Card-Based Cryptography Meets 3D Printer*

Yuki Ito[1], Hayato Shikata[1], Takuo Suganuma[1], and Takaaki Mizuki[1]

Tohoku University, Sendai, Japan
mizuki+lncs[atmark]tohoku.ac.jp

Abstract. Card-based protocols perform cryptographic functionalities, such as secure computations, using a deck of cards. Basically, these protocols are supposed to be implemented by humans' manipulating physical cards. This paper is the first attempt to make use of a 3D printer for better physical implementations of card-based protocols: we have designed and fabricated a couple of physical devices using a 3D printer that are useful for humans to implement protocols. The first device we created is the "five-card-trick turner," which can turn over five cards simultaneously in an amusing manner; this operation appears in the final step of the five-card trick, which is the most famous card-based protocol. The second device we created is a special card box for storing a pile of cards, whose concept was proposed in 2015 but the device had not been created in reality thus far. The special boxes can be used for implementing complex shuffles that seem difficult to implement only by hand. Furthermore, we propose another use of these special boxes so that we can efficiently perform secure computations of symmetric functions.

Keywords: Card-based cryptography · 3D printer · Secure computation · Symmetric function

1 Introduction

The development of *card-based cryptography*, which performs cryptographic functionalities such as secure computations using a deck of cards, has continued in recent years (e.g. [7, 8]). Card-based cryptography is a very unconventional type of computation.

Numerous card-based cryptographic protocols have been developed (cf. [3, 10, 16]), and many of them are simple enough to be easily implemented by non-experts, including high school students. Such protocols are implemented by humans' manipulating a physical deck of cards (along with a table on which the cards are placed). This paper begins with a description of the five-card trick [1] as a concrete example of a practical card-based protocol.

* This paper appears in Proceedings of UCNC 2024. This version of the contribution has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-031-63742-1_6. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use: https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms.

1.1 The Five-Card Trick

The five-card trick [1] is the first card-based protocol in history; it performs a secure two-input AND computation using five cards.

Suppose that Alice and Bob want to perform a secure AND computation. That is, Alice and Bob holding private bits $a \in \{0, 1\}$ and $b \in \{0, 1\}$, respectively, want to know only the AND value $a \wedge b$. Using two cards $\boxed{\clubsuit}\boxed{\heartsuit}$ of different colors, whose backs are both $\boxed{?}$, Alice secretly creates a *commitment* to her private bit $a \in \{0, 1\}$

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

which follows the following encoding rule:

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

That is, the pair of face-down cards is a commitment to a, and its order is $\boxed{\clubsuit}\boxed{\heartsuit}$ when $a = 0$ and $\boxed{\heartsuit}\boxed{\clubsuit}$ when $a = 1$. Bob creates a commitment to $b \in \{0, 1\}$ in the same way, and another red card $\boxed{\heartsuit}$ is placed in the middle:

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

Given such a sequence of five cards as input [i], the five-card trick proceeds as follows.

1. The leftmost two cards are swapped so that the commitment to a is converted to its negation $\overline{a}$:

$$\begin{array}{c}\boxed{?}\boxed{?}\boxed{\heartsuit}\boxed{?}\boxed{?}\\ \searrow\!\!\!\!\times\\ \underbrace{\boxed{?}\boxed{?}}_{\overline{a}}\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{b}\end{array}.$$

2. Turn over the center ♥ card:

$$\underbrace{\boxed{?}\boxed{?}}_{\overline{a}}\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{b} \quad \rightarrow \quad \underbrace{\boxed{?}\boxed{?}}_{\overline{a}}\boxed{?}\underbrace{\boxed{?}\boxed{?}}_{b}.$$

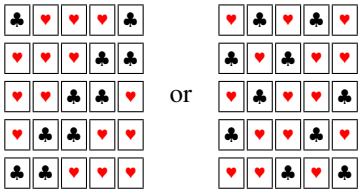
   Note that $a \wedge b = 1$ if and only if the three cards in the middle are $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$.
3. Apply a *random cut* (denoted by the symbol $\langle\cdot\rangle$):

$$\left\langle \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?} \right\rangle.$$

   A random cut is a cyclic shuffling operation, where the sequence of cards is shifted by a random number. A random cut can be easily implemented in the real world by the so-called *Hindu cut* [20].

[i] The original paper [1] used $\boxed{\spadesuit}$ instead of $\boxed{\clubsuit}$, and put $\boxed{\spadesuit}$ in the middle. We place $\boxed{\heartsuit}$ in the middle because three heart suits $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$ may be more convincing when announcing $a \wedge b = 1$.

