Efficient and Secure Multiparty Computations Using a Standard Deck of Playing Cards[*]

Takaaki Mizuki

Cyberscience Center, Tohoku University,
6–3 Aramaki-Aza-Aoba, Aoba, Sendai 980–8578, Japan
tm-paper+cardstd[AT]g-mail.tohoku-university.jp

Abstract. It is known that secure multiparty computation can be performed using physical cards with identical backs, and numerous card-based cryptographic protocols have been proposed. Almost all existing protocols require multiple cards that have the same pattern on their face sides; thus, a standard deck of playing cards cannot be used for executing these protocols. However, there is one exception: Niemi and Renvall's protocols, proposed in 1999, can be used with standard playing cards. In this paper, we continue their efforts to improve secure multiparty computation using a standard deck of playing cards, and propose efficient AND, XOR, and copy protocols that require significantly fewer shuffles compared to previous protocols.

1 Introduction

Secure multiparty computation enables a group of players to learn only the value of a predetermined function of their private inputs (without revealing more information than necessary). Although such a cryptographic task is usually implemented digitally on computers and/or network systems, there is another research direction in which cryptographic protocols are implemented physically (e.g. [4, 6]). In this paper, we consider the use of a deck of physical cards. In fact, it is known that secure multiparty computation can be conducted using physical cards with identical backs (such as $\boxed{?}$), and numerous card-based cryptographic protocols have been designed. Almost all existing protocols use cards whose face sides have a pattern such as black $\boxed{\clubsuit}$ or red $\boxed{\heartsuit}$; further, multiple cards having the same pattern are necessary (e.g., [1, 2, 7, 8, 11, 12, 14–16]). This paper begins with a brief introduction to such protocols.

1.1 Mainstream Card-based Protocols

Most card-based protocols manipulate Boolean values based on the following encoding:

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

[*] This paper appears in Proceedings of CANS 2016. The final publication is available at link.springer.com.

That is, considering a pair of black and red cards, postulate that bit value 0 represents the left card being black, and bit value 1 represents the left card being red. Based on this encoding rule (1), input is given to a card-based protocol. For example, the secure NOT computation, which is the simplest protocol, receives a pair of face-down cards equaling the value of input bit $a \in \{0,1\}$ (which is called a *commitment* to a), and reverses their order to obtain a commitment to negation $\bar{a}$:

$$\underbrace{\boxed{?}\boxed{?}}_{a} \to \overbrace{\boxed{?}\boxed{?}}^{\rightleftharpoons} \to \underbrace{\boxed{?}\boxed{?}}_{\bar{a}}.$$

Another example: given commitments to input bits a and b

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

a protocol for secure AND computation outputs a commitment to $a \wedge b$

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

without revealing any information about the values of a and b after applying a predetermined series of operations such as shuffling, rearranging, and turning over cards [2, 7, 11, 12, 16].

One of the efficient AND protocols works with two additional cards [11]:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{b}\boxed{\clubsuit}\boxed{\heartsuit} \to \cdots \to \underbrace{\boxed{?}\boxed{?}}_{a \wedge b}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit};$$

during the protocol's execution, several operations are performed, among them a shuffling operation called a *random bisection cut* (the details of which will be introduced in Section 3.1) is applied once. In regard to XOR computation, it is known that one random bisection cut enables a secure XOR to be performed without any additional cards [11]:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{b} \to \cdots \to \underbrace{\boxed{?}\boxed{?}}_{a \oplus b}\boxed{\clubsuit}\boxed{\heartsuit}.$$

Furthermore, making two copied commitments can be achieved with four additional cards and one random bisection cut [11]:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit} \to \cdots \to \underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{a}\boxed{\clubsuit}\boxed{\heartsuit}.$$

There are also other protocols designed for specific functions such as the adder [14] and 3-variable Boolean functions [15].

Because the above-mentioned protocols require multiple cards having the same pattern (such as $\boxed{\clubsuit}$ and $\boxed{\heartsuit}$), a standard deck of playing cards, unfortunately, cannot be utilized to execute these protocols. (Note that each card in a standard deck has a unique pattern on its face side, namely its suit and number.)

1.2 Use of a Standard Deck of Playing Cards

As seen thus far, almost all existing protocols do not work with a standard deck of playing cards. However, there is one exception: Niemi and Renvall's protocols [13] proposed in 1999 can be executed with the use of a normal deck of playing cards.

A standard, commercially available deck of playing cards consists of 52 cards (excluding jokers). Each card's face has a unique pattern (its suit and number), and hence we can easily create a total order on the set of these 52 cards. Therefore, hereafter we assume the following deck of 52 cards:

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

where, of course, the backs of all cards are identical $\boxed{?}$.

