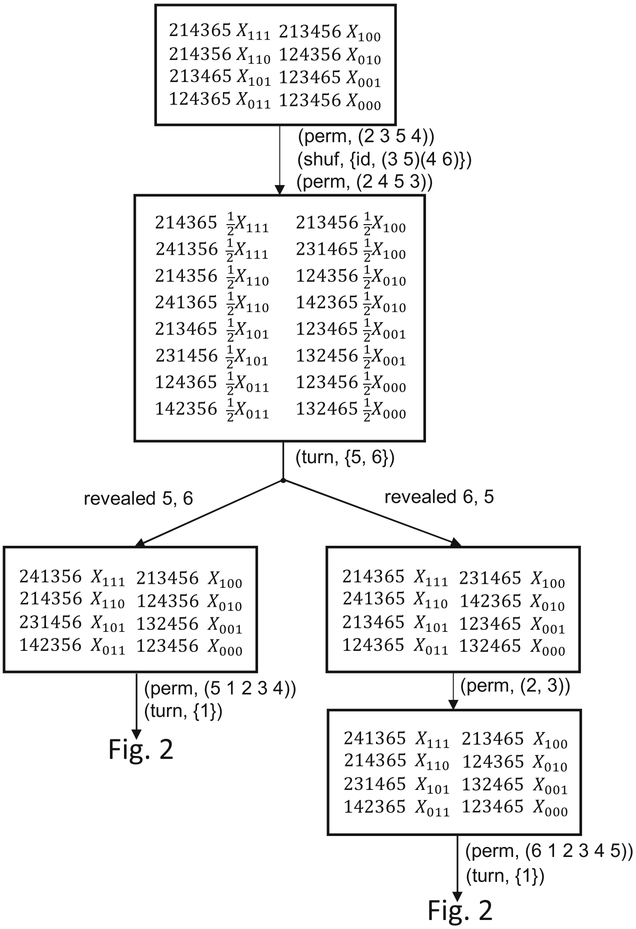
214365 X_{111} 213456 X_{100}
214356 X_{110} 124356 X_{010}
213465 X_{101} 123465 X_{001}
124365 X_{011} 123456 X_{000}
(perm, (2 3 5 4))
(shuf, {id, (3 5)(4 6)})
(perm, (2 4 5 3))
214365 $\frac{1}{2}X_{111}$ 213456 $\frac{1}{2}X_{100}$
241356 $\frac{1}{2}X_{111}$ 231465 $\frac{1}{2}X_{100}$
214356 $\frac{1}{2}X_{110}$ 124356 $\frac{1}{2}X_{010}$
241365 $\frac{1}{2}X_{110}$ 142365 $\frac{1}{2}X_{010}$
213465 $\frac{1}{2}X_{101}$ 123465 $\frac{1}{2}X_{001}$
231456 $\frac{1}{2}X_{101}$ 132456 $\frac{1}{2}X_{001}$
124365 $\frac{1}{2}X_{011}$ 123456 $\frac{1}{2}X_{000}$
142356 $\frac{1}{2}X_{011}$ 132465 $\frac{1}{2}X_{000}$
(turn, {5, 6})
revealed 5, 6
revealed 6, 5
241356 X_{111} 213456 X_{100}
214356 X_{110} 124356 X_{010}
231456 X_{101} 132456 X_{001}
142356 X_{011} 123456 X_{000}
(perm, (5 1 2 3 4))
(turn, {1})
Fig. 2
214365 X_{111} 231465 X_{100}
241365 X_{110} 142365 X_{010}
213465 X_{101} 123465 X_{001}
124365 X_{011} 132465 X_{000}
(perm, (2, 3))
241365 X_{111} 213465 X_{100}
214365 X_{110} 124365 X_{010}
231465 X_{101} 132465 X_{001}
142365 X_{011} 123465 X_{000}
(perm, (6 1 2 3 4 5))
(turn, {1})
Fig. 2