4. Turn over all the five cards and obtain the value of $a \wedge b$ as follows:



$a \wedge b = 1$ $a \wedge b = 0$.

This AND protocol is very simple and can be easily implemented with five physical cards by human hand, as shown in the picture in Fig. 1.

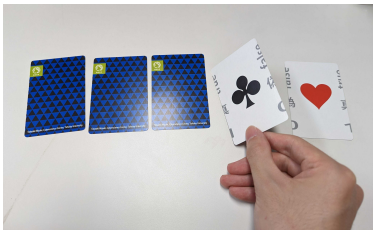


Fig. 1. An implementation of the final step of the five-card trick

In this paper, we use a 3D printer to create a device, called the "five-card-trick turner," that turns over five cards at once in the final step, as mentioned later.

1.2 Special Boxes for Implementing Complex Shuffles

In the computational model of card-based protocols [9], a "shuffle" is mathematically defined, and the model includes "complex" shuffles which seem difficult for humans to implement only by hand. On the other hand, somewhat interestingly, it is known that making use of such complex shuffles leads to protocols having a smaller number of cards or shuffles (e.g. [6, 13, 19]). Furthermore, it was pointed out that some of such complex shuffles could be implemented (in the real world) by using special card boxes illustrated in Fig. 2a [12–14].

The main feature of this box is that several boxes can be stacked and piles of cards stored in boxes can be combined into a single pile, as shown in Fig. 2b.

To the best of our knowledge, such a special card box had never been made in the real world. Thus, we created such physical boxes using a 3D printer for the first time.

1.3 Contribution of This Paper

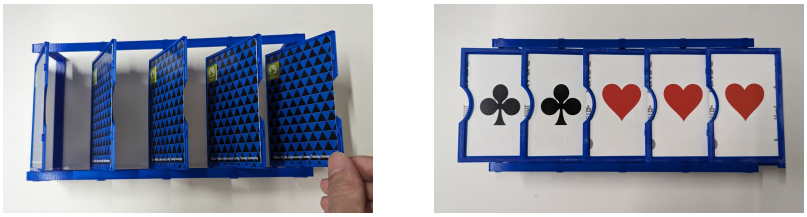
As mentioned in Sects. 1.1 and 1.2, this paper reports that, using a 3D printer, we designed and created two physical devices useful for implementing card-based protocols: the *five-card-trick turner* and the *special card boxes* for storing and combining piles.

(a) Special box whose lid and bottom can be slid. (b) Stack boxes and combine piles of cards.

Fig. 2. The special card boxes

Five-card-trick turner

We created a device for flipping five cards simultaneously in an amusing manner in the final step of the five-card trick; we name it the five-card-trick turner, and our actually created one is shown in Fig. 3. Due to the page limitation, we omit the details in this paper. As mentioned above, the five-card trick is very simple, and it has attracted many lay-people such as high school students; thus, we expect that this new device will further increase the appeal of the five-card trick.



(a) Flipping five face-down cards (b) Revealed five cards

Fig. 3. The five-card-trick turner

Special card boxes

As mentioned in Sect. 1.2, the special card boxes illustrated in Fig. 2 were considered to potentially implement complex shuffles (some of which could theoretically reduce the numbers of required cards or shuffles), but we had not seen any real implementation. We then actually created special card boxes using a 3D printer, as shown in Fig. 4.

As a demonstration, using the created boxes, we have actually executed the existing five-copy protocol [13], which requires a complex shuffle, and confirmed that the protocol can be executed securely and reliably.

In addition, we propose a novel use of these special card boxes: we show that they can be used for securely computing any symmetric function efficiently. Specifically, in terms of the number of times we make a pile of cards, the existing method [15] requires $n^2/2 + 3n/2 - 2$ times, whereas our method uses only $2n - 2$ times. This result is very interesting from both theoretical and practical perspectives.

Fig. 4. Created special card boxes for complex shuffles

Let us emphasize that the main scientific contribution of this paper lies in revealing how new physical devices can enable a novel and efficient protocol.

1.4 Organization of This Paper

The rest of this paper is organized as follows. In Sect. 2, to be familiar with the computational model of card-based protocols, we show a pseudo-code for the five-card trick. Next, in Sect. 3, we report on the creation of the special card boxes on a 3D printer and the actual execution of the five-card copy protocol. Next, in Sect. 4, we show that the special card boxes provide "sorting functionality," (whereby we will construct an efficient protocol for any symmetric function in Sect. 6). Next, in Sect. 5, we introduce the existing protocol for symmetric functions. Then, in Sect. 6, we present our protocol for symmetric functions based on the special card boxes. Finally, we conclude in Sect. 7.