Similar to encoding rule (1) mentioned before, Niemi and Renvall [13] considered an encoding rule based on which of two cards is smaller or larger. That is, for any two cards $\boxed{i}\boxed{j}$ with $1 \leq i < j \leq 52$, they define the encoding rule as:

$$\boxed{i}\boxed{j} = 0, \quad \boxed{j}\boxed{i} = 1. \tag{2}$$

Thus, 0 represents the left card being smaller, and 1 represents the left card being larger. We can naturally consider a commitment as well, and throughout this paper, a commitment to bit x using two cards $\boxed{i}\boxed{j}$ is written as

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

where we call such a set $\{i,j\}$ a *base* of the commitment. For example,

$$\underbrace{\boxed{?}\boxed{?}}_{[x]^{\{1,2\}}}$$

is a commitment of base $\{1,2\}$; when we turn over these two cards, the order $\boxed{1}\boxed{2}$ implies $x = 0$, and $\boxed{2}\boxed{1}$ implies $x = 1$. Under this encoding rule, reversing the order of two cards constituting a commitment also corresponds to the NOT computation.

Based on encoding rule (2), Niemi and Renvall designed a protocol for realizing the following as a secure AND computation with five cards [13]:

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

During the protocol's execution, a random cut, which represents a cyclic shuffle, is applied an average of 9.5 times.

Regarding XOR computation, they showed that, on average, seven random cuts would provide the following result with four cards [13]:

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

Furthermore, as for copying a commitment, 5.5 random cuts suffice to realize the following with six cards [13]:

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

The details of these three existing protocols will be introduced in Section 2.

1.3 Our Results

In this paper, we focus on secure multiparty computation using a standard deck of playing cards (as introduced in Section 1.2), and enhance the efficiency. That is, we propose efficient AND, XOR, and copy protocols. As seen later, our three protocols will be constructed partially based on the ideas behind the mainstream card-based protocols [11] that use custom-made cards $\boxed{\clubsuit}\boxed{\heartsuit}$ and random bisection cuts.

Table 1 indicates the performance of our three protocols. As shown by the table data, our protocols require significantly fewer shuffles. Specifically, whereas the existing protocol requires an average of 9.5 shuffles for AND computation, our protocol terminates after applying exactly 4 shuffles. As for both XOR computation and secure copy, our protocols require only one shuffle. Because the "cost" of a card-based protocol comes mainly from shuffling operations in general, reducing the number of required shuffles is very important. (Note that the "cost" would be directly linked to human motivation to execute a protocol practically.) Further, whereas the existing protocols are so-called Las Vegas algorithms that require an average number of trials to be conducted, our protocols always terminate after applying a fixed number of shuffles.

Table 1. Performance comparison between existing protocols and our protocols.

	# of	# of shuffles		
	cards	avg.	fixed	total
○ AND computation				
Niemi-Renvall [13] (§2.2)	5	9.5	0	9.5
Ours (§3)	8	0	4	4
○ XOR computation				
Niemi-Renvall [13] (§2.3)	4	7	0	7
Ours (§4)	4	0	1	1
○ Secure copy				
Niemi-Renvall [13] (§2.4)	6	4.5	1	5.5
Ours (§5)	6	0	1	1

Furthermore, our protocols utilize random bisection cuts, whereas existing protocols use random cuts. The details will be discussed in the succeeding sections. Although the random bisection cut may be an unfamiliar shuffling operation, humans can easily implement a random bisection cut that is similar to a random cut (as will be seen in Section 3.1).

When considering the number of required cards, our protocols work with the same number of cards as the existing protocols for both XOR computation and secure copy; however, for AND computation, our protocol requires three more cards than the existing protocol. This might be perceived as a disadvantage; however, we believe that such a three-card increase would not be an issue, because card players can use the 52 cards as they like after they buy a standard deck of playing cards at a toy store.

The remainder of this paper is organized as follows. First, in Section 2, we introduce the details of the existing protocols. Then, in Section 3, we propose an efficient AND protocol. Next, we describe an efficient XOR protocol in Section 4, and an efficient copy protocol in Section 5. Finally, the paper is concluded in Section 6.

2 Niemi-Renvall Protocols

In this section, we introduce the details of the three protocols provided by Niemi and Renvall [13]. As preliminary information, we first introduce the random cut and its application to card searching in Section 2.1. Then, we explain the AND protocol in Section 2.2, the XOR protocol in Section 2.3, and the copy protocol in Section 2.4.

