

Card-based protocols using unequal division shuffles

Akihiro Nishimura¹ · Takuya Nishida¹ · Yu-ichi Hayashi² · Takaaki Mizuki³ · Hideaki Sone³

Published online: 4 October 2017
© Springer-Verlag GmbH Germany 2017

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

Communicated by C.M. Vide, A.H. Dediu.

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 (Nishimura et al. 2015).


✉ Takaaki Mizuki
tm-paper+card5cop@g-mail.tohoku-university.jp


- ¹ Sone-Mizuki Laboratory, Graduate School of Information Sciences, Tohoku University, 6-3 Aramaki-Aza-Aoba, Aoba, Sendai 980-8578, Japan
- ² Graduate School of Information Sciences, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 630-0192, Japan
- ³ Cyberscience Center, Tohoku University, 6-3 Aramaki-Aza-Aoba, Aoba, Sendai 980-8578, Japan

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.

Keywords Cryptography · Card-based protocols · Card games · Cryptography without computers · Real-life hands-on cryptography · Secure multi-party computations

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 (Boer 1990; Crépeau and Kilian 1994; Niemi and Renvall 1998; Stiglic 2001; Mizuki and Sone 2009; Mizuki et al. 2012; Cheung et al. 2013; Koch et al. 2015), one of which can be selected by them for secure AND computation. For example, if they select the six-card AND protocol (Mizuki and Sone 2009), they can securely produce a hidden value of $a \wedge b$ using six playing cards, e.g., , along with a “random bisection cut,” which will be explained later.




Cheung et al. (2013) presented a protocol that securely produces a hidden AND value using only five cards (); however, it fails (and has to restart) with a probability of 1/2 (we refer to it as the *CHLAND 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 por-

tion 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  or red ) are indistinguishable from one another.
2. Each card has the same pattern  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:



$$\begin{matrix} \clubsuit & \heartsuit \\ \heartsuit & \clubsuit \end{matrix} = 0, \quad \begin{matrix} \heartsuit & \heartsuit \\ \clubsuit & \clubsuit \end{matrix} = 1. \tag{1}$$

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

$$\underbrace{\begin{matrix} \text{?} & \text{?} \\ \text{?} & \text{?} \end{matrix}}_x. \tag{2}$$

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

$$\underbrace{\begin{matrix} \text{?} \\ \text{?} \end{matrix}}_{x^0} \quad \underbrace{\begin{matrix} \text{?} \\ \text{?} \end{matrix}}_{x^1}$$

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

$$\underbrace{\begin{matrix} \text{?} & \text{?} \\ \text{?} & \text{?} \end{matrix}}_{a \wedge b}$$

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.

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 (Sect. 4)	5	UDS	2

(i) Crépeau and Kilian (1994)


(ii) Mizuki and Sone (2009)

RC Random cut, RBC random bisection cut, UDS unequal division shuffle

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.,



$$\begin{matrix} 1 & 2 & 3 & \dots & d \\ \text{?} & \clubsuit & \heartsuit & \dots & \text{?} \end{matrix}.$$

1.2 Our results

As mentioned above, given commitments to Alice's bit a and Bob's bit b together with an additional card , 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 (Mizuki and Sone 2009), our new protocols improve upon the existing results in terms of the number of required cards, as given 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.

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

¹ As seen in Sect. 4, we repeat applying a shuffle until  is found, and the probability that  appears is $1/2$.

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, Sect. 7 summarizes our findings and concludes the paper.

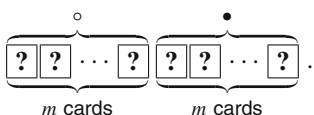
An earlier version of this study was presented and appeared as an lecture notes in computer science (LNCS) paper (Nishimura et al. 2015). The present paper is substantially extended as compared to the LNCS paper: This paper extends the previous results to designing a general copy protocol that produces 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

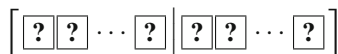
In this section, we first introduce a random bisection cut (Mizuki and Sone 2009). Then, we give a general definition of unequal division shuffle. Finally, we introduce the CHL AND protocol (Cheung et al. 2013).