Fig. 1. The KWH-tree for our three-input AND protocol (Steps 1 to 2)

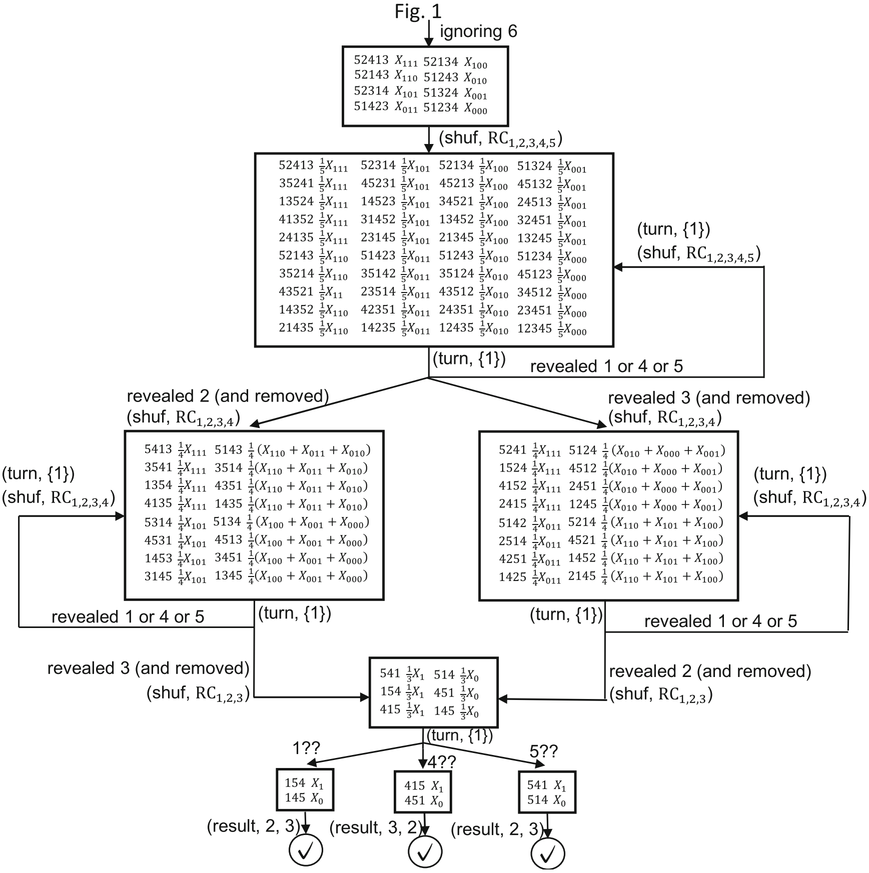
Fig. 1
ignoring 6
52413 X_{111} 52134 X_{100}
52143 X_{110} 51243 X_{010}
52314 X_{101} 51324 X_{001}
51423 X_{011} 51234 X_{000}
(shuf, $\mathrm{RC}_{1,2,3,4,5}$)
52413 $\frac{1}{5}X_{111}$ 52314 $\frac{1}{5}X_{101}$ 52134 $\frac{1}{5}X_{100}$ 51324 $\frac{1}{5}X_{001}$
35241 $\frac{1}{5}X_{111}$ 45231 $\frac{1}{5}X_{101}$ 45213 $\frac{1}{5}X_{100}$ 45132 $\frac{1}{5}X_{001}$
13524 $\frac{1}{5}X_{111}$ 14523 $\frac{1}{5}X_{101}$ 34521 $\frac{1}{5}X_{100}$ 24513 $\frac{1}{5}X_{001}$
41352 $\frac{1}{5}X_{111}$ 31452 $\frac{1}{5}X_{101}$ 13452 $\frac{1}{5}X_{100}$ 32451 $\frac{1}{5}X_{001}$
24135 $\frac{1}{5}X_{111}$ 23145 $\frac{1}{5}X_{101}$ 21345 $\frac{1}{5}X_{100}$ 13245 $\frac{1}{5}X_{001}$
52143 $\frac{1}{5}X_{110}$ 51423 $\frac{1}{5}X_{011}$ 51243 $\frac{1}{5}X_{010}$ 51234 $\frac{1}{5}X_{000}$
35214 $\frac{1}{5}X_{110}$ 35142 $\frac{1}{5}X_{011}$ 35124 $\frac{1}{5}X_{010}$ 45123 $\frac{1}{5}X_{000}$
43521 $\frac{1}{5}X_{11}$ 23514 $\frac{1}{5}X_{011}$ 43512 $\frac{1}{5}X_{010}$ 34512 $\frac{1}{5}X_{000}$
14352 $\frac{1}{5}X_{110}$ 42351 $\frac{1}{5}X_{011}$ 24351 $\frac{1}{5}X_{010}$ 23451 $\frac{1}{5}X_{000}$
21435 $\frac{1}{5}X_{110}$ 14235 $\frac{1}{5}X_{011}$ 12435 $\frac{1}{5}X_{010}$ 12345 $\frac{1}{5}X_{000}$
(turn, {1})
(shuf, $\mathrm{RC}_{1,2,3,4,5}$)
(turn, {1})
revealed 1 or 4 or 5
revealed 2 (and removed)
(shuf, $\mathrm{RC}_{1,2,3,4}$)
revealed 3 (and removed)
(shuf, $\mathrm{RC}_{1,2,3,4}$)
(turn, {1})
(shuf, $\mathrm{RC}_{1,2,3,4}$)
