Few-helping-card Protocols for Some Wider Class of Symmetric Boolean Functions with Arbitrary Ranges

Hayato Shikata
Graduate School of Information Sciences, Tohoku University
Sendai, Japan

Daiki Miyahara
The University of Electro-Communications
National Institute of Advanced Industrial Science and Technology
Tokyo, Japan

Takaaki Mizuki
Cyberscience Center, Tohoku University
National Institute of Advanced Industrial Science and Technology
Sendai, Japan

ABSTRACT

In card-based cryptography, which uses a physical deck of cards to realize secure multiparty computations, a one-bit value is usually encoded by a pair of cards. Thus, when performing a secure computation of an n-input Boolean function, a sequence of $2n$ cards representing n bits is needed for input, and some helping cards are typically added to form a protocol. In 2020, Ruangwises and Itoh constructed a card-based protocol for a symmetric Boolean function with an arbitrary range using two helping cards. (Note that a symmetric Boolean function depends only on the number of 1s in its input). At the same time, they showed that the helping cards can be eliminated if the target function is limited to "doubly symmetric" Boolean functions (also known as symmetric self-anti-dual functions). A doubly symmetric Boolean function satisfies the following for all k: when inputting exactly a number k of 1s, the output is the same as the output when inputting exactly a number $n-k$ of 1s. In this paper, we loosen the restriction on doubly symmetric Boolean functions by fixing $k = 0$, and construct new protocols which require less than two helping cards for that wider class of symmetric Boolean functions. Specifically, we design a one-helping-card protocol for any $n > 4$, and helping-card-free protocols for $n = 3$ and $n = 4$.

CCS CONCEPTS

• **Security and privacy → Cryptography**.

KEYWORDS

Card-based cryptography, Secure computation, Real-life hands-on cryptography, Card-based protocols

ACM Reference Format:
Hayato Shikata, Daiki Miyahara, and Takaaki Mizuki. 2023. Few-helping-card Protocols for Some Wider Class of Symmetric Boolean Functions with Arbitrary Ranges. In *Proceedings of the10th ACM Asia Public-Key Cryptography Workshop (APKC '23), July 10, 2023, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3591866.3593073

1 INTRODUCTION

Secure multiparty computations enable us to obtain the output value of a predetermined function while keeping information on input values secret. *Card-based cryptography* achieves secure multiparty computations using a deck of physical cards, and the research area has been significantly growing in recent years (c.f. [18, 19]); refer to [9, 10, 22, 35] for surveys.

In card-based cryptography, a one-bit value is typically represented with the order of a black card ♣ and a red card ♡ according to the two-card-per-bit encoding:

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

When two face-down cards represent a bit $x \in \{0, 1\}$ according to Eq. (1), we call these two cards a *commitment to* x and denote it by

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

where we assume that all black ♣ and red cards ♡ have the identical backs ?.

Given a number of commitments as input, a *card-based cryptographic protocol* (simply referred to as a *protocol* often hereinafter) should perform a secure multiparty computation via a series of actions, such as turning over and shuffling cards.

1.1 Protocol for Symmetric Functions

This paper deals with card-based cryptographic protocols for "symmetric" Boolean functions. An n-input Boolean function $f : \{0, 1\}^n \to R$ with some set R (as its range) is said to be *symmetric* if f satisfies the following for every i and j, $1 \le i, j \le n$:

$$f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n) = f(x_1, \ldots, x_j, \ldots, x_i, \ldots, x_n).$$

Because one bit is encoded with two cards as per the encoding rule (1) above, any protocol for an n-input Boolean function requires at least $2n$ cards. That is, it takes n commitments corresponding to n inputs $x_1, x_2, \ldots, x_n \in \{0, 1\}$

$$\underbrace{\boxed{?}\boxed{?}}_{x_1}\ \underbrace{\boxed{?}\boxed{?}}_{x_2}\ \cdots\ \underbrace{\boxed{?}\boxed{?}}_{x_n}.$$

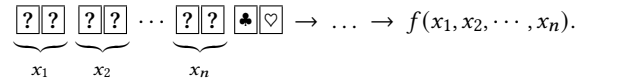
In addition to the n input commitments, some helping cards (like ♣♡♣♡) are often required as input. Thus, a protocol for an n-input Boolean function f takes n commitments to $x_1, x_2, \ldots, x_n$ along with some helping cards as input, applies a series of actions, such as revealing and shuffling cards, and outputs only the value

APKC '23, July 10, 2023, Melbourne, VIC, Australia
© 2023 Copyright held by the owner/author(s).
This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of the10th ACM Asia Public-Key Cryptography Workshop (APKC '23), July 10, 2023, Melbourne, VIC, Australia*, https://doi.org/10.1145/3591866.3593073.

of $f(x_1, x_2, \ldots, x_n)$:

When devising new protocols, a smaller number of helping cards is considered to be preferable.