2.1 Random bisection cut

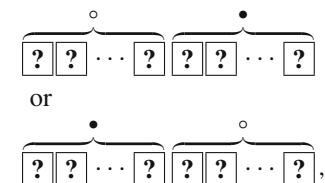
Suppose that there is a sequence of $2m$ face-down cards for some $m \in \mathbb{N}$:



Then, a *random bisection cut* (Mizuki and Sone 2009) on these cards (denoted by $[\cdot|\cdot]$)



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



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

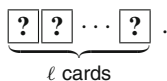
The introduction of the random bisection cut led to a significant reduction of the number of cards in AND and XOR protocols (Mizuki et al. 2012; Mizuki and Sone 2009). Using random bisection cuts, we can also construct a six-card copy protocol (Mizuki and Sone 2009) (as given in Table 1), adder protocols (Mizuki et al. 2013), protocols for any three-variable symmetric functions (Nishida et al. 2013), and so on.

Whereas the committed-format AND protocol (Mizuki and Sone 2009) 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 et al. (2015) 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.

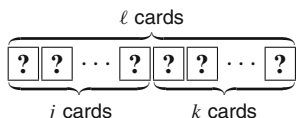
2.2 Unequal division shuffle

Here, we present a formal definition of unequal division shuffle, which first appeared in the CHL AND protocol (Cheung et al. 2013).

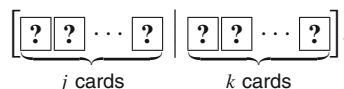
Suppose that there is a sequence of $\ell \geq 3$ ($\ell \in \mathbb{N}$) face-down cards:



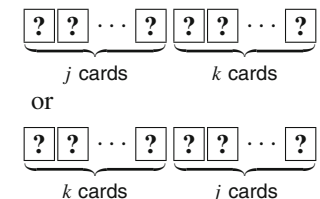
Divide it into two portions of unequal sizes, say, j cards and k cards, where $j + k = \ell$, $j \neq k$, as follows:



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 $[\cdot|\cdot]$):




Thus, the result of the operation will be either

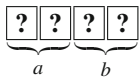


where each case occurs with a probability of exactly 1/2.

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

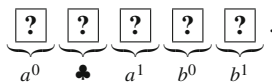
2.3 The CHL AND protocol

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

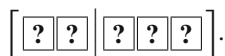



placed by Alice and Bob, respectively. As mentioned in Sect. 2.2, the protocol uses unequal division shuffle, specifically a (2, 3)-division shuffle, as follows.

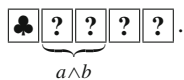
1. Arrange the cards of the two input commitments and the additional card as




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

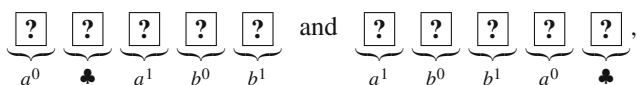


3. Reveal the card at position 1.
 - (a) If the card is , then the cards at positions 2 and 3 constitute a commitment to $a \wedge b$:













































- (b) If the card is , 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 . There are two possibilities due to the outcome of (2, 3)-division shuffle:



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

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



Input (<i>a, b</i>)	Card sequences									
	<i>a</i> ⁰		<i>a</i> ¹	<i>b</i> ⁰	<i>b</i> ¹	<i>a</i> ¹	<i>b</i> ⁰	<i>b</i> ¹	<i>a</i> ⁰	
(0, 0)										
(0, 1)										
(1, 0)										
(1, 1)										

position 1 is  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 $a \wedge b$.