5413 $\frac{1}{4}X_{111}$ 5143 $\frac{1}{4}(X_{110}+X_{011}+X_{010})$
3541 $\frac{1}{4}X_{111}$ 3514 $\frac{1}{4}(X_{110}+X_{011}+X_{010})$
1354 $\frac{1}{4}X_{111}$ 4351 $\frac{1}{4}(X_{110}+X_{011}+X_{010})$
4135 $\frac{1}{4}X_{111}$ 1435 $\frac{1}{4}(X_{110}+X_{011}+X_{010})$
5314 $\frac{1}{4}X_{101}$ 5134 $\frac{1}{4}(X_{100}+X_{001}+X_{000})$
4531 $\frac{1}{4}X_{101}$ 4513 $\frac{1}{4}(X_{100}+X_{001}+X_{000})$
1453 $\frac{1}{4}X_{101}$ 3451 $\frac{1}{4}(X_{100}+X_{001}+X_{000})$
3145 $\frac{1}{4}X_{101}$ 1345 $\frac{1}{4}(X_{100}+X_{001}+X_{000})$
5241 $\frac{1}{4}X_{111}$ 5124 $\frac{1}{4}(X_{010}+X_{000}+X_{001})$
1524 $\frac{1}{4}X_{111}$ 4512 $\frac{1}{4}(X_{010}+X_{000}+X_{001})$
4152 $\frac{1}{4}X_{111}$ 2451 $\frac{1}{4}(X_{010}+X_{000}+X_{001})$
2415 $\frac{1}{4}X_{111}$ 1245 $\frac{1}{4}(X_{010}+X_{000}+X_{001})$
5142 $\frac{1}{4}X_{011}$ 5214 $\frac{1}{4}(X_{110}+X_{101}+X_{100})$
2514 $\frac{1}{4}X_{011}$ 4521 $\frac{1}{4}(X_{110}+X_{101}+X_{100})$
4251 $\frac{1}{4}X_{011}$ 1452 $\frac{1}{4}(X_{110}+X_{101}+X_{100})$
1425 $\frac{1}{4}X_{011}$ 2145 $\frac{1}{4}(X_{110}+X_{101}+X_{100})$
(turn, {1})
(shuf, $\mathrm{RC}_{1,2,3,4}$)
revealed 1 or 4 or 5
(turn, {1})
(turn, {1})
revealed 1 or 4 or 5
revealed 3 (and removed)
(shuf, $\mathrm{RC}_{1,2,3}$)
541 $\frac{1}{3}X_1$ 514 $\frac{1}{3}X_0$
154 $\frac{1}{3}X_1$ 451 $\frac{1}{3}X_0$
415 $\frac{1}{3}X_1$ 145 $\frac{1}{3}X_0$
revealed 2 (and removed)
(shuf, $\mathrm{RC}_{1,2,3}$)
(turn, {1})
1??
4??
5??
154 X_1
145 X_0
415 X_1
451 X_0
541 X_1
514 X_0
(result, 2, 3)
(result, 3, 2)
(result, 2, 3)
✓
✓
✓