In 2020, Ruangwises and Itoh [30, 32] proposed a generic way for constructing a protocol with two helping cards for any symmetric Boolean function $f : \{0,1\}^n \to R$ with an arbitrary range R:[i]

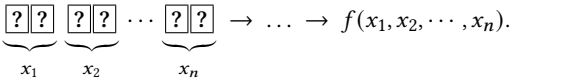


This is a non-trivial upper bound on the number of required helping cards for any symmetric Boolean function having an arbitrary range R (which is not necessarily two-valued but can be multi-valued, say $R = \{0, 1, 2, \ldots, n\}$). It is open to determine whether the upper bound, i.e., two helping cards, are necessary or could be lowered.

Ruangwises and Itoh [30, 32] also considered the class of "doubly symmetric" Boolean functions as a subclass of symmetric Boolean functions to construct a helping-card-free protocol, as follows. If a symmetric Boolean function f satisfies

$$f(x_1, x_2, \ldots, x_n) = f(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n})$$

for all $x_1, x_2, \ldots, x_n \in \{0, 1\}$, then f is called a *doubly symmetric* Boolean function (also known as a symmetric anti-self-dual function). For any doubly symmetric Boolean function $f : \{0,1\}^n \to R$ with an arbitrary range R, they constructed a protocol without any helping card:



Thus, we have a helping-card-free protocol for any function in this narrow class.

To summarize, the existing research [30, 32] gives the following theorem.

Theorem 1.1 ([30, 32]). *Let $n \geq 2$, let R be an arbitrary set, and let $f : \{0,1\}^n \to R$ be a symmetric Boolean function.*

- *There is a two-helping-card protocol for f.*
- *If f is doubly symmetric, there is a helping-card-free protocol for f.*

To provide better bounds on the number of required helping cards (than Theorem 1.1) is an important open problem.

1.2 Contribution

This paper tackles the aforementioned open problem and presents a partial answer to it. For this purpose, we define "partially doubly symmetric" Boolean functions as a wider subclass of symmetric Boolean functions than the class of doubly symmetric Boolean functions, and construct a protocol requiring less than two helping cards for any function in that wider class.

[i]It should be noted that, as will be seen in Sect. 3, the output value is not given as a simple encoding.

To define "partially doubly symmetric" Boolean functions, first, let us review the basic properties of symmetric Boolean functions. Let $f : \{0,1\}^n \to R$ be a symmetric Boolean function. The value of its output $f(x_1, x_2, \ldots, x_n)$ depends only on the number of 1s in the input, i.e., the sum $\sum_{i=1}^{n} x_i$. In other words, there exists a function $g : \{0, 1, \ldots, n\} \to R$ such that

$$f(x_1, x_2, \ldots, x_n) = g\left(\sum_{i=1}^{n} x_i\right) \tag{2}$$

for every $x_1, x_2, \ldots, x_n \in \{0, 1\}$. Bearing this in mind, observe that for a doubly symmetric Boolean function, the output value when the sum is k is equal to the one when the sum is $n - k$. Thus, for a doubly symmetric Boolean function f, the corresponding function g in Eq. (2) satisfies the following:

$$\begin{aligned} g(0) &= g(n), \\ g(1) &= g(n-1), \\ &\vdots \\ g(\lfloor (n-1)/2 \rfloor) &= g(n - \lfloor (n-1)/2 \rfloor)\,. \end{aligned} \tag{3}$$

As known from the list of equations (3) above, doubly symmetric Boolean functions represent a quite restricted subclass within the class of symmetric Boolean functions. In this paper, we relax the restriction and consider a class that imposes only the topmost restriction $g(0) = g(n)$ in the list (3), which we call "$\{0\}$-partially doubly symmetric" Boolean functions; we will construct few-helping-card protocols for this wider class.

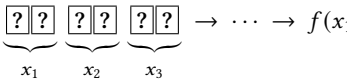
More generally, for a set $I \subseteq \{0, 1, \ldots, \lfloor (n-1)/2 \rfloor\}$, we define an *$I$-partially doubly symmetric* Boolean function, as follows.