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  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 , 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  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 , the sequence of five cards

$$\left[\begin{array}{c} \heartsuit \\ \heartsuit \\ \heartsuit \\ \heartsuit \\ \heartsuit \end{array} \right] \tag{3}$$

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

Therefore, the card at position 4 indicates the value of $a \wedge b$, i.e., if the card at position 4 is , then $a \wedge b = 0$, and if the card is , 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,

Table 3 Possible sequences when the CHL AND protocol fails

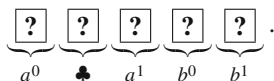
Input (a, b)	Sequence of five cards
(0, 0)	
(0, 1)	
(1, 0)	
(1, 1)	

as follows. Note that, as given in Table 3, the position of the face-down card \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 \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).

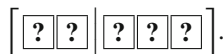
3.3 Improved protocol

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

1. Arrange the five cards as follows:

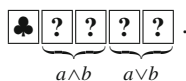


2. Apply (2, 3)-division shuffle:



3. Reveal the card at position 1.

- (a) If the card is \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 $a \vee b$:



- (b) If the card is \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 \heartsuit appears, then $f(a, b) = 1$; otherwise, $f(a, b) = 0$.

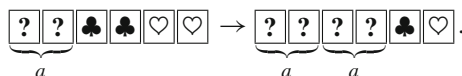
Because this protocol is neither somewhat committed-format nor non-committed format, we may call it a *hybrid protocol*. From this hybrid protocol, we can immediately derive two five-card protocols; the first one is a two-bit output

(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 (Francis et al. 2017) 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 Mizuki et al. (2012).

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 (Mizuki and Sone 2009), a commitment to bit $a \in \{0, 1\}$ can be copied with four additional cards:



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 \clubsuit s and one \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 Sect. 4.2.

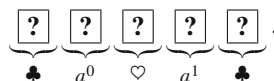
4.1 Copy protocol using unequal division shuffle

Given a commitment



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

1. Arrange the five cards as



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

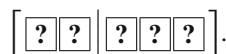
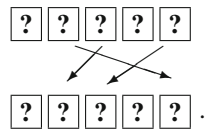


Table 4 Possible sequences after step 3 of our first copy protocol

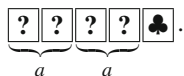
Input	Card sequences									
a	♣	♥	♣	a^1	a^0	♥	♣	a^0	♣	a^1
0	♣	♥	♣	♥	♣	♥	♣	♣	♣	♥
1	♣	♥	♣	♣	♥	♥	♣	♥	♣	♣

3. Rearrange the sequence of five cards as

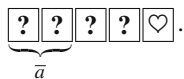


4. Reveal the card at position 5.

(a) If the card is ♣, then we have two commitments to a as follows:



(b) If the card is ♥, then we have



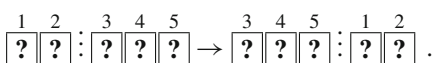
Swap the cards at positions 1 and 2 to obtain a commitment to a . After revealing the cards at positions 3 and 4 (which must be ♣♣), return to step 1.

After step 3, there are two possibilities due to the shuffle outcome: The sequence of five cards is either ♣♥♣ a^1a^0 or ♥♣ $a^0a^1a^1$. 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.

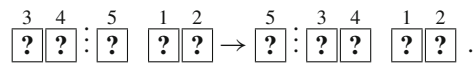
4.2 Copy protocol using double unequal division shuffle

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

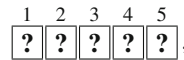
Remember that (2, 3)-division shuffle changes the order of the two portions:



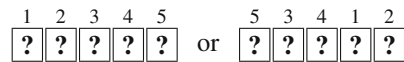
Here, we consider a further division of the three-card portion:



Thus, given a sequence of five cards



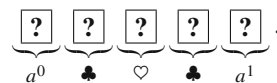
a shuffle operation resulting in either



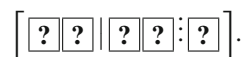
is called a *double unequal division shuffle*.

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

1. Arrange the five cards as



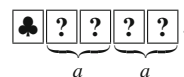
2. Apply a double unequal division shuffle:



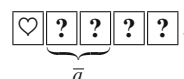
Remember that this results in one of the two possible sequences.

