Card-Based Protocols Using Unequal Division Shuffles*

Akihiro Nishimura Takuya Nishida Yu-ichi Hayashi Takaaki Mizuki
Hideaki Sone

Tohoku University
`tm-paper+card5cop[atmark]g-mail.tohoku-university.jp`

Abstract

Card-based cryptographic protocols can perform secure computation of Boolean functions. In 2013, Cheung et al. presented a protocol that securely produces a hidden AND value using five cards; however, it fails with a probability of 1/2. The protocol uses an unconventional shuffle operation called an unequal division shuffle; after a sequence of five cards is divided into a two-card portion and a three-card portion, these two portions are randomly switched so that nobody knows which is which. In this paper, we first show that the protocol proposed by Cheung et al. securely produces not only a hidden AND value but also a hidden OR value (with a probability of 1/2). We then modify their protocol such that, even when it fails, we can still evaluate the AND value in the clear. Furthermore, we present two five-card copy protocols (which can duplicate a hidden value) using unequal division shuffle. Because the most efficient copy protocol currently known requires six cards, our new protocols improve upon the existing results. We also design a general copy protocol that produces multiple copies using an unequal division shuffle. Furthermore, we show feasible implementations of unequal division shuffles by the use of card cases.

1 Introduction

Suppose that Alice and Bob have Boolean values $a \in \{0, 1\}$ and $b \in \{0, 1\}$, respectively, each of which describes his/her private opinion (or something similar), and they want to conduct secure AND computation by themselves, i.e., they wish to know only the value of $a \wedge b$. In such a situation, a card-based cryptographic protocol is a convenient solution. Many such protocols for this purpose have already been proposed (Boe90; CK94; NR98; Sti01; MS09; MKS12; CHL13; KWH15), one of which can be selected by them for secure AND computation. For example, if they select the six-card AND protocol (MS09), they can securely produce a hidden value of $a \wedge b$ using six playing cards, e.g., $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$, along with a "random bisection cut," which will be explained later.

*This is an Author's accepted manuscript for publication in Soft Computing. The final publication is available at Springer via `http://dx.doi.org/10.1007/s00500-017-2858-2`. An earlier version of this study was presented at 4th International Conference on the Theory and Practice of Natural Computing, TPNC 2015, Spain, December 15–16, 2015, and appeared in Proc. TPNC 2015, Lecture Notes in Computer Science, Springer International Publishing, vol. 9477, pp. 109–120, 2015 (NNH$^+$15).

In 2013, Cheung et al. presented a protocol that securely produces a hidden AND value using only five cards ($\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}$); however, it fails (and has to restart) with a probability of 1/2 (CHL13) (we refer to it as the *CHL AND protocol* in this paper). The protocol uses an unconventional shuffling operation that we refer to as an "unequal division shuffle"; after a sequence of five cards is divided into a two-card portion and a three-card portion, these two portions are randomly switched so that nobody knows which is which. The objective of this paper is to improve the CHL AND protocol and propose other efficient protocols using unequal division shuffles.

This paper begins by presenting some definitions related to card-based protocols.

1.1 Preliminary Definitions

Throughout this paper, we assume that cards satisfy the following properties.

1. All cards of the same type (black $\boxed{\clubsuit}$ or red $\boxed{\heartsuit}$) are indistinguishable from one another.
2. Each card has the same pattern $\boxed{?}$ on its back side, and hence, all face-down cards are indistinguishable from one another.

We define the following encoding scheme to deal with a Boolean value:

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

Given a bit $x \in \{0,1\}$, when a pair of face-down cards $\boxed{?}\boxed{?}$ describes the value of x with encoding scheme (1), it is called a *commitment* to x and is expressed as

$$\underbrace{\boxed{?}\boxed{?}}_{x}\,. \tag{2}$$

For a commitment to $x \in \{0,1\}$, we sometimes write

$$\underbrace{\boxed{?}}_{x^0}\ \underbrace{\boxed{?}}_{x^1}$$

instead of expression (2), where $x^0 := x$ and $x^1 := \overline{x}$. In other words, we sometimes use a one-card encoding scheme, $\boxed{\clubsuit} = 0$, $\boxed{\heartsuit} = 1$, for convenience.

Given commitments to players' private inputs, a card-based protocol is supposed to produce a sequence of cards as its output. *Committed-format* protocols produce their output as a commitment. For example, any committed-format AND protocol outputs

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

from input commitments to a and b. It should be noted that such an output commitment can be used as an input for another computation. On the other hand, *non-committed-format* protocols produce their output in another form.

Hereafter, for a sequence consisting of $d \in \mathbb{N}$ cards, each card of the sequence is sequentially numbered from the left (position 1, position 2, ..., position d), e.g.,

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{\clubsuit}}\overset{3}{\boxed{\heartsuit}}\cdots\overset{d}{\boxed{?}}\,.$$

1.2 Our Results

As mentioned above, given commitments to Alice's bit a and Bob's bit b together with an additional card $\boxed{\clubsuit}$, the CHL AND protocol produces a commitment to $a \wedge b$ with a probability of 1/2; when it fails, the players have to create their input commitments again. This paper shows that in the last step of the CHL AND protocol, a commitment to the OR value $a \vee b$ is also obtained when the protocol succeeds in producing a commitment to $a \wedge b$. Next, we show that, even when the protocol fails, we can still evaluate the AND value (more precisely, any Boolean function) in the clear by slightly modifying the last step of the protocol. Thus, the improved protocol, which can be called a "hybrid protocol," never fails to compute the AND value.