2.1 Random Cuts and Search for Cards

As mentioned previously, a random cut represents a cyclic shuffle; given a sequence of cards, it shifts their positions randomly without changing the order apart from cyclic rotation. For instance, consider five cards $\boxed{1}\boxed{2}\boxed{3}\boxed{4}\boxed{5}$ placed with their faces down (on a table) in this order:

$$\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?};$$

then, applying a random cut results in one of the following five sequences (if the table had eyes):

$$\boxed{1}\boxed{2}\boxed{3}\boxed{4}\boxed{5}\quad\boxed{5}\boxed{1}\boxed{2}\boxed{3}\boxed{4}\quad\boxed{4}\boxed{5}\boxed{1}\boxed{2}\boxed{3}\quad\boxed{3}\boxed{4}\boxed{5}\boxed{1}\boxed{2}\quad\boxed{2}\boxed{3}\boxed{4}\boxed{5}\boxed{1},$$

where each case occurs with a probability of exactly 1/5.

It is known that Humans are able to implement a random cut easily [18].

Next, as an application of the random cut, we explain a technique to search for designated cards. For example, assume that five cards from $\{1, 2, 3, 4, 5\}$ are placed with their faces down:

$$\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?},$$

and that their order is unknown, i.e., we do not know which one of 5! possible orders the sequence represents. Now, suppose that we want to search for card $\boxed{2}$. To this end, we apply a random cut to the sequence of five cards, and then reveal the first card (counting from the left). Unless the face-up card is $\boxed{2}$, turn over the card, apply a random cut, and reveal the first card again. Repeating this, we obtain the following after an average of five trials:

$$\boxed{2}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\,.$$

Note that the order of the sequence following $\boxed{2}$ has not changed, apart from the cyclic rotation, and that no information has leaked other than the fact that $\boxed{2}$ is the first card.

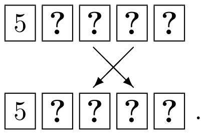
Generalizing this, given a sequence of face-down cards from a set $C \subseteq \{1, 2, \ldots, 52\}$ together with target cards $S \subseteq C$, we can find a card contained in S after applying an average of $|C|/|S|$ random cuts. As we soon show, Niemi and Renvall's protocols frequently use this random-cut-based search as their sub-protocol.

2.2 AND computation

Here, we elaborate Niemi and Renvall's AND protocol. The protocol uses five cards $\boxed{1}\boxed{2}\boxed{3}\boxed{4}\boxed{5}$. The first four cards are utilized for commitments to bit a and b, and the remaining $\boxed{5}$ is an additional card; thus, input to the protocol is:

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

Now, consider the following rearrangement of the sequence of input cards:



The four face-down cards would be in one of these four possible sequences depending on values (a, b):

(a, b)	seq. of cards
$(0, 0)$	$\boxed{5}\boxed{1}\boxed{3}\boxed{2}\boxed{4}$
$(0, 1)$	$\boxed{5}\boxed{1}\boxed{4}\boxed{2}\boxed{3}$
$(1, 0)$	$\boxed{5}\boxed{2}\boxed{3}\boxed{1}\boxed{4}$
$(1, 1)$	$\boxed{5}\boxed{2}\boxed{4}\boxed{1}\boxed{3}$

.

Suppose here that we could somehow delete both cards $\boxed{2}$ and $\boxed{3}$:

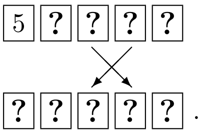
(a,b)	seq. of cards
$(0,0)$	$\boxed{5}\boxed{1}\boxed{}\boxed{}\boxed{4}$
$(0,1)$	$\boxed{5}\boxed{1}\boxed{4}\boxed{}\boxed{}$
$(1,0)$	$\boxed{5}\boxed{}\boxed{}\boxed{1}\boxed{4}$
$(1,1)$	$\boxed{5}\boxed{}\boxed{4}\boxed{1}\boxed{}$

;

then, one can easily notice that only when $(a,b) = (1,1)$, i.e., $a \wedge b = 1$, the order would be $\boxed{5}\boxed{4}\boxed{1}$; when $a \wedge b = 0$, it would be $\boxed{5}\boxed{1}\boxed{4}$. Therefore, this implies that we could obtain

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

Based on this idea, an AND protocol is constructed immediately.

1. For input sequence
$$\boxed{5}\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}},$$
turn over card $\boxed{5}$ and rearrange the sequence as:


2. Using the random-cut-based search (explained in Section 2.1), find card $\boxed{2}$ or $\boxed{3}$, and then discard it. This step requires an average of $5/2 = 2.5$ trials.
3. For the sequence of the remaining four cards, find the card, $\boxed{2}$ or $\boxed{3}$, that has not been found at the previous step, and then discard it. This step requires an average of $4/1 = 4$ trials.
4. For the sequence of the remaining three cards, using the random-cut-based search, find card $\boxed{5}$ to obtain a commitment to $a \wedge b$:
$$\boxed{5}\ \underbrace{\boxed{?}\boxed{?}}_{[a\wedge b]^{\{1,4\}}}\ .$$
This step requires an average of $3/1 = 3$ trials.