3. Reveal the card at position 1.

(a) If the card is ♣, then we have two commitments to a :



(b) If the card is ♥, then we have



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 $a^0♣♥♣a^1$ or $a^1♥♣a^0♣$. 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	♥	♣	♥	♣	♣	♣	♥	♣	♥	♣

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 (Sect. 5)	$2n + 1$	DUDS	2

RC Random cut, RBC random bisection cut, DUDS double unequal division shuffle

- (i) Crépeau and Kilian (1994)
- (ii) Mizuki and Sone (2009)

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 given 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 Sect. 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 Sect. 4.2. Thus, we employ a double unequal division shuffle. Specifically, given a sequence of $2n + 1$ cards

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

we use the following double unequal division shuffle:

$$\left[\boxed{?}^1 \boxed{?}^2 \mid \boxed{?}^3 \boxed{?}^4 \cdots \boxed{?}^{2n} \mid \boxed{?}^{2n+1} \right].$$

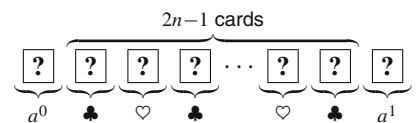
Therefore, the result of the operation must be either

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

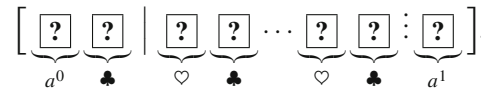
where each occurs with a probability of exactly 1/2.

The following is the procedure of our general copy protocol.

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

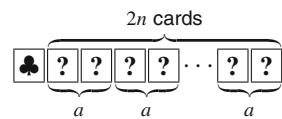


2. Apply the following double unequal division shuffle:

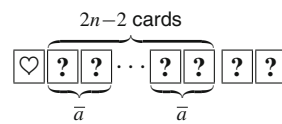


3. Reveal the card at position 1.

- (a) If the card is $\boxed{♣}$, then we have n commitments to a as follows:



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



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

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 Sect. 2.1) can be easily implemented by humans [see Ueda et al. (2016) 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.

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

a	Card sequences															
	a^0	♣	♥	...	♣	♥	♣	a^1	a^1	♥	♣	...	♥	♣	a^0	♣
0	♣	♣	♥	...	♣	♥	♣	♥	♥	♥	♣	...	♥	♣	♣	♣
1	♥	♣	♥	...	♣	♥	♣	♣	♣	♥	♣	...	♥	♣	♥	♣

Fig. 1 A box suitable for a card case

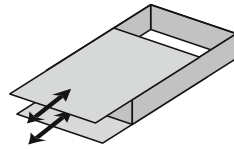
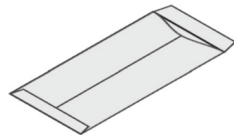


Fig. 2 An envelope suitable for a card case



Specifically, we consider the card cases shown in Fig. 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 (Fig. 1) or envelopes (Fig. 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 Nishimura et al. (2016) for the details.

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

$$\left[\begin{array}{cc|ccc} 1 & 2 & 3 & 4 & 5 \\ \hline ? & ? & ? & ? & ? \end{array} \right],$$

we must have either

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ ? & ? & ? & ? & ? \end{array} \text{ or } \begin{array}{ccccc} 3 & 4 & 5 & 1 & 2 \\ ? & ? & ? & ? & ? \end{array},$$

where each occurs with a probability of 1/2.

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

$$\begin{array}{cc} 1 & 2 \\ ? & ? \end{array} \rightarrow C_1 \mid \begin{array}{ccc} 3 & 4 & 5 \\ ? & ? & ? \end{array} \rightarrow C_2.$$