2 Pseudo-Code for the Five-Card Trick

In this section, we give a formal description of the five-card trick introduced in Sect. 1.1 using a pseudo-code as shown in Pseudo-code 1. Remember that this protocol uses five cards $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}$ and each input commitment is placed using two cards $\boxed{\clubsuit}\boxed{\heartsuit}$ according to the encoding rule (1).

Here, $(\mathsf{perm}, (1\,2))$ means an action permuting a sequence of cards based on the cyclic permutation $(1\,2)$, and $(\mathsf{turn}, \{3\})$ means to turn over the third card. The next action

$$(\mathsf{shuf}, \{\mathsf{id}, (1\,2\,3\,4\,5), (1\,2\,3\,4\,5)^2, (1\,2\,3\,4\,5)^3, (1\,2\,3\,4\,5)^4\})$$

indicates a random cut: one of the five permutations is chosen uniformly at random and applied, where $(1\,2\,3\,4\,5)$ is a cyclic permutation and id denotes the identity. The next action $(\mathsf{turn}, \{1, 2, 3, 4, 5\})$ means that all five cards are turned over.

3 Special Card Boxes for Complex Shuffles

In card-based cryptography, complex shuffles have sometimes been used in protocol construction, especially when designing protocols with extremely small numbers of

Pseudo-code 1. Five-card trick

Input: $\underbrace{?\,?}_{a}\ \heartsuit\ \underbrace{?\,?}_{b}$
(perm, (1 2))
(turn, {3})
(shuf, {id, (1 2 3 4 5), $(1\,2\,3\,4\,5)^2$, $(1\,2\,3\,4\,5)^3$, $(1\,2\,3\,4\,5)^4$})
(turn, {1, 2, 3, 4, 5})
if ♥♥♥ appears **then**
(result, "$a \wedge b = 1$")
else
(result, "$a \wedge b = 0$")
endif

Pseudo-code 2. Five-card copy

Input: $\underbrace{?\,?}_{a}\ \clubsuit\ \clubsuit\ \heartsuit$
(perm, (1 2 4 5 3))
(shuf, {id, (1 4 2 5 3)})
(perm, (2 5 3))
(turn, {5})
if ♣ appears **then**
(result, (1, 2), (3, 4))
else
(perm, (1 2))
(turn, {5})
return to the beginning
endif

cards or shuffles. In this section, we report that, using a 3D printer, we actually created special card boxes for implementing such complex shuffles.

3.1 Complex Shuffles by Special Card Boxes

A card-based protocol is formulated by an abstract machine [9]. Specifically, as seen in the pseudo-code of the five-card trick shown in Sect. 2, a protocol consists mainly of a combination of three actions perm, turn, and shuf. Of these, shuf is said to be the most important action, and a shuffle is formulated as $(\mathsf{shuf}, \Pi, \mathcal{F})$ using a set of permutations Π and a probability distribution $\mathcal{F}$ on it. That is, a permutation $\pi \in \Pi$ is chosen according to the distribution $\mathcal{F}$, and π is applied to the sequence of cards.

Thus, Π and $\mathcal{F}$ in a shuffle $(\mathsf{shuf}, \Pi, \mathcal{F})$ in the computational model of card-based protocols can be arbitrary. If $\mathcal{F}$ is a uniform distribution, the shuffle is said to be *uniform* and is sometimes written as (shuf, Π) for short. If the set of permutations Π is closed, i.e., a subgroup of permutations, we say that the shuffle is *closed*.

A non-uniform or non-closed shuffle is often called a *complex* shuffle and is difficult for humans to implement. The first concrete complex shuffle in the history of card-based cryptography is $(\mathsf{shuf}, \{\mathsf{id}, (1\,4\,2\,5\,3)\})$. This is a shuffle in which the left two cards and the remaining three cards in a sequence of five cards are considered as piles, and the two piles are swapped with a probability of $1/2$:

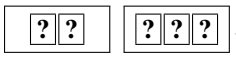
$$\begin{array}{c}1\ 2\ 3\ 4\ 5\\ ?\,?\,?\,?\,?\end{array} \quad \rightarrow \quad \left\{ \begin{array}{c} \begin{array}{c}1\ 2\ 3\ 4\ 5\\ ?\,?\,?\,?\,?\end{array} \\ \begin{array}{c}3\ 4\ 5\ 1\ 2\\ ?\,?\,?\,?\,?\end{array} \end{array} \right. .$$

This shuffle is uniform but not closed (because $(1\,4\,2\,5\,3)^2 \neq \mathsf{id}$), and is difficult to implement with human hands (e.g., if you make a pile of two cards and a pile of three cards by fixing them with rubber bands and stir them together, the difference in the

number of cards will tell you whether the two piles have been switched or not). This shuffle was presented by Eddie Cheung, Christa Hawthorne, and Patrick Lee, students in D. R. Stinson's class CS 758: Cryptography / Network Security (Fall Semester, 2013) project at the University of Waterloo.