This is Niemi and Renvall's AND protocol, which requires $2.5 + 4 + 3 = 9.5$ random cuts on average.

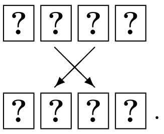
2.3 XOR Computation

Here, we explain Niemi and Renvall's XOR protocol, which requires no additional cards, and performs an XOR computation using only input commitments.

1. For input sequence

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

   rearrange it as:

   Now, similar to the case of the AND protocol above, the sequence of these four cards is the same as the left below; if we could somehow delete card $\boxed{3}$, then it would be the same as the right:

(a,b)	seq. of cards		seq. of cards
$(0,0)$	$\boxed{1}\boxed{3}\boxed{2}\boxed{4}$		$\boxed{1}\boxed{}\boxed{2}\boxed{4}$
$(0,1)$	$\boxed{1}\boxed{4}\boxed{2}\boxed{3}$	⇒	$\boxed{1}\boxed{4}\boxed{2}\boxed{}$
$(1,0)$	$\boxed{2}\boxed{3}\boxed{1}\boxed{4}$		$\boxed{2}\boxed{}\boxed{1}\boxed{4}$
$(1,1)$	$\boxed{2}\boxed{4}\boxed{1}\boxed{3}$		$\boxed{2}\boxed{4}\boxed{1}\boxed{}$

   .

   Note that if we cyclically shift the three cards so that $\boxed{4}$ is the first card, then the two cards following $\boxed{4}$ would be a commitment to $a \oplus b$ of base $\{1,2\}$.
2. Using the random-cut-based search, find card $\boxed{3}$, and then discard it. This step requires an average of four trials.
3. Using the random-cut-based search, find card $\boxed{4}$, and obtain a commitment to $a \oplus b$:

   $$\boxed{4}\;\underbrace{\boxed{?}\boxed{?}}_{[a\oplus b]^{\{1,2\}}}\;.$$

   This step requires an average of three trials.

This is Niemi and Renvall's XOR protocol, which requires $4+3=7$ random cuts on average.

2.4 Secure Copy

Here, we explain Niemi and Renvall's copy protocol. The protocol makes two copied commitments to input bit a with four additional cards.

1. For input sequence

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

turn over cards $\boxed{3}\boxed{4}$, and apply a random cut[1] to the two face-down cards to create a commitment to a uniformly distributed random bit r:

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

Turn over cards $\boxed{5}$ and $\boxed{6}$ as well:

$$\boxed{?}\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\boxed{?}\underbrace{\boxed{?}\boxed{?}}_{[r]^{\{3,4\}}}.$$

2. Using the random-cut-based search, find a card in $\{1,2,3,4\}$, and reveal the fourth card. (This requires $6/4 = 1.5$ trials on average.) For instance, if we found $\boxed{1}$, we have either

$$\boxed{1}\boxed{?}\boxed{?}\boxed{3}\boxed{?}\boxed{?} \text{ or } \boxed{1}\boxed{?}\boxed{?}\boxed{4}\boxed{?}\boxed{?}.$$

If the two face-up cards are from either $\{1,3\}$ or $\{2,4\}$, then $r = a$. Otherwise, i.e., they are from either $\{1,4\}$ or $\{2,3\}$, $\bar{r} = a$.

3. Turn over the two face-up cards, and find a card in $\{5,6\}$ using the random-cut-based search. (This requires $6/2 = 3$ trials on average.) In this case, we have

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

Apply the NOT computation to the commitment to r in the case of $\bar{r} = a$. Thus, in either case, we obtain

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

This is Niemi and Renvall's copy protocol, which requires one fixed number of a random cut together with $1.5 + 3 = 4.5$ random cuts on average.

3 Our AND Protocol

In this section, we propose an efficient AND protocol. Whereas Niemi and Renvall's AND protocol requires an average of 9.5 random cuts (as seen in Section 2.2), our protocol requires exactly four random bisection cuts.

As preliminary information, we first introduce the random bisection cut [11] in Section 3.1. Then, in Section 3.2, we propose a method for changing the base of a commitment using a random bisection cut. Next, in Section 3.3, we introduce an "opaque commitment pair," which is a new concept. Finally, we present our protocol in Section 3.4.

[1] Because there are only two cards here, it is just a shuffle.

3.1 Random Bisection Cuts

The random bisection cut is a shuffling operation that was proposed in 2009 [11]. Since the random bisection cut appeared, the performance of card-based protocols has increased significantly (e.g., [8, 11, 14, 15]). As described later, this paper applies random bisection cuts to a standard deck of playing cards to provide efficient protocols.