Furthermore, we present two five-card copy protocols using unequal division shuffles. Because the most efficient copy protocol currently known requires six cards (MS09), our new protocols improve upon the existing results in terms of the number of required cards, as shown in Table 1. Note that our protocols require an average of two trials[1], and the protocol (i) in Table 1 uses a random cut, which is a cyclic shuffling operation as sometimes used in usual card games. We also design a general copy protocol that produces n copied commitments using an unequal division shuffle for an arbitrary $n \geq 3$. In addition, we show feasible implementations of unequal division shuffles by the use of card cases.

Table 1: Protocols for making two copied commitments

	# of cards	Type of shuffle	Avg. # of trials
(i)	8	RC	1
(ii)	6	RBC	1
Ours (§ 4)	5	UDS	2

RC: Random Cut, RBC: Random Bisection Cut,
UDS: Unequal Division Shuffle
(i): (CK94)
(ii): (MS09)

The remainder of this paper is organized as follows. Section 2 first introduces the CHL AND protocol along with known shuffle operations and then presents a more general definition of unequal division shuffle. Section 3 describes our slight modification to the last step of the CHL AND protocol to expand its functionality. Section 4 proposes two new copy protocols that outperform the previous protocols in terms of the number of required cards. Section 5 presents a general copy protocol. Section 6 demonstrates how to practically implement unequal division shuffle with physical card cases. Finally, Section 7 summarizes our findings and concludes the paper.

An earlier version of this study was presented and appeared as an LNCS (Lecture Notes in Computer Science) paper (NNH^{+}15). The present paper is substantially extended as compared to the LNCS paper: this paper extends the previous results to designing a

[1] As seen in Section 4, we repeat applying a shuffle until $\boxed{\clubsuit}$ is found, and the probability that $\boxed{\clubsuit}$ appears is 1/2.

general copy protocol that produces n copied commitments, and also demonstrates how to practically implement unequal division shuffle in details. Sections 5 and 6 are devoted to these new results.

2 Card Shuffling Operations and the CHL AND Protocol

In this section, we first introduce a random bisection cut (MS09). Then, we give a general definition of unequal division shuffle. Finally, we introduce the CHL AND protocol (CHL13).

2.1 Random Bisection Cut

Suppose that there is a sequence of $2m$ face-down cards for some $m \in \mathbb{N}$:

$$\underbrace{\overbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}^{\circ}}_{m\ \text{cards}}\,\underbrace{\overbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}^{\bullet}}_{m\ \text{cards}}.$$

Then, a *random bisection cut* (MS09) on these cards (denoted by $[\cdot|\cdot]$)

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

means that we bisect the sequence and randomly switch the two portions (of size m). Thus, the result of the operation will be either

$$\overbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}^{\circ}\,\overbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}^{\bullet}$$

or

$$\overbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}^{\bullet}\,\overbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}^{\circ},$$

where each occurs with a probability of exactly 1/2, and nobody knows which is the current sequence.

The introduction of the random bisection cut led to a significant reduction of the number of cards in AND and XOR protocols (MKS12; MS09). Using random bisection cuts, we can also construct a six-card copy protocol (MS09) (as seen in Table 1), adder protocols (MAS13), protocols for any three-variable symmetric functions (NMS13), and so on.

Whereas the committed-format AND protocol (MS09) using a random bisection cut requires six cards as stated above, Cheung et al. introduced an unequal division shuffle whereby they constructed a five-card committed-format AND protocol that works with a probability of 1/2. Its details are presented in the next two subsections. It should be noted that Koch, Walzer, and Härtel (KWH15) reduced the number of cards further using unequal division shuffle and its variant, that is, they proposed a four-card committed-format AND protocol that never fails.

2.2 Unequal Division Shuffle

Here, we present a formal definition of unequal division shuffle, which first appeared in the CHL AND protocol (CHL13).

Suppose that there is a sequence of $\ell \geq 3$ ($\ell \in \mathbb{N}$) face-down cards:

$$\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{\ell\text{ cards}}.$$

Divide it into two portions of unequal sizes, say, j cards and k cards, where $j+k=\ell$, $j \neq k$, as follows:

$$\overbrace{\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{j\text{ cards}}\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{k\text{ cards}}}^{\ell\text{ cards}}.$$

We consider an operation that randomly switches these two portions of unequal sizes; we refer to it as an *unequal division shuffle* or a (j,k)*-division shuffle* (denoted by $[\cdot|\cdot]$) :

$$\Big[\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{j\text{ cards}}\Big|\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{k\text{ cards}}\Big].$$

Thus, the result of the operation will be either

$$\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{j\text{ cards}}\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{k\text{ cards}}$$

or

$$\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{k\text{ cards}}\underbrace{\boxed{?}\boxed{?}\cdots\boxed{?}}_{j\text{ cards}},$$

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

We demonstrate feasible implementations (for humans) of unequal division shuffle in Section 6.

2.3 The CHL AND Protocol

In this subsection, we introduce the CHL AND protocol. It requires an additional card $\boxed{\clubsuit}$ to produce a commitment to $a \wedge b$ from two commitments

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

placed by Alice and Bob, respectively. As mentioned in Section 2.2, the protocol uses unequal division shuffle, specifically a $(2,3)$-division shuffle, as follows.

1. Arrange the cards of the two input commitments and the additional card as

$$\underbrace{\boxed{?}}_{a^0}\;\underbrace{\boxed{?}}_{\clubsuit}\;\underbrace{\boxed{?}}_{a^1}\;\underbrace{\boxed{?}}_{b^0}\;\underbrace{\boxed{?}}_{b^1}.$$

2. Apply a $(2,3)$-division shuffle:

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

3. Reveal the card at position 1.

   (a) If the card is $\boxed{\clubsuit}$, then the cards at positions 2 and 3 constitute a commitment to $a \wedge b$:

   $$\boxed{\clubsuit}\underbrace{\boxed{?}\,\boxed{?}}_{a\wedge b}\boxed{?}\,\boxed{?}.$$

   (b) If the card is $\boxed{\heartsuit}$, then Alice and Bob create input commitments again to restart the protocol.