Fig. 2. The KWH-tree for our three-input AND protocol (Step 3), where $X_0 = X_{000} + X_{001} + X_{010} + X_{011} + X_{100} + X_{101} + X_{110}$ and $X_1 = X_{111}$. The result action indicates the positions of the output commitment.

4 Conclusion

In this study, we designed a simple three-input AND protocol using playing cards by only using six cards; in other words, we do not need any additional cards (aside from three input commitments). To the best of our knowledge, this is the first type of protocol that can be used for directly computing a three-input Boolean function with a standard deck of cards.

A natural open problem that presents itself from our research is the construction of efficient AND protocols for more than three inputs. It would also be interesting to investigate whether a finite AND protocol can be constructed using only random cuts even for two inputs. Making use of a standard deck in the "private permutation" setting (e.g., [8,17,22,33]) would be another interesting topic. In addition, it would be worthwhile to construct zero-knowledge proof protocols working only on a standard deck for pencil puzzles, cf. [3,23,25,26].

Acknowledgements. We thank the anonymous referees, whose comments have helped us to improve the presentation of the paper. This work was supported in part by JSPS KAKENHI Grant Number JP19J21153.

References

1. Abe, Y., Hayashi, Y., Mizuki, T., Sone, H.: Five-card AND computations in committed format using only uniform cyclic shuffles. New Gener. Comput. **39**(1), 97–114 (2021). https://doi.org/10.1007/s00354-020-00110-2
2. Boer, B.: More efficient match-making and satisfiability *the five card trick*. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 208–217. Springer, Heidelberg (1990). https://doi.org/10.1007/3-540-46885-4_23
3. 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.) 8th International Conference on Fun with Algorithms (FUN 2016). Leibniz International Proceedings in Informatics (LIPIcs), vol. 49, pp. 8:1–8:20. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2016). https://doi.org/10.4230/LIPIcs.FUN.2016.8
4. Crépeau, C., Kilian, J.: Discreet solitary games. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 319–330. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48329-2_27
5. Kastner, J., et al.: The minimum number of cards in practical card-based protocols. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10626, pp. 126–155. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70700-6_5
6. Koch, A., Schrempp, M., Kirsten, M.: Card-based cryptography meets formal verification. New Gener. Comput. **39**(1), 115–158 (2021). https://doi.org/10.1007/s00354-020-00120-0
7. Koch, A., Walzer, S., Härtel, K.: Card-based cryptographic protocols using a minimal number of cards. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 783–807. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_32
8. Manabe, Y., Ono, H.: Secure card-based cryptographic protocols using private operations against malicious players. In: Maimut, D., Oprina, A.-G., Sauveron, D. (eds.) SecITC 2020. LNCS, vol. 12596, pp. 55–70. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-69255-1_5
9. 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
10. Mizuki, T.: Efficient and secure multiparty computations using a standard deck of playing cards. In: Foresti, S., Persiano, G. (eds.) CANS 2016. LNCS, vol. 10052, pp. 484–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48965-0_29
11. 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.) UCNC 2013. LNCS, vol. 7956, pp. 162–173. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39074-6_16