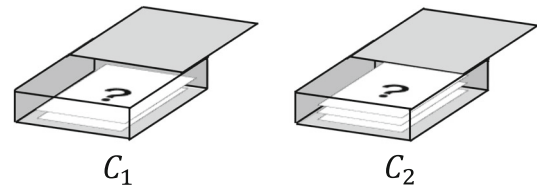


Fig. 3 Storing the two portions

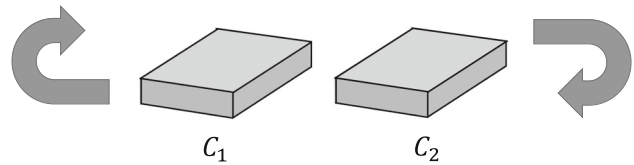


Fig. 4 Switching C_1 and C_2 randomly

Fig. 5 Stacking up the cases

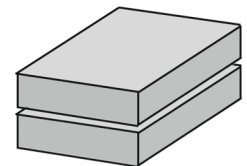
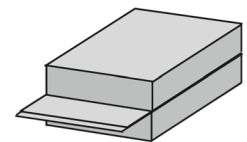


Fig. 6 Removing the two covers



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

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

where each occurs with a probability of 1/2.

3. Stack up these cases, as illustrated in Fig. 5.
4. Remove the two middle sliding covers simultaneously, as illustrated in Fig. 6. Then, we have a sequence of five cards.

As a result of this operation, we have either

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

² If players have difficulty to shuffle the two boxes publicly, they may shuffle the two boxes behind their backs (Ueda et al. 2016).

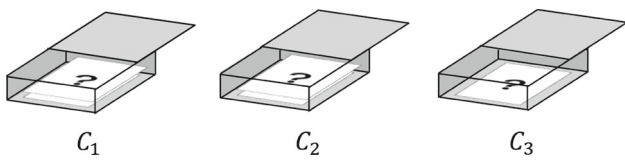
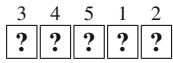


Fig. 7 Storing the three portions

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



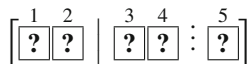
(in the case of $C_2 C_1$).

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

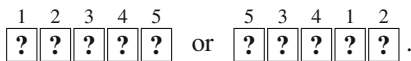
6.2 Implementation of double unequal division shuffle used in five-card copy protocol

In Sect. 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

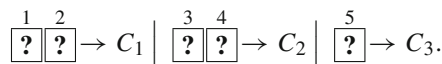


results in either



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 (Fig. 7):



2. Switch C_1 and C_3 randomly (Fig. 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.

3. Stack up these cases (without changing the order), as illustrated in Fig. 9.
4. Remove all sliding covers except for the top-most and bottom-most covers simultaneously, as illustrated in Fig. 10. Then, we have a five-card sequence.

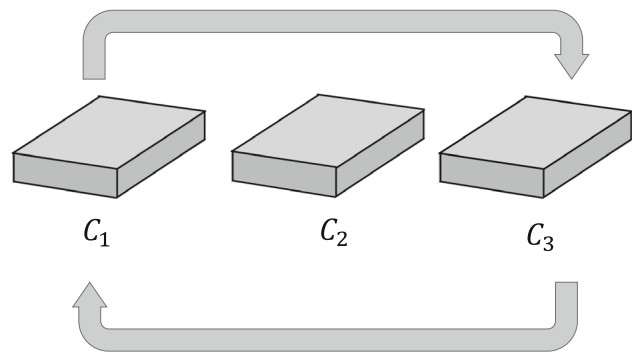


Fig. 8 Switching C_1 and C_3 randomly

Fig. 9 Stacking up the cases

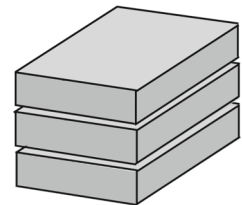
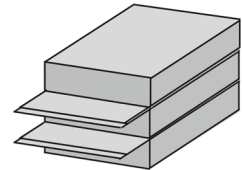
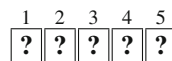


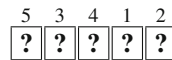
Fig. 10 Removing the four covers



As a result of this operation, we have either



(in the case of $C_1 C_2 C_3$), or



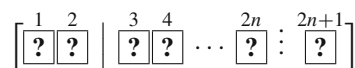
(in the case of $C_3 C_2 C_1$).