Later in 2015, this shuffle was used to construct a five-card copy protocol [13], which makes two copied commitments from a given commitment (before this protocol, the copy protocol using the fewest cards required six cards [11]). A pseudo-code of the five-card copy protocol is shown in Pseudo-code 2, where $(\mathsf{result}, (1,2), (3,4))$ indicates that the first and second cards are a commitment to a, and so are the third and fourth.

The first mention of how to physically implement the shuffle $(\mathsf{shuf}, \{\mathsf{id}, (1\,4\,2\,5\,3)\})$ was made in the paper [13], and it was expected that it could actually be implemented using two special card boxes like Fig. 2a in Sect. 1. Specifically, a pile of two cards and a pile of three cards are each placed in a box and switch them randomly:



.

Then, slide the top and bottom of the boxes to merge the two piles into a single pile, as shown in Fig. 2b in Sect. 1. Although how to implement is shown in this way, to the best of our knowledge, such a card box has never been physically created and implemented in the real world.

In the paper [12], the conditions on special card boxes that must be satisfied are listed on p. 1498. The following is a quote from that part, where this special box-shaped equipment is referred to as a 'case.'

> "..., we assume physical cases that satisfy the following properties.
> 1. It is possible to store a pile of cards in a case without changing its order.
> 2. It is possible to take out a pile of cards from a case without changing its order.
> 3. It is possible to take out multiple piles without changing their orders by opening multiple cases at the same time. No information leak will be caused by this action.
> 4. A number of cases that possibly contain piles of cards are indistinguishable from one another, and we cannot obtain any information about the cards inside.

Based on the above, we have designed a special card box, as will be described in Sect. 3.2.

3.2 Design and Creation of Special Boxes

Basically, the special box was designed to meet the requirements described in Sect. 3.1. In addition, design innovations were added to ease protocol execution. Specifically, notches are included at the top and bottom of the stacked boxes to prevent dislodging during operation. In addition, when sliding the bottom or lid of the stacked boxes, the other slides in tandem with it. The productions are shown in Fig. 4 in Sect. 1.

We added another useful feature: the ejector shown in Fig. 5, which helps securely eject a pile of cards from boxes. If we flip the card box to eject a pile of cards, then it

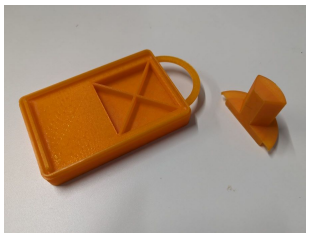


Fig. 5. Ejector device and support equipment

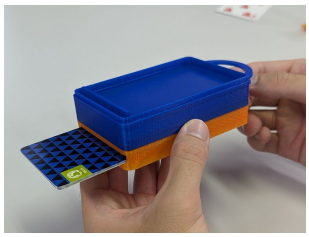


Fig. 6. Removing cards using the ejector

may lead to a partial view of the cards and protocol failure. Instead, by using the ejector, cards can be securely ejected without the need to flip the box. The process of ejecting piles of cards using these devices is shown in Fig. 6.

3.3 Real Implementation of the Five-Card Copy Protocol

Fig. 7 shows an actual implementation of the complex shuffle $(\mathsf{shuf}, \{\mathsf{id}, (1\,4\,2\,5\,3)\})$. Based on this, we have actually implemented the five-card copy protocol described in Sect. 3.1. We think that this was the first secure physical implementation of the five-card copy protocol in history.

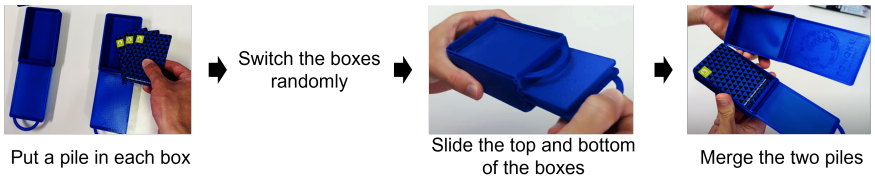


Fig. 7. Implementation of the complex shuffle with special boxes

4 Sorting Piles Secretly with Special Boxes

In this section, we consider another use of the special card box: we introduce a new mechanism for storing a card on top of each box whereby we can sort a sequence of piles secretly according to the order of some face-down cards.