This is the CHL AND protocol. We confirm its correctness. As above, the input to the CHL AND protocol consists of commitments to $a, b \in \{0, 1\}$ along with an additional card $\boxed{\clubsuit}$. There are two possibilities due to the outcome of $(2,3)$-division shuffle:

$$\underbrace{\boxed{?}}_{a^0}\;\underbrace{\boxed{?}}_{\clubsuit}\;\underbrace{\boxed{?}}_{a^1}\;\underbrace{\boxed{?}}_{b^0}\;\underbrace{\boxed{?}}_{b^1}\quad\text{and}\quad\underbrace{\boxed{?}}_{a^1}\;\underbrace{\boxed{?}}_{b^0}\;\underbrace{\boxed{?}}_{b^1}\;\underbrace{\boxed{?}}_{a^0}\;\underbrace{\boxed{?}}_{\clubsuit},$$

where each case occurs with a probability of 1/2. We enumerate all possibilities of input and card sequences after step 2 of the protocol in Table 2 (recall encoding scheme (1)). Looking at the cards at positions 2 and 3 when the card at position 1 is $\boxed{\clubsuit}$ in Table 2, we can easily confirm the correctness of the protocol, i.e., the cards at positions 2 and 3 surely constitute a commitment to $a \wedge b$.

Table 2: All possibilities of input and card sequences after step 2

Input	Card sequences	
(a,b)	a^0 ♣ a^1 b^0 b^1	a^1 b^0 b^1 a^0 ♣
$(0,0)$	♣ ♣ ♡ ♣ ♡	♡ ♣ ♡ ♣ ♣
$(0,1)$	♣ ♣ ♡ ♡ ♣	♡ ♡ ♣ ♣ ♣
$(1,0)$	♡ ♣ ♣ ♣ ♡	♣ ♣ ♡ ♡ ♣
$(1,1)$	♡ ♣ ♣ ♡ ♣	♣ ♡ ♣ ♡ ♣

3 Improved CHL AND Protocol

In this section, we analyze the CHL AND protocol and change its last step to develop an improved protocol.

3.1 Bonus Commitment to OR

When we succeed in obtaining a commitment to $a \wedge b$, i.e., when the card at position 1 is $\boxed{\clubsuit}$ in the last step of the CHL AND protocol, we are also able to simultaneously obtain a commitment to the OR value $a \vee b$. Thus, as indicated in Table 2, if the card at position 1 is $\boxed{\clubsuit}$, then the cards at positions 4 and 5 constitute a commitment to $a \vee b$.

3.2 In Case of Failure

Suppose that the card at position 1 is $\boxed{\heartsuit}$ in the last step of the CHL AND protocol. This means that the AND computation failed and we have to start from scratch, i.e., Alice and Bob need to create their private input commitments again. However, we show that they need not do so: they can evaluate the AND value even when the CHL AND protocol fails, as follows.

From Table 2, if the card at position 1 is $\boxed{\heartsuit}$, the sequence of five cards

$$\boxed{\heartsuit}\boxed{?}\boxed{?}\boxed{?}\boxed{?} \tag{3}$$

is one of the four possibilities shown in Table 3, depending on the value of (a, b).

Table 3: Possible sequences when the CHL AND protocol fails

Input (a, b)	Sequence of five cards
$(0, 0)$	$\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}$
$(0, 1)$	$\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}$
$(1, 0)$	$\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}$
$(1, 1)$	$\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}$

Therefore, the card at position 4 indicates the value of $a \wedge b$, i.e., if the card at position 4 is $\boxed{\clubsuit}$, then $a \wedge b = 0$, and if the card is $\boxed{\heartsuit}$, then $a \wedge b = 1$. Note that opening the card does not reveal any information about the inputs a and b besides the value of $a \wedge b$. Thus, this protocol does not fail to compute the AND value.

Actually, we can compute any Boolean function $f(a, b)$ in a non-committed format, given the sequence (3) above, as follows. Note that, as seen in Table 3, the position of the face-down card $\boxed{\heartsuit}$ (which is between 2 and 5) uniquely determines the value of the input (a, b). We scramble all cards at positions corresponding to $f(a, b) = 1$ (possibly one card as in the case of $f(a, b) = a \wedge b$) and reveal all these cards. If $\boxed{\heartsuit}$ appears anywhere, then $f(a, b) = 1$; otherwise, $f(a, b) = 0$. Thus, we can evaluate the desired function (in a non-committed format).

3.3 Improved Protocol

From the discussion above, we have the following improved protocol.

1. Arrange the five cards as follows:

$$\underbrace{\boxed{?}}_{a^0}\;\underbrace{\boxed{?}}_{\clubsuit}\;\underbrace{\boxed{?}}_{a^1}\;\underbrace{\boxed{?}}_{b^0}\;\underbrace{\boxed{?}}_{b^1}.$$

2. Apply $(2, 3)$-division shuffle:

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

3. Reveal the card at position 1.

(a) If the card is $\boxed{\clubsuit}$, then the cards at positions 2 and 3 constitute a commitment to $a \wedge b$; moreover, the cards at positions 4 and 5 constitute a commitment to $a \vee b$:

$$\boxed{\clubsuit}\underbrace{\boxed{?}\boxed{?}}_{a\wedge b}\underbrace{\boxed{?}\boxed{?}}_{a\vee b}.$$

(b) If the card is $\boxed{\heartsuit}$, then we can evaluate any desired Boolean function $f(a,b)$. Scramble all cards at positions corresponding to $f(a,b)=1$ and reveal them. If $\boxed{\heartsuit}$ appears, then $f(a,b)=1$; otherwise, $f(a,b)=0$.