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

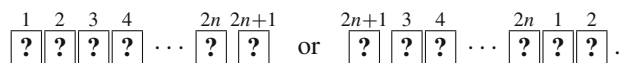
6.3 Implementation of double unequal division shuffle used in general copy protocol

We proposed a general copy protocol using $2n + 1$ cards in Sect. 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



results in either



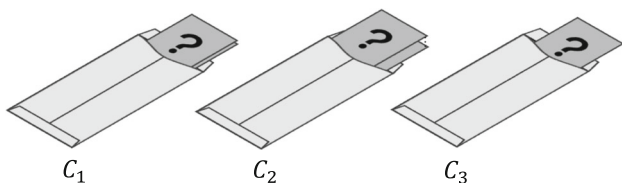


Fig. 11 Putting the three portions

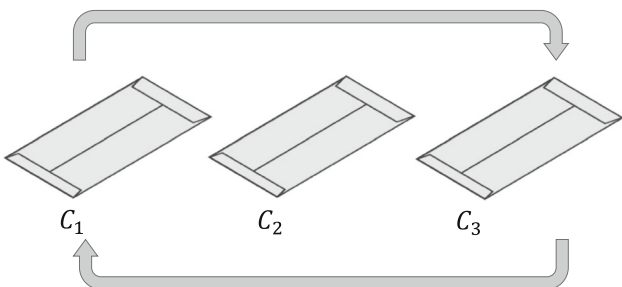


Fig. 12 Switching C_1 and C_3 randomly

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 $\begin{matrix} 3 & 4 \\ \boxed{?} & \boxed{?} \end{matrix}$ $\dots \begin{matrix} 2n \\ \boxed{?} \end{matrix}$ for $\begin{matrix} 3 & 4 \\ \boxed{?} & \boxed{?} \end{matrix}$. 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 Fig. 11:

$$\begin{matrix} 1 & 2 \\ \boxed{?} & \boxed{?} \end{matrix} \mid \begin{matrix} 3 & 4 \\ \boxed{?} & \boxed{?} \end{matrix} \dots \begin{matrix} 2n \\ \boxed{?} \end{matrix} : \begin{matrix} 2n+1 \\ \boxed{?} \end{matrix} .$$

2. Switch C_1 and C_3 randomly (Fig. 12). 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 $1/2$.

3. Heap up the three cases (without changing the order), as illustrated in Fig. 13.
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 Fig. 14. Then, we have a sequence of $2n + 1$ cards

$$\overbrace{\begin{matrix} 2n+1 \text{ cards} \\ \boxed{?} \boxed{?} \boxed{?} \boxed{?} \boxed{?} \dots \boxed{?} \boxed{?} \end{matrix}}.$$

Fig. 13 Heaping up the three cases

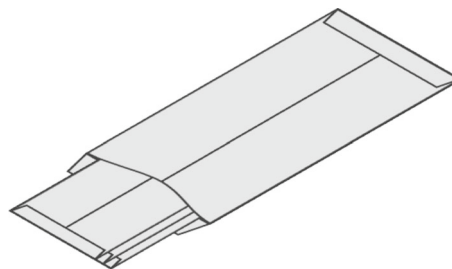
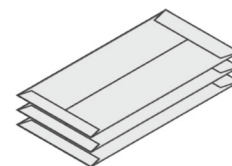


Fig. 14 Using a large envelop

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.