12. Mizuki, T., Komano, Y.: Analysis of information leakage due to operative errors in card-based protocols. In: Iliopoulos, C., Leong, H.W., Sung, W.-K. (eds.) IWOCA 2018. LNCS, vol. 10979, pp. 250–262. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94667-2_21
13. 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
14. 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
15. Mizuki, T., Shizuya, H.: Computational model of card-based cryptographic protocols and its applications. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **100**(1), 3–11 (2017). https://doi.org/10.1587/transfun.E100.A.3
16. 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
17. 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
18. Niemi, V., Renvall, A.: Solitaire zero-knowledge. Fundam. Inf. **38**(1,2), 181–188 (1999). https://doi.org/10.3233/FI-1999-381214
19. Nishida, T., Hayashi, Y., Mizuki, T., Sone, H.: Card-based protocols for any Boolean function. In: Jain, R., Jain, S., Stephan, F. (eds.) TAMC 2015. LNCS, vol. 9076, pp. 110–121. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-17142-5_11
20. Nishida, T., Hayashi, Y., Mizuki, T., Sone, H.: Securely computing three-input functions with eight cards. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **98**(6), 1145–1152 (2015). https://doi.org/10.1587/transfun.E98.A.1145
21. Nishida, T., Mizuki, T., Sone, H.: Securely computing the three-input majority function with eight cards. In: Dediu, A.-H., Martín-Vide, C., Truthe, B., Vega-Rodríguez, M.A. (eds.) TPNC 2013. LNCS, vol. 8273, pp. 193–204. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45008-2_16
22. Ono, H., Manabe, Y.: Card-based cryptographic logical computations using private operations. New Gener. Comput. **39**(1), 19–40 (2020). https://doi.org/10.1007/s00354-020-00113-z
23. Robert, L., Miyahara, D., Lafourcade, P., Mizuki, T.: Physical zero-knowledge proof for Suguru puzzle. In: Devismes, S., Mittal, N. (eds.) SSS 2020. LNCS, vol. 12514, pp. 235–247. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64348-5_19
24. Ruangwises, S., Itoh, T.: AND protocols using only uniform shuffles. In: van Bevern, R., Kucherov, G. (eds.) CSR 2019. LNCS, vol. 11532, pp. 349–358. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19955-5_30
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 (2020). https://doi.org/10.1007/s00354-020-00114-y
26. Ruangwises, S., Itoh, T.: Physical zero-knowledge proof for ripple effect. In: Uehara, R., Hong, S.-H., Nandy, S.C. (eds.) WALCOM 2021. LNCS, vol. 12635, pp. 296–307. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68211-8_24
27. 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

28. Shinagawa, K., Mizuki, T.: Card-based protocols using triangle cards. In: Ito, H., Leonardi, S., Pagli, L., Prencipe, G. (eds.) 9th International Conference on Fun with Algorithms (FUN 2018). Leibniz International Proceedings in Informatics (LIPIcs), vol. 100, pp. 31:1–31:13. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2018). https://doi.org/10.4230/LIPIcs.FUN.2018.31
29. Shinagawa, K., et al.: Secure computation protocols using polarizing cards. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **99**(6), 1122–1131 (2016)
30. Shinagawa, K., et al.: Card-based protocols using regular polygon cards. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **100**(9), 1900–1909 (2017)
31. Ueda, I., Miyahara, D., Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: Secure implementations of a random bisection cut. Int. J. Inf. Secur. **19**, 445–452 (2020). https://doi.org/10.1007/s10207-019-00463-w
32. 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.) TPNC 2016. LNCS, vol. 10071, pp. 58–69. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49001-4_5
33. Yasunaga, K.: Practical card-based protocol for three-input majority. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **103**(11), 1296–1298 (2020). https://doi.org/10.1587/transfun.2020EAL2025