4.1 Attaching Card on Top of Box

We consider attaching a face-down card to a special card box, as shown in Fig. 8. The attached card could be fixed with the box using a rubber band or something, but, using a 3D printer, we introduced a new feature to realize this more easily.

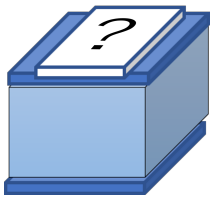


Fig. 8. Attach a card to the top of the box

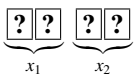
Fig. 9. Card housing mechanism

Specifically, we have created a card box with a mechanism for storing a card in the top of the box, as shown in Fig. 9. The card storage mechanism is attached to the top lid of the box, allowing a single card to be inserted. This structure improves preventing card removal and other protocol errors when shuffling card boxes.

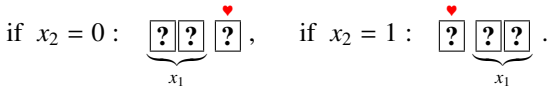
4.2 Example of Box Use

Here, we give a concrete example of making use of the feature of the special box explained in Sect. 4.1.

Given commitments to $x_1, x_2 \in \{0, 1\}$

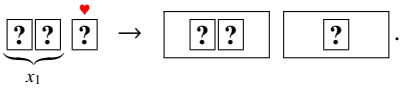


along with $\boxed{\heartsuit}$, we want to "sort" the commitment to x_1 and the face-down $\boxed{\heartsuit}$ according to the value of x_2 while keeping the value of x_2 secret. That is, we want the following to be:

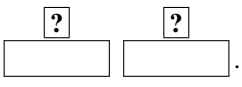


This can be done as follows.

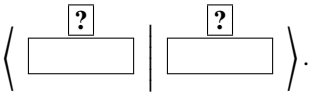
1. Open the tops of two special card boxes, insert the commitment to x_1 and the face-down ♥ card, and close the tops:



2. Place the left card of the commitment to x_2 on top of the left box (by the storage mechanism) and the right card to the top of the right box:



3. Shuffle the two special boxes until they are no longer clear which is which:



4. Take out the cards at the tops of the boxes and turn them over; then, one is ♣ and the other ♥. Stack the two boxes so that the ♥ box is under the ♣ box, and pull out the middle dividers to merge the two piles inside the boxes into a single pile.

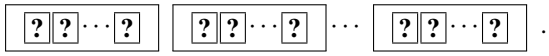
This procedure accomplishes the above.

4.3 General Description

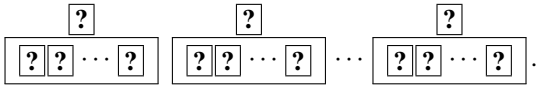
Here, we generalize the method mentioned in Sect. 4.2. That is, we now demonstrate that we can sort k piles (of possibly different sizes) based on face-down numbered cards ranging from 1 to k.

Assume that there are k piles of arbitrary sizes, and that we have k face-down numbered cards arranged by the numbers from 1 to k in a specific order.

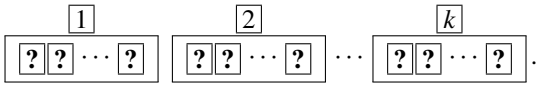
1. Place each pile in a special box:



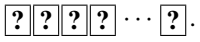
2. Place the face-down numbered card on top of each special box (without changing the order):



3. Shuffle the k special boxes, take out the numbered cards, turn them over, and sort the boxes in ascending order based on the revealed numbered cards:



4. All the special boxes are concatenated and the partitions are pulled out to obtain a sorted sequence of cards:



For piles of the same size, the sort sub-protocol proposed in [5] can be used, but for piles of irregular sizes, the method described in this paper is effective.

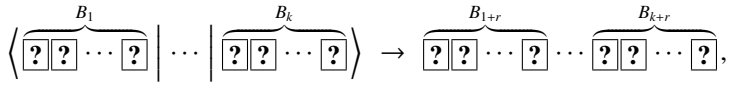
As seen later, our method above will provide an efficient addition of commitments, by which we can construct an efficient protocol for any symmetric function. To compare our protocol with the existing one, we first introduce the existing protocol in Sect. 5, followed by our protocol presented in Sect. 6.

5 Secure Computation of Symmetric Function: Existing Method

Ruangwises and Itoh [15] proposed a generic protocol for any symmetric function $f : \{0,1\}^n \to R$ such that its range R can be an arbitrary set. In this section, we introduce this existing protocol. Note that a function is said to be *symmetric* if it is invariant under permutations of its variables.