In a random bisection cut, a given sequence of cards is bisected, and then the two portions are switched (or not) with a probability of 1/2. For example, consider four cards $\boxed{1}\boxed{2}\boxed{3}\boxed{4}$ placed with their faces down in this order:

$$\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{1,2\}}}\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{3,4\}}}.$$

Apply a random bisection cut (denoted by $[\,\cdot\,|\,\cdot\,]$) to the sequence:

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

Then, the sequence of these four cards will be either

$$\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{1,2\}}}\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{3,4\}}} \quad \text{or} \quad \underbrace{\boxed{?}\boxed{?}}_{[0]^{\{3,4\}}}\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{1,2\}}},$$

where each case occurs with a probability of exactly 1/2.

Similar to the random cut, it is known that the random bisection cut can be easily performed by humans [18].

3.2 Change of Base

Here, we propose a method for changing the base of a commitment using a random bisection cut.

Take a commitment of base $\{1, 2\}$

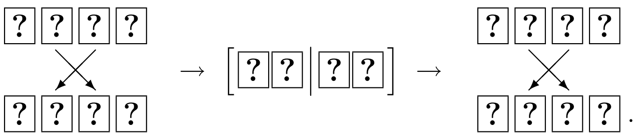
$$\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}$$

as an example, and assume that we have other cards $\boxed{3}\boxed{4}$. We want to convert the base into $\{3, 4\}$; of course, we do not want to reveal the value of bit a. The following procedure achieves this.

1. Turn over $\boxed{3}\boxed{4}$ so that they become a commitment to 0:

$$\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{3,4\}}}.$$

2. Rearrange the sequence, apply a random bisection cut, and rearrange it again:



Then, we have

$$\underbrace{\boxed{?}\boxed{?}}_{[a\oplus r]^{\{1,2\}}}\ \underbrace{\boxed{?}\boxed{?}}_{[r]^{\{3,4\}}}$$

where r is a uniformly distributed random bit.

3. Reveal the first two cards; then, we know whether $r = a$ or $r = \bar{a}$, and hence we have

$$\boxed{1}\boxed{2}\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{3,4\}}} \text{ or } \boxed{2}\boxed{1}\underbrace{\boxed{?}\boxed{?}}_{[\bar{a}]^{\{3,4\}}}.$$

(In the latter case, apply the NOT computation to the commitment to transform it into a commitment to a.)

Note that, because r is random, the information about a does not leak even if the commitment to $a \oplus r$ is revealed.

Thus, the base of a given commitment can be easily changed.

Our AND protocol utilizes this change-of-base method. The method seems quite useful because any base can be assigned to a given commitment whose base is unknown. In addition, the method can be used for detecting irregular cards such as jokers among two face-down cards placed as an input commitment. (This is a similar idea to that behind the checking-input method designed for custom-made cards $\boxed{\clubsuit}\boxed{\heartsuit}$ [10].)

3.3 Opaque Commitment Pair

Here, we consider a situation in which the base of a commitment is opaque. Now, assume that there are two commitments under $\{1, 2, 3, 4\}$:

$$\underbrace{\boxed{?}\boxed{?}}_{[a]^{B_1}}\underbrace{\boxed{?}\boxed{?}}_{[b]^{B_2}}$$

where we do not know which base is $\{1, 2\}$. That is, we cannot determine whether (i) $B_1 = \{1, 2\}$ and $B_2 = \{3, 4\}$, or (ii) $B_1 = \{3, 4\}$ and $B_2 = \{1, 2\}$ (the probabilities of events (i) and (ii) are both $1/2$, and these events are independent of any input values). We call two such commitments an *opaque commitment pair*, and write it as

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

Given an opaque commitment pair, if the base of one of the two commitments is found, then the base of the other commitment is also determined. For instance, for the above opaque commitment pair, if we turn over the first commitment (to a) and know that its base was $\{1,2\}$, then the base of the commitment to b is determined, and hence we have

$$\underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}}.$$

Furthermore, assume that there are a commitment to b and an opaque commitment pair

$$\underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}}\quad\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{5,6\},\{7,8\}}}\quad\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{5,6\},\{7,8\}}}\quad.$$

Then, we can make the base of the commitment to b opaque. That is, applying the change-of-base method shown in Section 3.2 to the first and third commitments results in

$$\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{5,6\},\{7,8\}}}\quad\underbrace{\boxed{?}\boxed{?}}_{[b]^{\{5,6\},\{7,8\}}}\quad.$$

3.4 Description of Our Protocol

Now, we are ready to present our AND protocol. The protocol performs a secure AND computation using eight cards, as follows.