Because this protocol is neither somewhat committed-format nor non-committed format, we may call it a *hybrid protocol.* From this hybrid protocol, we can immediately derive two five-card protocols; the first one is a two-bit output (AND and OR) protocol in committed format, and the second one is a non-committed-format protocol for any Boolean function. Both the protocols fail with a probability of 1/2 and need to restart. The recent paper (FAN$^+$17) showed that six cards are necessary for producing commitments to the AND and OR values (without restarting), and hence the 5-card AND-and-OR protocol implies that there is a possibility to reduce the number of cards if we accept a failure causing a restart. On the other hand, the second protocol, namely, the 5-card non-committed-format protocol is not so interesting because we can have a 4-card non-committed-format protocol for any symmetric Boolean function by combining the way in Step 3(b) above with the idea behind the four-card non-committed-format AND protocol given in (MKS12).

4 Five-Card Copy Protocols

In this section (and the next section), we focus on protocols for copying a commitment.

From Table 1, using the six-card copy protocol (MS09), a commitment to bit $a \in \{0,1\}$ can be copied with four additional cards:

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

This is the most efficient copy protocol (in terms of the number of cards) known prior to this study. In contrast, we prove that three additional cards (two $\boxed{\clubsuit}$s and one $\boxed{\heartsuit}$) are sufficient by proposing a five-card copy protocol using unequal division shuffle. We also propose another copy protocol that has fewer steps by considering a different shuffle in Section 4.2.

4.1 Copy Protocol Using Unequal Division Shuffle

Given a commitment

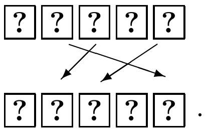
$$\underbrace{\boxed{?}\boxed{?}}_{a}$$

together with additional cards $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}$, our protocol makes two copied commitments, as follows.

1. Arrange the five cards as
$$\underbrace{\boxed{?}}_{\clubsuit}\,\underbrace{\boxed{?}}_{a^0}\,\underbrace{\boxed{?}}_{\heartsuit}\,\underbrace{\boxed{?}}_{a^1}\,\underbrace{\boxed{?}}_{\clubsuit}\,.$$

2. Apply a $(2,3)$-division shuffle:
$$\left[\,\boxed{?}\boxed{?}\,\middle|\,\boxed{?}\boxed{?}\boxed{?}\,\right].$$

3. Rearrange the sequence of five cards as



4. Reveal the card at position 5.
    (a) If the card is $\boxed{\clubsuit}$, then we have two commitments to a as follows:
    $$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{a}\boxed{\clubsuit}.$$
    (b) If the card is $\boxed{\heartsuit}$, then we have
    $$\underbrace{\boxed{?}\boxed{?}}_{\overline{a}}\boxed{?}\boxed{?}\boxed{\heartsuit}.$$
    Swap the cards at positions 1 and 2 to obtain a commitment to a. After revealing the cards at positions 3 and 4 (which must be $\boxed{\clubsuit}\boxed{\clubsuit}$), return to step 1.

After step 3, there are two possibilities due to the shuffle outcome: the sequence of five cards is either $\clubsuit\,\heartsuit\,\clubsuit\,a^1\,a^0$ or $\heartsuit\,\clubsuit\,a^0\,\clubsuit\,a^1$. Table 4 enumerates all possibilities of input and card sequences after step 3 of the protocol. As can be easily seen in the table, we surely have two copied commitments in step 4(a). Note that opening the card at position 5 does not reveal any information about the input a. Thus, we have designed a five-card copy protocol that improves upon the previous results in terms of the number of required cards. It should be noted that the protocol is a Las Vegas algorithm with an average of two trials.

Table 4: Possible sequences after step 3 of our first copy protocol

Input	Card sequences	
a	$\clubsuit\ \heartsuit\ \clubsuit\ a^1\ a^0$	$\heartsuit\ \clubsuit\ a^0\ \clubsuit\ a^1$
0	$\clubsuit\ \heartsuit\ \clubsuit\ \heartsuit\ \clubsuit$	$\heartsuit\ \clubsuit\ \clubsuit\ \clubsuit\ \heartsuit$
1	$\clubsuit\ \heartsuit\ \clubsuit\ \clubsuit\ \heartsuit$	$\heartsuit\ \clubsuit\ \heartsuit\ \clubsuit\ \clubsuit$

4.2 Copy Protocol Using Double Unequal Division Shuffle

In this subsection, we reduce the number of steps for achieving copy computation by modifying the unequal division shuffle approach.

Remember that (2,3)-division shuffle changes the order of the two portions:

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}} \vdots \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{5}{\boxed{?}} \rightarrow \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{5}{\boxed{?}} \vdots \overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\ .$$

Here, we consider a further division of the three-card portion:

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

Thus, given a sequence of five cards

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{5}{\boxed{?}},$$

a shuffle operation resulting in either

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{5}{\boxed{?}} \quad \text{or} \quad \overset{5}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}$$

is called a *double unequal division shuffle.*

Using such a shuffle, we can avoid rearranging the cards in step 3 of the protocol presented in Section 4.1, as follows.

1. Arrange the five cards as
$$\underbrace{\boxed{?}}_{a^0}\ \underbrace{\boxed{?}}_{\clubsuit}\ \underbrace{\boxed{?}}_{\heartsuit}\ \underbrace{\boxed{?}}_{\clubsuit}\ \underbrace{\boxed{?}}_{a^1}\ .$$

2. Apply a double unequal division shuffle:
$$\left[\boxed{?}\boxed{?}\,\middle|\,\boxed{?}\boxed{?}\vdots\boxed{?}\right].$$
Remember that this results in one of the two possible sequences.