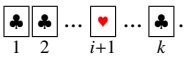
5.1 Preliminaries

We first introduce the *pile-shifting shuffle* [12, 18]. Suppose that there are k piles $B_1, \cdots, B_k$ of the same size. Applying a pile-shifting shuffle to this results in



where $r \in \{0, 1, \cdots, k-1\}$ is random and if the subscript exceeds k, the value shall return to 1. The pile-shifting shuffle can be implemented by fixing piles with envelopes, rubber bands, or sleeves and applying the Hindu cut to them.

We next explain how to express a non-negative integer as a sequence of cards [15]. Let $k \geq 2$; then, using $k-1$ ♣ cards and a ♥ card, we represent an integer i, $0 \leq i \leq k-1$, by placing ♥ at the $(i+1)$-th position:

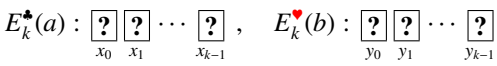


Hereinafter, such a sequence of face-down cards is denoted by $E_k^{\heartsuit}(i)$. If we reverse the colors, we denote the resulting sequence by $E_k^{\clubsuit}(i)$.

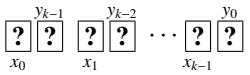
5.2 Addition of Non-Negative Integers

Ruangwises and Itoh [15] proposed a method for performing addition, given two face-down sequences representing non-negative integers. (This method is based on the idea of Shinagawa et al. [17].)

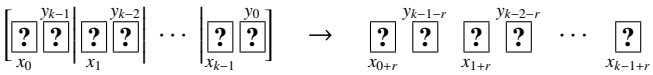
1. Given $E_k^{\clubsuit}(a)$ and $E_k^{\heartsuit}(b)$ representing two non-negative integers a, b, for the sake of explanation, we name each card in each sequence as follows:



2. Rearrange the cards as follows:



3. Apply a pile-shifting shuffle, where r is a random number:



4. Rearrange the cards as in the original:

$$E_k^{\clubsuit}(a-r): \underset{x_{0+r}}{\boxed{?}}\ \underset{x_{1+r}}{\boxed{?}}\ \cdots\ \underset{x_{k-1+r}}{\boxed{?}}\ , \quad E_k^{\heartsuit}(b+r): \underset{y_{0-r}}{\boxed{?}}\ \underset{y_{1-r}}{\boxed{?}}\ \cdots\ \underset{y_{k-1-r}}{\boxed{?}}\ .$$

   Here, the random value r is subtracted from a and r is added to b.
5. Turn over the cards in $E_k^{\heartsuit}(b+r)$, and the sequence of $E_k^{\clubsuit}(a-r)$ is cyclically shifted to the right by the revealed number $s = b + r$:

$$E_k^{\clubsuit}(a-r): \underset{x_{0+r}}{\boxed{?}}\ \underset{x_{1+r}}{\boxed{?}}\ \cdots\ \underset{x_{k-1+r}}{\boxed{?}} \quad \to \quad E_k^{\clubsuit}(a+b): \underset{x_{0+r-s}}{\boxed{?}}\ \underset{x_{1+r-s}}{\boxed{?}}\ \cdots\ \underset{x_{k-1+r-s}}{\boxed{?}}\ .$$

This allows a secure computation of $(a-r)+(b+r) = a+b$ without leaking the values of a and b. That is, $E_k^{\clubsuit}(a+b)$ is obtained.

Although the addition method has been explained for $E_k^{\clubsuit}(a)$ and $E_k^{\heartsuit}(b)$, it can also be performed for other color combinations such as $E_k^{\heartsuit}(a)$ and $E_k^{\clubsuit}(b)$.

5.3 Secure Computation of Symmetric Function

Let $f : \{0,1\}^n \to R$ be a symmetric function. We want to construct a protocol that outputs only the value of $f(x_1, \ldots, x_n)$, given n commitments to $x_1, \ldots, x_n$:

$$\underbrace{\boxed{?}\boxed{?}}_{x_1}\ \underbrace{\boxed{?}\boxed{?}}_{x_2} \cdots \underbrace{\boxed{?}\boxed{?}}_{x_n} \ \to\ \cdots\ \to\ f(x_1, \ldots, x_n)\,,$$

where each commitment follows the encoding rule (1).

Since f is symmetric, it is well known that the output value of $f(x_1, \ldots, x_n)$ depends only on the sum $\sum_{i=1}^{n} x_i$. That is, there exists a function $g : \{0, 1, \ldots, n\} \to \{0, 1\}$ such that $f(x_1, \ldots, x_n) = g\left(\sum_{i=1}^{n} x_i\right)$. Therefore, if we want to calculate the symmetric function f, we only need to find the sum $\sum_{i=1}^{n} x_i$.