1. Arrange input commitments and two commitments to 0:

$$\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}}\boxed{5}\boxed{6}\boxed{7}\boxed{8}\quad\rightarrow\quad\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}}\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{5,6\}}}\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{7,8\}}}.$$

2. Apply a random bisection cut to the third and fourth commitments:

$$\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}}\quad\left[\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\right];$$

then, we have an opaque commitment pair:

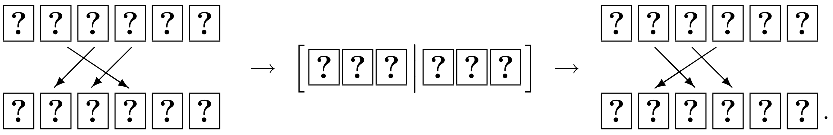
$$\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\underbrace{\boxed{?}\boxed{?}}_{[b]^{\{3,4\}}}\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{5,6\},\{7,8\}}}\quad\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{5,6\},\{7,8\}}}\quad.$$

3. Apply the change-of-base method presented in Section 3.2 to the second and fourth commitments:

$$\underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\underbrace{\boxed{?}\boxed{?}}_{[0]^{\{5,6\},\{7,8\}}}\quad\underbrace{\boxed{?}\boxed{?}}_{[b]^{\{5,6\},\{7,8\}}}\quad.$$

(From here up through step 5, simulate the AND protocol [11] that is based on custom-made cards $\boxed{\clubsuit}\boxed{\heartsuit}$.)

4. For the sequence of these six cards, apply rearrangements and a random bisection cut as:



Then, we have either

$$\text{(i)}\ \underbrace{\boxed{?}\boxed{?}}_{[a]^{\{1,2\}}}\ \underbrace{\boxed{?}\boxed{?}}_{[0]^{\{5,6\},\{7,8\}}}\ \underbrace{\boxed{?}\boxed{?}}_{[b]^{\{5,6\},\{7,8\}}} \quad \text{or} \quad \text{(ii)}\ \underbrace{\boxed{?}\boxed{?}}_{[\bar{a}]^{\{1,2\}}}\ \underbrace{\boxed{?}\boxed{?}}_{[b]^{\{5,6\},\{7,8\}}}\ \underbrace{\boxed{?}\boxed{?}}_{[0]^{\{5,6\},\{7,8\}}}\ ,$$

where each case occurs with a probability of exactly 1/2.

5. Reveal the first two cards.
   (a) Assume that the two face-up cards are $\boxed{1}\boxed{2}$. Then, in the case of (i) above, we have $a = 0$, and hence $a \wedge b = 0$ and $\bar{a} \wedge b = b$. In the case of (ii), we have $a = 1$, and hence $a \wedge b = b$ and $\bar{a} \wedge b = 0$. Therefore, in either case, we have

$$\boxed{1}\boxed{2}\ \ \underbrace{\boxed{?}\boxed{?}}_{[a\wedge b]^{\{5,6\},\{7,8\}}}\ \ \underbrace{\boxed{?}\boxed{?}}_{[\bar{a}\wedge b]^{\{5,6\},\{7,8\}}}\ .$$

   (b) Assume that the two face-up cards are $\boxed{2}\boxed{1}$. Similarly, we have

$$\boxed{2}\boxed{1}\ \ \underbrace{\boxed{?}\boxed{?}}_{[\bar{a}\wedge b]^{\{5,6\},\{7,8\}}}\ \ \underbrace{\boxed{?}\boxed{?}}_{[a\wedge b]^{\{5,6\},\{7,8\}}}\ .$$

6. After applying a random bisection cut (namely, a shuffle) to the commitment to $\bar{a} \wedge b$, reveal it to find the base of the commitment to $a \wedge b$; then, we obtain

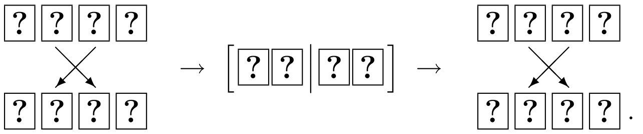
$$\underbrace{\boxed{?}\boxed{?}}_{[a\wedge b]^{\{5,6\}}} \quad \text{or} \quad \underbrace{\boxed{?}\boxed{?}}_{[a\wedge b]^{\{7,8\}}}\ .$$

This is our AND protocol, which uses four random bisection cuts in total. At step 5, although we reveal the first commitment, no information about bit a leaks because both (i) and (ii) occur with a probability of 1/2.

We can easily give a more formal proof of the security by using the "Koch-Walzer-Härtel diagram [7]" although we omit it due to the page limitation.