3. Reveal the card at position 1.

   (a) If the card is $\boxed{\clubsuit}$, then we have two commitments to a:
   $$\boxed{\clubsuit}\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{a}.$$

   (b) If the card is $\boxed{\heartsuit}$, then we have
   $$\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{\overline{a}}\boxed{?}\boxed{?}.$$
   Swap the cards at positions 2 and 3 to obtain a commitment to a. After revealing the cards at positions 4 and 5, return to step 1.

This protocol has two possibilities after step 2: the sequence of five cards is either $a^0\clubsuit\heartsuit\clubsuit a^1$ or $a^1\heartsuit\clubsuit a^0\clubsuit$. Table 5 confirms the correctness of the protocol.

In the next section, we will extend this protocol to a general protocol that can produce three or more copied commitments.

Table 5: Possible sequences after step 2 of our second copy protocol

Input	Card sequences	
a	a^0 ♣ ♡ ♣ a^1	a^1 ♡ ♣ a^0 ♣
0	♣ ♣ ♡ ♣ ♡	♡ ♡ ♣ ♣ ♣
1	♡ ♣ ♡ ♣ ♣	♣ ♡ ♣ ♡ ♣

5 General Copy Protocol

In this section, we propose a general copy protocol, that produces n identical copied commitments from a given commitment to $a \in \{0,1\}$ using $2n+1$ cards, where $n \geq 2$.

As a comparison to the previous results is shown in Table 6, this protocol reduces the number of cards required to obtain n copied commitments by one, whereas the average number of trials doubles. Note that, n commitments to a can be obtained using $2n+1$ cards by repeating the five-card copy protocols presented in Section 4 $n-1$ times; however, the following protocol requires fewer steps and trials.

Our protocol is a generalization of the five-card copy protocol constructed in Section 4.2. Thus, we employ a double unequal division shuffle. Specifically, given a sequence of $2n+1$ cards

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}\;\overset{2n+1}{\boxed{?}}\;,$$

we use the following double unequal division shuffle:

$$\left[\,\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\;\middle|\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}\;\vdots\;\overset{2n+1}{\boxed{?}}\,\right].$$

Therefore, the result of the operation must be either

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}\;\overset{2n+1}{\boxed{?}}\quad\text{or}\quad\overset{2n+1}{\boxed{?}}\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}\overset{1}{\boxed{?}}\overset{2}{\boxed{?}},$$

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

The following is the procedure of our general copy protocol.

Table 6: Copy protocols for making n commitments

	# of cards	Type of shuffle	Avg. # of trials
(i)	$2n+4$	RC	1
(ii)	$2n+2$	RBC	1
Ours (§ 5)	$2n+1$	DUDS	2

RC: Random Cut, RBC: Random Bisection Cut,
DUDS: Double Unequal Division Shuffle
(i): (CK94)
(ii): (MS09)

1. Arrange a given commitment to a and $2n-1$ additional cards as

$$\underbrace{\boxed{?}}_{a^0}\,\overbrace{\underbrace{\boxed{?}}_{\clubsuit}\,\underbrace{\boxed{?}}_{\heartsuit}\,\underbrace{\boxed{?}}_{\clubsuit}\cdots\underbrace{\boxed{?}}_{\heartsuit}\,\underbrace{\boxed{?}}_{\clubsuit}}^{2n-1\text{ cards}}\,\underbrace{\boxed{?}}_{a^1}.$$

2. Apply the following double unequal division shuffle:

$$\left[\underbrace{\boxed{?}}_{a^0}\,\underbrace{\boxed{?}}_{\clubsuit}\,\middle|\,\underbrace{\boxed{?}}_{\heartsuit}\,\underbrace{\boxed{?}}_{\clubsuit}\cdots\underbrace{\boxed{?}}_{\heartsuit}\,\underbrace{\boxed{?}}_{\clubsuit}\,\vdots\,\underbrace{\boxed{?}}_{a^1}\right].$$

3. Reveal the card at position 1.
   (a) If the card is $\boxed{\clubsuit}$, then we have n commitments to a as follows:

   $$\boxed{\clubsuit}\overbrace{\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{a}\cdots\underbrace{\boxed{?}\boxed{?}}_{a}}^{2n\text{ cards}}.$$

   (b) If the card is $\boxed{\heartsuit}$, then we have $n-1$ commitments to negation of a as follows:

   $$\boxed{\heartsuit}\overbrace{\underbrace{\boxed{?}\boxed{?}}_{\overline{a}}\cdots\underbrace{\boxed{?}\boxed{?}}_{\overline{a}}}^{2n-2\text{ cards}}\boxed{?}\boxed{?}.$$

   To obtain one more commitment to a, after revealing the last two cards (which must be $\boxed{\clubsuit}\boxed{\clubsuit}$), execute the five-card copy protocol shown in Section 4.2.

Table 7 shows all possibilities before revealing the card at position 1. Note that this protocol takes an average number of two trials.

Table 7: Possible sequences after step 2 of our general copy protocol

	Card sequences	
a	$a^0\ \clubsuit\ \heartsuit\ \cdots\clubsuit\ \heartsuit\ \clubsuit\ a^1$	$a^1\ \heartsuit\ \clubsuit\ \cdots\heartsuit\ \clubsuit\ a^0\ \clubsuit$
0	$\clubsuit\ \clubsuit\ \heartsuit\ \cdots\clubsuit\ \heartsuit\ \clubsuit\ \heartsuit$	$\heartsuit\ \heartsuit\ \clubsuit\ \cdots\heartsuit\ \clubsuit\ \clubsuit\ \clubsuit$
1	$\heartsuit\ \clubsuit\ \heartsuit\ \cdots\clubsuit\ \heartsuit\ \clubsuit\ \clubsuit$	$\clubsuit\ \heartsuit\ \clubsuit\ \cdots\heartsuit\ \clubsuit\ \heartsuit\ \clubsuit$

6 Implementation of Unequal Division Shuffle and Double Unequal Division Shuffle

This section discusses how to implement unequal division shuffle and double unequal division shuffle with everyday objects.