Thus, it suffices to obtain a sequence $E_{n+1}^{\clubsuit}(x_1 + \cdots + x_n)$ or $E_{n+1}^{\heartsuit}(x_1 + \cdots + x_n)$ from the input commitments. Using the addition of non-negative integers described in Sect. 5.2, we can generate $E_{n+1}^{\heartsuit}(x_1 + \cdots + x_n)$ with two additional cards [15]. Although the details are omitted, the value of $f(x_1, \ldots, x_n)$ can be obtained by dividing, shuffling, and turning over cards, based on the above function g [15].

Remember that when applying a pile-shifting shuffle, we have to make a pile of cards using an envelope or something. During an execution of the existing protocol, the number of times making piles is $n^2/2 + 3n/2 - 2$. In practice, making a pile, i.e., fixing a number of cards using an envelope, is a time-consuming operation, and hence, we want to reduce this repetition. To this end, the special card boxes will be used for reducing the number of times making piles in our proposed protocol presented in the next section.

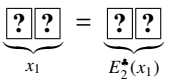
6 Secure Computation of Symmetric Function: Our Method

In this section, making use of the special card boxes described in Sect. 4, we propose a generic protocol for any symmetric function $f : \{0,1\}^n \to R$. Our protocol uses only one additional card and two special boxes; the number of piles made is $2n - 2$.

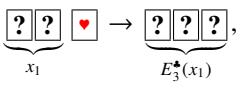
6.1 Adding Commitments with Special Boxes

As explained in Sect. 5.3, to securely compute a symmetric function, it suffices to obtain $E^{\clubsuit}_{n+1}(x_1 + \cdots + x_n)$ or $E^{\heartsuit}_{n+1}(x_1 + \cdots + x_n)$. Therefore, we give only a procedure for generating the sum $E^{\clubsuit}_{n+1}(x_1 + \cdots + x_n)$ from input commitments.

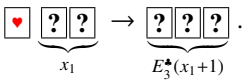
Note that a commitment to x_1 can be regarded as $E^{\clubsuit}_2(x_1)$:



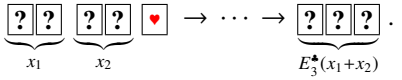
Note furthermore that placing a face-down ♥ card to the right of it results in $E^{\clubsuit}_3(x_1)$:



and placing a face-down ♥ card to the left results in $E^{\clubsuit}_3(x_1 + 1)$:

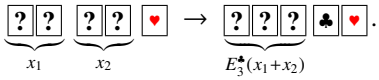


Therefore, notice that the example described in Sect. 4.2 is actually a procedure which produces $E^{\clubsuit}_3(x_1 + x_2)$:

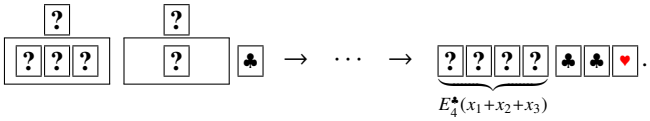


We extend this idea; given n commitments along with one additional card, using two special boxes, we obtain $E^{\clubsuit}_{n+1}(x_1 + \cdots + x_n)$ as follows.

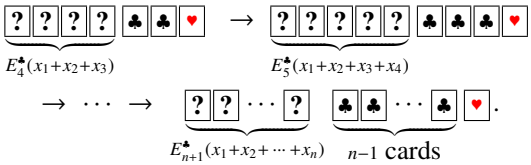
1. Add two commitments to x_1 and x_2 using the method described in Sect. 4.2, and obtain $E^{\clubsuit}_3(x_1 + x_2)$:



2. Place $E^{\clubsuit}_3(x_1 + x_2)$ obtained in Step 1 in the left box and the ♥ card in the right box, place the commitment to x_3 on tops of the boxes, and apply the addition in a similar way:



3. Repeat such an operation until x_n is reached:



This is our protocol.

The existing protocol [15] requires two additional cards, while our protocol uses only one additional card, i.e., it reduces the number of cards by one by using two special boxes.

Also, the number of times to make piles (namely, the number of times a pile is placed in a sleeve or envelope) is $n^2/2 + 3n/2 - 2$ times in the existing protocol [15], while the number of times to make piles (namely, the number of times we place cards in a special box) is $2n - 2$ times. In terms of this, our protocol is very efficient[ii].

Table. 1 summarizes their performances.

Table 1. The existing protocol and our protocol for symmetric functions

	# of cards	# of making a pile
Ruangwises–Itoh [15]	$2n + 2$	$n^2/2 + 3n/2 - 2$
Ours	$2n + 1$	$2n - 2$

7 Conclusion

In this paper, we reported that, using a 3D printer, we created the five-card-trick turner and the special card boxes useful for implementing card-based protocols. We also proposed another use of the special card boxes so that we can efficiently perform secure computations of symmetric functions. Specifically, our protocol needs only $2n-2$ times for making piles while the existing protocol needs $n^2/2 + 3n/2 - 2$ times.