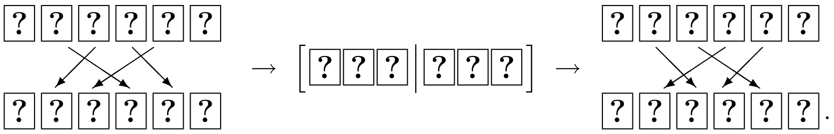
4 Our XOR Protocol

In this section, we propose an efficient XOR protocol. Whereas Niemi and Renvall's XOR protocol requires an average of seven random cuts (as seen in Section 2.3), our protocol terminates after only one random bisection cut. The protocol is obtained by simulating the XOR protocol [11] (which is based on custom-made cards $\boxed{\clubsuit}\boxed{\heartsuit}$).

1. Arrange two commitments:
$$\underbrace{\boxed{?}\,\boxed{?}}_{[a]^{\{1,2\}}}\;\underbrace{\boxed{?}\,\boxed{?}}_{[b]^{\{3,4\}}}.$$
2. Rearrange the sequence, apply a random bisection cut, and rearrange it again:

Then, we have
$$\underbrace{\boxed{?}\,\boxed{?}}_{[a\oplus r]^{\{1,2\}}}\;\;\underbrace{\boxed{?}\,\boxed{?}}_{[b\oplus r]^{\{3,4\}}}$$
where r is a uniformly distributed random bit.
3. Reveal the first commitment; then, we have
$$\boxed{1}\,\boxed{2}\;\;\underbrace{\boxed{?}\,\boxed{?}}_{[a\oplus b]^{\{3,4\}}}\;\;\text{or}\;\;\boxed{2}\,\boxed{1}\;\;\underbrace{\boxed{?}\,\boxed{?}}_{[\overline{a\oplus b}]^{\{3,4\}}}\;\;.$$

5 Our Copy Protocol

In this section, we propose an efficient copy protocol. Whereas Niemi and Renvall's copy protocol requires an average of 5.5 random cuts (as seen in Section 2.4), our protocol terminates after only one random bisection cut. The protocol is obtained by simulating the copy protocol [11] as well.

1. Arrange an input commitment and two commitments to 0:
$$\underbrace{\boxed{?}\,\boxed{?}}_{[a]^{\{1,2\}}}\boxed{3}\,\boxed{4}\,\boxed{5}\,\boxed{6}\;\;\rightarrow\;\;\underbrace{\boxed{?}\,\boxed{?}}_{[a]^{\{1,2\}}}\;\underbrace{\boxed{?}\,\boxed{?}}_{[0]^{\{3,4\}}}\;\underbrace{\boxed{?}\,\boxed{?}}_{[0]^{\{5,6\}}}.$$
2. Apply rearrangements and a random bisection cut as:

Then, we have
$$\underbrace{\boxed{?}\,\boxed{?}}_{[a\oplus r]^{\{1,2\}}}\;\;\underbrace{\boxed{?}\,\boxed{?}}_{[r]^{\{3,4\}}}\;\underbrace{\boxed{?}\,\boxed{?}}_{[r]^{\{5,6\}}}$$
where r is a uniformly distributed random bit.
3. Reveal the first commitment; then, we have
$$\boxed{1}\,\boxed{2}\underbrace{\boxed{?}\,\boxed{?}}_{[a]^{\{3,4\}}}\;\underbrace{\boxed{?}\,\boxed{?}}_{[a]^{\{5,6\}}}\;\;\text{or}\;\;\boxed{2}\,\boxed{1}\underbrace{\boxed{?}\,\boxed{?}}_{[\bar{a}]^{\{3,4\}}}\;\underbrace{\boxed{?}\,\boxed{?}}_{[\bar{a}]^{\{5,6\}}}.$$

6 Conclusion

Although almost all existing card-based protocols cannot be executed with a standard deck of playing cards, there is one exception: Niemi and Renvall's protocols [13] achieve secure AND, XOR, and copy computations using normal playing cards. In this paper, we continued their efforts to improve card-based protocols that use a standard deck of playing cards, and proposed efficient AND, XOR, and copy protocols. Our protocols were constructed by applying random bisection cuts [11] to a standard deck of playing cards; as a result, we succeeded in significantly reducing the number of required shuffles. Specifically, for AND computation, whereas the existing protocol requires an average of 9.5 random cuts, our protocol terminates after applying exactly four random cuts. Regarding XOR computation and copy, the existing protocols require an average of seven and 5.5 random cuts, respectively; in contrast, our protocols require only one random bisection cut.

An intriguing future work might involve finding lower bounds on the number of required cards and shuffles. It should be noted that there is a formalization for the card-based computation model [9]; a standard deck of playing cards is within the model. Therefore, to obtain lower bounds, the existing formalization could be useful.