Acknowledgements 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 Nos. 25289068, 26330001, and 17K00001.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Cheung E, Hawthorne C, Lee P (2013) CS 758 project: secure computation with playing cards. https://csclub.uwaterloo.ca/~cdchawth/files/papers/secure_playing_cards.pdf
- Crépeau C, Kilian J (1994) Discreet solitary games. In: Stinson DR (ed) *Advances in cryptology—CRYPTO '93*, Lecture notes in computer science, vol 773. Springer, Berlin, pp 319–330. ISBN 978-3-540-57766-9. doi:10.1007/3-540-48329-2_27
- den Boer B (1990) 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. Springer, Berlin, pp 208–217. ISBN 978-3-540-53433-4. doi:10.1007/3-540-46885-4_23
- Francis D, Aljunid SR, Nishida T, Hayashi Y, Mizuki T, Sone H (2017) Necessary and sufficient numbers of cards for securely computing two-bit output functions. In: Phan RC-W, Yung M (eds) *Paradigms in cryptology—Mycrypt 2016 malicious and exploratory cryptology second international conference*. Springer, Cham, pp 193–211. ISBN 978-3-319-61273-7. doi:10.1007/978-3-319-61273-7_10
- Koch A, Walzer S, Härtel K (2015) Card-based cryptographic protocols using a minimal number of cards. In: Iwata T, Cheon JH (eds) *Advances in cryptology—ASIACRYPT 2015*, Lecture notes in computer science, vol 9452. Springer, Berlin, pp 783–807. ISBN 978-3-662-48796-9. doi:10.1007/978-3-662-48797-6_32
- Mizuki T, Sone H (2009) Six-card secure AND and four-card secure XOR. In: Deng X, Hopcroft JE, Xue J (eds) *Frontiers in algorithmics*, Lecture notes in computer science, vol 5598. Springer, Berlin, pp 358–369. ISBN 978-3-642-02269-2. doi:10.1007/978-3-642-02270-8_36
- Mizuki T, Kumamoto M, Sone H (2012) 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. Springer, Berlin, pp 598–606. ISBN 978-3-642-34960-7. doi:10.1007/978-3-642-34961-4_36
- Mizuki T, Asiedu IK, Sone H (2013) Voting with a logarithmic number of cards. In: Mauri G, Dennunzio A, Manzoni L, Porreca AE (eds) *Unconventional computation and natural computation*, Lecture notes in computer science, vol 7956. Springer, Berlin, pp 162–173. ISBN 978-3-642-39073-9. doi:10.1007/978-3-642-39074-6_16
- Niemi V, Renvall A (1998) Secure multiparty computations without computers. *Theor Comput Sci* 191(1–2):173–183. ISSN 0304-3975. doi:10.1016/S0304-3975(97)00107-2
- Nishida T, Mizuki T, Sone H (2013) Securely computing the three-input majority function with eight cards. In: Dediu A-H, Martín-Vide C, Truthe B, Vega-Rodríguez MA (eds) *Theory and practice of natural computing*, Lecture notes in computer science, vol 8273. Springer, Berlin, pp 193–204. ISBN 978-3-642-45007-5. doi:10.1007/978-3-642-45008-2_16
- Nishimura A, Nishida T, Hayashi Y, Mizuki T, Sone H (2015) Five-card secure computations using unequal division shuffle. In: Dediu A-H, Magdalena L, Martín-Vide C (eds) *Theory and practice of natural computing*, Lecture notes in computer science, vol 9477. Springer, Cham, pp 109–120. ISBN 978-3-319-26840-8. doi:10.1007/978-3-319-26841-5_9
- Nishimura A, Hayashi Y, Mizuki T, Sone H (2016) An implementation of non-uniform shuffle for secure multi-party computation. In: *Proceedings of the 3rd ACM international workshop on Asia public-key cryptography, AsiaPKC '16*, pp 49–55. ACM, New York. ISBN 978-1-4503-4286-5. doi:10.1145/2898420.2898425
- Stiglic A (2001) Computations with a deck of cards. *Theor Comput Sci* 259(1–2):671–678. ISSN 0304-3975. doi:10.1016/S0304-3975(00)00409-6
- Ueda I, Nishimura A, Hayashi Y, Mizuki T, Sone H (2016) How to implement a random bisection cut. In: Martín-Vide C, Mizuki T, Vega-Rodríguez MA (eds) *Theory and practice of natural computing*. Springer, Cham, pp 58–69. ISBN 978-3-319-49001-4. doi:10.1007/978-3-319-49001-4_5