Note that a random bisection cut (introduced in Section 2.1) can be easily implemented by humans (see (UNH$^+$16) for details); after bisecting a given card sequence, Alice and Bob take turns to randomly switch the two portions until they are satisfied. On the other hand, if Alice and Bob try to implement unequal division shuffle in the same way, then they will realize the current order of the two portions because of the different sizes of the portions. To avoid such information leakage, we propose to utilize physical cases that satisfy some properties.

Specifically, we consider the card cases shown in Figure 1. Each case can store a portion of cards and has two sliding covers, an upper cover and a lower cover. We assume that the weight of a deck of cards is negligible compared to the case. We think, for instance, that boxes (Figure 1) or envelopes (Figure 2) can be used as such cases.

In the sequel, we implement every unequal division shuffle appearing so far in this paper using card cases; the use of different tools will be illustrated. It should be noted that these card cases can be used for implementing "non-uniform" shuffles; see (NHMSe16) for the details.

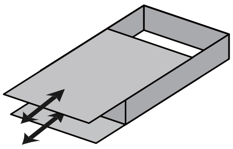


Figure 1: A box suitable for a card case

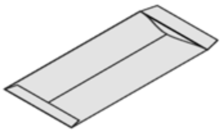


Figure 2: An envelope suitable for a card case

6.1 How to Implement the (2,3)-division Shuffle

Here, we propose an implementation of the (2,3)-division shuffle using two cases.

Remember that, after applying the (2,3)-division shuffle

$$\left[\overset{1}{\boxed{?}}\,\overset{2}{\boxed{?}}\;\middle|\;\overset{3}{\boxed{?}}\,\overset{4}{\boxed{?}}\,\overset{5}{\boxed{?}}\right],$$

we must have either

$$\overset{1}{\boxed{?}}\,\overset{2}{\boxed{?}}\,\overset{3}{\boxed{?}}\,\overset{4}{\boxed{?}}\,\overset{5}{\boxed{?}} \text{ or } \overset{3}{\boxed{?}}\,\overset{4}{\boxed{?}}\,\overset{5}{\boxed{?}}\,\overset{1}{\boxed{?}}\,\overset{2}{\boxed{?}},$$

where each occurs with a probability of 1/2.

The following steps perform the (2,3)-division shuffle used in the CHL AND protocol (Section 2.3), its improved protocol (Section 3.3), and the five-card copy protocol (Section 4.1).

1. Divide a given five-card sequence into a two-card portion and a three-card portion; then, store the first portion in the first case C_1, and the second portion in the second case C_2 (Figure 3):

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\mapsto C_1 \Big| \ \overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{5}{\boxed{?}}\mapsto C_2.$$

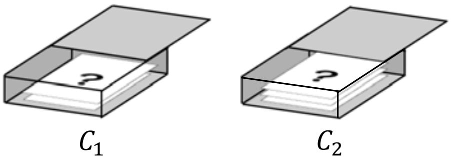


Figure 3: Storing the two portions

2. Switch C_1 and C_2 randomly (Figure 4)[2]. This operation results in two possible outcomes:

$$C_1\,C_2 \quad \text{or} \quad C_2\,C_1,$$

where each occurs with a probability of 1/2.

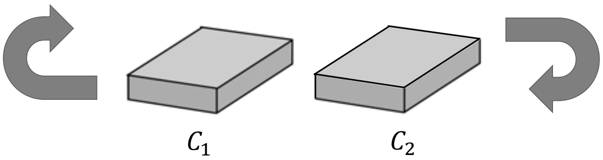


Figure 4: Switching C_1 and C_2 randomly

3. Stack up these cases, as illustrated in Figure 5.

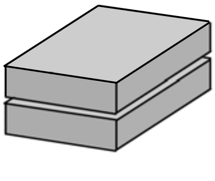


Figure 5: Stacking up the cases

4. Remove the two middle sliding covers simultaneously, as illustrated in Figure 6. Then, we have a sequence of five cards.

As a result of this operation, we have either

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

[2] If players have difficulty to shuffle the two boxes publicly, they may shuffle the two boxes behind their backs (UNH$^+$16).

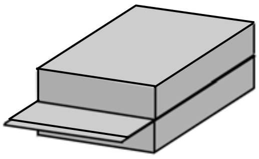


Figure 6: Removing the two covers

(in the case of $C_1\,C_2$), or

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

(in the case of $C_2\,C_1$).

Therefore, the (2,3)-division shuffle can be implemented by humans with card cases.

6.2 Implementation of Double Unequal Division Shuffle Used in Five-Card Copy Protocol

In Section 4.2, a five-card copy protocol using double unequal division shuffle was proposed. We show that it is possible to perform the double unequal division shuffle using three cases.

Remember that the used double unequal division shuffle

$$\left[\,\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\;\middle|\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\,\vdots\,\overset{5}{\boxed{?}}\,\right]$$

results in either

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{5}{\boxed{?}}\ \text{ or }\ \overset{5}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\,.$$

The following steps perform the desired shuffle.