In order for secure computations or other cryptographic technologies to be widely used in society, it is necessary for a wide range of stakeholders to understand the meaning and significance of these cryptographic functionalities [2]. We hope that card-based cryptography and its implementation will help in this regard.

Acknowledgements

We thank the anonymous referees, whose comments have helped us improve the presentation of the paper. This work was supported in part by JSPS KAKENHI Grant Numbers JP24K02938 and JP23H00479.

References

1. Boer, B.D.: More efficient match-making and satisfiability the five card trick. In: EUROCRYPT 1989. LNCS, vol. 434, pp. 208–217. Springer, Heidelberg (1990)

[ii] Although making piles when applying a pile-shifting shuffle is a reasonable implementation, it should be noted that this metric depends on implementations in a sense (cf. [4]).

2. Hanaoka, G.: Towards user-friendly cryptography. In: Paradigms in Cryptology–Mycrypt 2016. Malicious and Exploratory Cryptology. LNCS, vol. 10311, pp. 481–484. Springer, Cham (2017)
3. Koch, A.: Cryptographic Protocols from Physical Assumptions. Ph.D. thesis, Karlsruhe Institute of Technology (2019)
4. Koch, A., Walzer, S.: Foundations for actively secure card-based cryptography. In: Fun with Algorithms. LIPIcs, vol. 157, pp. 17:1–17:23. Schloss Dagstuhl, Dagstuhl, Germany (2020)
5. Koch, A., Walzer, S.: Private function evaluation with cards. New Gener. Comput. **40**, 115–147 (2022)
6. Koch, A., Walzer, S., Härtel, K.: Card-based cryptographic protocols using a minimal number of cards. In: Advances in Cryptology—ASIACRYPT 2015. LNCS, vol. 9452, pp. 783–807. Springer, Berlin, Heidelberg (2015)
7. Mizuki, T.: Preface: Special issue on card-based cryptography. New Gener. Comput. **39**, 1–2 (2021)
8. Mizuki, T.: Preface: Special issue on card-based cryptography 2. New Gener. Comput. **40**, 47–48 (2022)
9. Mizuki, T., Shizuya, H.: A formalization of card-based cryptographic protocols via abstract machine. Int. J. Inf. Secur. **13**(1), 15–23 (2014)
10. Mizuki, T., Shizuya, H.: Computational model of card-based cryptographic protocols and its applications. IEICE Trans. Fundam. **E100.A**(1), 3–11 (2017)
11. Mizuki, T., Sone, H.: Six-card secure AND and four-card secure XOR. In: Frontiers in Algorithmics. LNCS, vol. 5598, pp. 358–369. Springer, Berlin, Heidelberg (2009)
12. Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: Pile-shifting scramble for card-based protocols. IEICE Trans. Fundam. **101**(9), 1494–1502 (2018)
13. Nishimura, A., Nishida, T., Hayashi, Y., Mizuki, T., Sone, H.: Five-card secure computations using unequal division shuffle. In: Theory and Practice of Natural Computing. LNCS, vol. 9477, pp. 109–120. Springer, Cham (2015)
14. Nishimura, A., Nishida, T., Hayashi, Y., Mizuki, T., Sone, H.: Card-based protocols using unequal division shuffles. Soft Comput. **22**, 361–371 (2018)
15. Ruangwises, S., Itoh, T.: Securely computing the n-variable equality function with $2n$ cards. Theor. Comput. Sci. **887**, 99–110 (2021)
16. Shinagawa, K.: On the Construction of Easy to Perform Card-Based Protocols. Ph.D. thesis, Tokyo Institute of Technology (2020)
17. Shinagawa, K., Mizuki, T., N., J.S.C., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Multi-party computation with small shuffle complexity using regular polygon cards. In: Provable Security. LNCS, vol. 9451, pp. 127–146. Springer, Cham (2015)
18. Shinagawa, K., Mizuki, T., Schuldt, J., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Card-based protocols using regular polygon cards. IEICE Trans. Fundam. **E100.A**(9), 1900–1909 (2017)
19. Shinagawa, K., Nuida, K.: A single shuffle is enough for secure card-based computation of any Boolean circuit. Discrete Applied Mathematics **289**, 248–261 (2021)
20. Ueda, I., Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: How to implement a random bisection cut. In: Theory and Practice of Natural Computing. LNCS, vol. 10071, pp. 58–69. Springer, Cham (2016)