The card-based protocol is easy to understand. By combining our AND, XOR, and copy protocols, any function can be securely computed using a commercially available deck of cards. We hope that people all over the world would perform secure multiparty computation in their daily activities by utilizing our protocols that require only a standard deck of cards. For example, to avoid an awkward situation, a group of friends can determine whether or not they go out for a drink by securely computing the conjunction $x_1 \wedge x_2 \wedge \cdots \wedge x_n$ of their NO/YES input bits $x_1, x_2, \ldots, x_n$. All they need is a deck of playing cards. Furthermore, in the literature, playing cards related to cryptography have been studied (e.g., [3, 5, 17]). These can reveal the underlying concepts of cryptography to non-specialists.

Acknowledgments

We thank the anonymous referees, whose comments have helped us to improve the presentation of the paper. This work was supported by JSPS KAKENHI Grant Number 26330001.

References

1. den Boer, B.: More efficient match-making and satisfiability: the five card trick. In: Quisquater, J.J., Vandewalle, J. (eds.) Advances in Cryptology — EUROCRYPT '89, Lecture Notes in Computer Science, vol. 434, pp. 208–217. Springer Berlin Heidelberg (1990)

2. Crépeau, C., Kilian, J.: Discreet solitary games. In: Stinson, D.R. (ed.) Advances in Cryptology — CRYPTO '93, Lecture Notes in Computer Science, vol. 773, pp. 319–330. Springer Berlin Heidelberg (1994)
3. Duan, Z., Yang, C.: Unconditional secure communication: a Russian cards protocol. Journal of Combinatorial Optimization 19(4), 501–530 (2010)
4. Fisch, B., Freund, D., Naor, M.: Physical zero-knowledge proofs of physical properties. In: Garay, J., Gennaro, R. (eds.) Advances in Cryptology CRYPTO 2014, Lecture Notes in Computer Science, vol. 8617, pp. 313–336. Springer Berlin Heidelberg (2014)
5. Fischer, M.J., Wright, R.N.: Bounds on secret key exchange using a random deal of cards. Journal of Cryptology 9(2), 71–99 (1996)
6. Glaser, A., Barak, B., Goldston, R.J.: A zero-knowledge protocol for nuclear warhead verification. Nature 510(7506), 497–502 (2014)
7. Koch, A., Walzer, S., Hrtel, K.: Card-based cryptographic protocols using a minimal number of cards. In: Iwata, T., Cheon, J. (eds.) Advances in Cryptology ASIACRYPT 2015, Lecture Notes in Computer Science, vol. 9452, pp. 783–807. Springer Berlin Heidelberg (2015)
8. Mizuki, T., Kumamoto, M., Sone, H.: The five-card trick can be done with four cards. In: Wang, X., Sako, K. (eds.) Advances in Cryptology — ASIACRYPT 2012, Lecture Notes in Computer Science, vol. 7658, pp. 598–606. Springer Berlin Heidelberg (2012)
9. Mizuki, T., Shizuya, H.: A formalization of card-based cryptographic protocols via abstract machine. International Journal of Information Security 13(1), 15–23 (2014)
10. Mizuki, T., Shizuya, H.: Practical card-based cryptography. In: Ferro, A., Luccio, F., Widmayer, P. (eds.) Fun with Algorithms, Lecture Notes in Computer Science, vol. 8496, pp. 313–324. Springer International Publishing (2014)
11. 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, Lecture Notes in Computer Science, vol. 5598, pp. 358–369. Springer Berlin Heidelberg (2009)
12. Niemi, V., Renvall, A.: Secure multiparty computations without computers. Theoretical Computer Science 191(1–2), 173–183 (1998)
13. Niemi, V., Renvall, A.: Solitaire zero-knowledge. Fundam. Inf. 38(1,2), 181–188 (1999)
14. Nishida, T., Hayashi, Y., Mizuki, T., Sone, H.: Card-based protocols for any Boolean function. In: Jain, R., Jain, S., Stephan, F. (eds.) Theory and Applications of Models of Computation, Lecture Notes in Computer Science, vol. 9076, pp. 110–121. Springer International Publishing (2015)
15. Nishida, T., Hayashi, Y., Mizuki, T., Sone, H.: Securely computing three-input functions with eight cards. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E98.A(6), 1145–1152 (2015)
16. Stiglic, A.: Computations with a deck of cards. Theoretical Computer Science 259(1–2), 671–678 (2001)
17. Swanson, C.M., Stinson, D.R.: Combinatorial solutions providing improved security for the generalized Russian cards problem. Designs, Codes and Cryptography 72(2), 345–367 (2014)
18. Ueda, I., Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: How to implement a random bisection cut. In: Theory and Practice of Natural Computing. Lecture Notes in Computer Science, Springer Berlin Heidelberg (2016, to appear)