1. Divide a given five-card sequence into two two-card portions and a one-card portion; then, store the first portion in the first case C_1, the second portion in the second case C_2, and the third portion in the third case C_3 (Figure 7):

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\rightarrow C_1\;\Big|\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\rightarrow C_2\;\Big|\;\overset{5}{\boxed{?}}\rightarrow C_3.$$

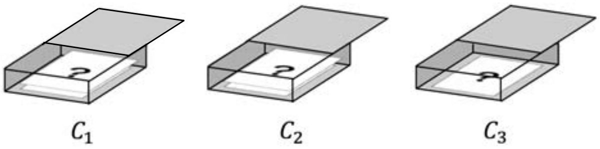


Figure 7: Storing the three portions

2. Switch C_1 and C_3 randomly (Figure 8). This operation results in two possible outcomes:

$$C_1\,C_2\,C_3\ \text{ or }\ C_3\,C_2\,C_1,$$

where each occurs with a probability of 1/2.

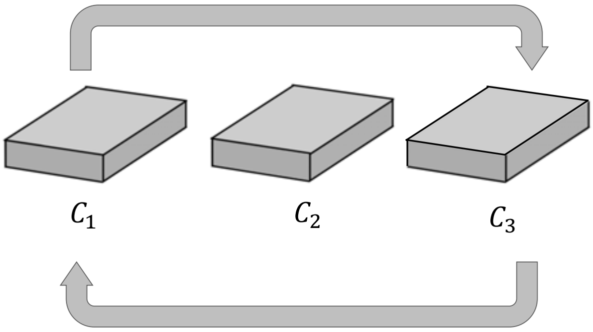


Figure 8: Switching C_1 and C_3 randomly

3. Stack up these cases (without changing the order), as illustrated in Figure 9.

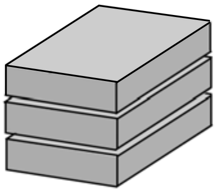


Figure 9: Stacking up the cases

4. Remove all sliding covers except for the top-most and bottom-most covers simultaneously, as illustrated in Figure 10. Then, we have a five-card sequence.

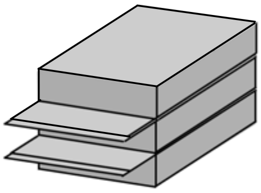


Figure 10: Removing the four covers

As a result of this operation, we have either

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} \end{array}$$

(in the case of C_1 C_2 C_3), or

$$\begin{array}{ccccc} 5 & 3 & 4 & 1 & 2 \\ \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} \end{array}$$

(in the case of C_3 C_2 C_1).

Therefore, the double unequal division shuffle can also be implemented.

6.3 Implementation of Double Unequal Division Shuffle Used in General Copy Protocol

We proposed a general copy protocol using $2n+1$ cards in Section 5. It is also possible to implement the double unequal division shuffle used in the protocol with three cases.

Remember that the used double unequal division shuffle

$$\left[\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\;\middle|\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}\;\vdots\;\overset{2n+1}{\boxed{?}}\right]$$

results in either

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}\;\overset{2n+1}{\boxed{?}}\quad\text{or}\quad\overset{2n+1}{\boxed{?}}\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}.$$

Implementing this shuffle is achieved by almost same operation as the previous subsection. The difference is only the portion to be stored in C_2. We just substitute $\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}$ for $\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}$. Storing the first two cards in C_1 and the last card in C_3 is the same.

Thus, the following steps should be performed. We now use envelopes instead of boxes to illustrate the cases.

1. Divide a given sequence into three portions, and store them in cases C_1, C_2, and C_3, as illustrated in Figure 11:

$$\overset{1}{\boxed{?}}\overset{2}{\boxed{?}}\;\Big|\;\overset{3}{\boxed{?}}\overset{4}{\boxed{?}}\cdots\overset{2n}{\boxed{?}}\;\vdots\;\overset{2n+1}{\boxed{?}}\;.$$

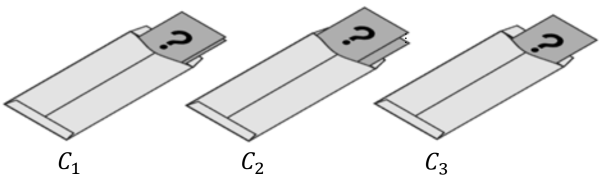


Figure 11: Putting the three portions

2. Switch C_1 and C_3 randomly (Figure 12). This operation results in two possible outcomes:

$$C_1\;C_2\;C_3\quad\text{or}\quad C_3\;C_2\;C_1,$$

where each occurs with a probability of 1/2.

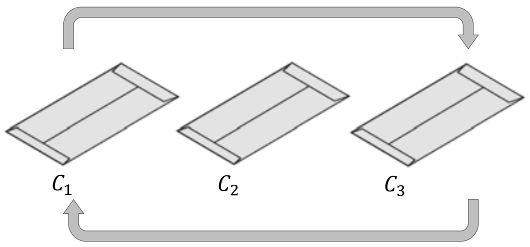


Figure 12: Switching C_1 and C_3 randomly

3. Heap up the three cases (without changing the order), as illustrated in Figure 13.

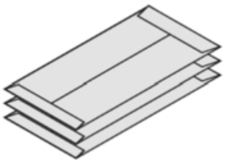


Figure 13: Heaping up the three cases

4. Take all cards out of the envelopes, so as not to change the order of cards and leak any information. We may put the three envelopes in a larger envelop, and remove all the cards inside the larger envelop, as illustrated in Figure 14. Then, we have a sequence of $2n+1$ cards

$$\overbrace{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}}^{2n+1\text{cards}}.$$

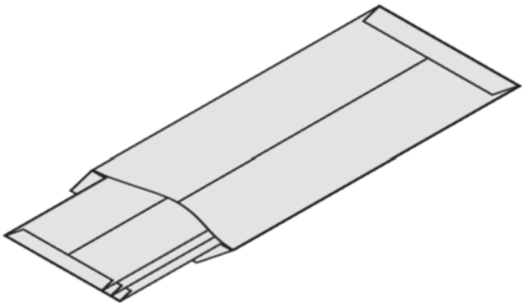


Figure 14: Using a large envelop

One can easily verify the correctness of our implementation.

7 Conclusion

In this paper, we discussed the properties of the AND protocol designed by Cheung et al. and proposed an improved protocol. Although their original protocol produces only a commitment to the AND value with a probability of 1/2, our improved protocol either produces commitments to the AND and OR values or evaluates any Boolean function. Thus, the improved protocol does not fail at all.

Furthermore, we proposed two five-card copy protocols that can securely copy an input commitment using three additional cards. Each of our protocols uses unequal division shuffle. Because the most efficient copy protocol currently known requires six cards, our new protocols improve upon the existing results in terms of the number of required cards.

Extending the results, we also designed a general copy protocol that produces n copied commitments using double unequal division shuffle. In addition, we demonstrated how to practically implement unequal division shuffle in details.

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 Numbers 25289068, 26330001, and 17K00001.

References

[Boe90] Bert den Boer. More efficient match-making and satisfiability: the five card trick. In Jean-Jacques Quisquater and Joos Vandewalle, editors, Advances in Cryptology — EUROCRYPT '89, volume 434 of Lecture Notes in Computer Science, pages 208–217. Springer Berlin Heidelberg, 1990.

[CHL13] Eddie Cheung, Chris Hawthorne, and Patrick Lee. CS 758 project: secure computation with playing cards. https://csclub.uwaterloo.ca/~cdchawth/files% slashpapers/secure_playing_cards.pdf, 2013.

[CK94] Claude Crépeau and Joe Kilian. Discreet solitary games. In Douglas R. Stinson, editor, Advances in Cryptology — CRYPTO '93, volume 773 of Lecture Notes in Computer Science, pages 319–330. Springer Berlin Heidelberg, 1994.

[FAN$^+$17] Danny Francis, Syarifah Ruqayyah Aljunid, Takuya Nishida, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. Necessary and sufficient numbers of cards for securely computing two-bit output functions. In Raphaël C.-W. Phan and Moti Yung, editors, Paradigms in Cryptology – Mycrypt 2016. Malicious and Exploratory Cryptology: Second International Conference, pages 193–211. Springer International Publishing, Cham, 2017.

[KWH15] Alexander Koch, Stefan Walzer, and Kevin Härtel. Card-based cryptographic protocols using a minimal number of cards. In Tetsu Iwata and JungHee Cheon, editors, Advances in Cryptology – ASIACRYPT 2015, volume 9452 of Lecture Notes in Computer Science, pages 783–807. Springer Berlin Heidelberg, 2015.

[MAS13] Takaaki Mizuki, Isaac Kobina Asiedu, and Hideaki Sone. Voting with a logarithmic number of cards. In Giancarlo Mauri, Alberto Dennunzio, Luca Manzoni, and Antonio E. Porreca, editors, Unconventional Computation and Natural Computation, volume 7956 of Lecture Notes in Computer Science, pages 162–173. Springer Berlin Heidelberg, 2013.

[MKS12] Takaaki Mizuki, Michihito Kumamoto, and Hideaki Sone. The five-card trick can be done with four cards. In Xiaoyun Wang and Kazue Sako, editors, Advances in Cryptology — ASIACRYPT 2012, volume 7658 of Lecture Notes in Computer Science, pages 598–606. Springer Berlin Heidelberg, 2012.

[MS09] Takaaki Mizuki and Hideaki Sone. Six-card secure AND and four-card secure XOR. In Xiaotie Deng, John Edward Hopcroft, and Jinyun Xue, editors, Frontiers in Algorithmics, volume 5598 of Lecture Notes in Computer Science, pages 358–369. Springer Berlin Heidelberg, 2009.

[NHMSe16] Akihiro Nishimura, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Son e. An implementation of non-uniform shuffle for secure multi-party comput ation. In Proceedings of the 3rd ACM International Workshop on ASIA Public-K ey Cryptography, AsiaPKC '16, pages 49–55, New York, NY, USA, 2016. ACM.

[NMS13] Takuya Nishida, Takaaki Mizuki, and Hideaki Sone. Securely computing the three-input majority function with eight cards. In Adrian-Horia Dediu, Carlos Martín-Vide, Bianca Truthe, and Miguel A. Vega-Rodríguez, editors, Theory and Practice of Natural Computing, volume 8273 of Lecture Notes in Computer Science, pages 193–204. Springer Berlin Heidelberg, 2013.

[NNH^{+}15] Akihiro Nishimura, Takuya Nishida, Yuichi Hayashi, Takaaki Mizuki, and Hideaki Sone. Five-card secure computations using unequal division shuffle. In Adrian-Horia Dediu, Luis Magdalena, and Carlos Martín-Vide, editors, Theory and Practice of Natural Computing, volume 9477 of Lecture Notes in Computer Science, pages 109–120. Springer International Publishing, 2015.

[NR98] Valtteri Niemi and Ari Renvall. Secure multiparty computations without computers. Theoretical Computer Science, 191(1–2):173–183, 1998.

[Sti01] Anton Stiglic. Computations with a deck of cards. Theoretical Computer Science, 259(1–2):671–678, 2001.

[UNH^{+}16] Itaru Ueda, Akihiro Nishimura, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. How to implement a random bisection cut. In Carlos Martín-Vide, Takaaki Mizuki, and Miguel A. Vega-Rodríguez, editors, Theory and Practice of Natural Computing, pages 58–69. Springer International Publishing, Cham, 2016.