Definition 1.2. Let $f : \{0,1\}^n \to R$ be a symmetric Boolean function, let $g : \{0, 1, \ldots, n\} \to R$ be the function satisfying Eq. (2), and let $I \subseteq \{0, 1, \ldots, \lfloor (n-1)/2 \rfloor\}$. If $g(k) = g(n-k)$ for every $k \in I$, then f is called an *I-partially doubly symmetric* Boolean function.

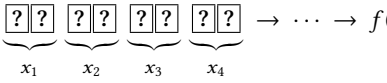
Under this definition, a doubly symmetric Boolean function is redefined as a $\{0, 1, \ldots, \lfloor (n-1)/2 \rfloor\}$-partially doubly symmetric Boolean function, and a symmetric Boolean function is an $\emptyset$-partially doubly symmetric Boolean function (where $\emptyset$ denotes the empty set).

As mentioned above, the main target of this paper is $\{0\}$-partially doubly symmetric Boolean functions. That is, we will design a protocol for any symmetric Boolean function that outputs the same value when the number of 1s in the input is 0 and when it is n; more simply, our target is any n-input symmetric Boolean function f such that $f(0, 0, \ldots, 0) = f(1, 1, \ldots, 1)$.

Specifically, we will construct few-helping-card protocols, divided into three cases, $n = 3$, $n = 4$, and $n \geq 5$. First, for the case of $n = 3$, we will show that one can obtain a protocol without any helping card (Sect. 4):



Next, for the case of $n = 4$, we will also show that we need no helping card (Sect. 5):



Next, we will design a protocol with one helping card for the case of $n \geq 5$ (Sect. 6):

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

Finally, when $n = 2$, we will state that for any 2-input symmetric Boolean function, without limiting ourselves to partially doubly symmetric Boolean functions, a helping-card-free protocol can be constructed by simply making use of the existing method (Sect. 7):

$$\underbrace{?\,?}_{x_1}\ \underbrace{?\,?}_{x_2}\ \to\ \cdots\ \to\ f(x_1, x_2)\,.$$

The following theorem summarizes our contribution in this paper.

Theorem 1.3. *Let $n \geq 2$, let R be an arbitrary set, and let $f : \{0,1\}^n \to R$ be a symmetric Boolean function.*

- *If $3 \leq n \leq 4$ and f is $\{0\}$-partially doubly symmetric, there is a helping-card-free protocol for f.*
- *If $n \geq 5$ and f is $\{0\}$-partially doubly symmetric, there is a one-helping-card protocol for f.*
- *If $n = 2$, there is a helping-card-free protocol for f.*

The above results and the existing studies are summarized in Table 1.

Before going to the next subsection, we display an example of a $\{0\}$-partially doubly symmetric Boolean function. Define a symmetric Boolean function $h : \{0,1\}^9 \to \{0,1,2\}$ as

$$h(x_1, x_2, \ldots, x_9) = \begin{cases} 1 & \text{if } \sum_{i=1}^{9} x_i = 4, \\ 2 & \text{if } \sum_{i=1}^{9} x_i = 8, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Then, h is a $\{0\}$-partially doubly symmetric Boolean function (because $h(0,0,\ldots,0) = h(1,1,\ldots,1) = 0$). A secure computation of this function h allows nine players to know only if four or eight people among them want to play a four-player game, such as mahjong, without awkwardness.

1.3 Related Work

The purpose of this paper is to construct protocols with a small number of helping cards for symmetric Boolean functions with arbitrary ranges.

Although protocols for arbitrary functions (having arbitrary domains and ranges) can be constructed if a sufficient number of helping cards are available [2], most of the existing research has focused on Boolean functions with range $\{0,1\}$, i.e., functions $f : \{0,1\}^n \to R$ with $R = \{0,1\}$. It is known that any Boolean function with range $\{0,1\}$ can be securely computed using six helping cards [25]; this is a general upper bound[ii]. It was also shown that two helping cards are sufficient when limited to symmetric Boolean functions [25]. A recent study [34] has shown that no helping card is needed for symmetric Boolean functions $f : \{0,1\}^n \to \{0,1\}$ when $n \geq 8$. There are also several specific functions for which there exist helping-card-free protocols: the two-input AND function [12, 20], the three-input AND function [7, 17], the XOR function [23], the three-input majority function [41], and the three-input equality function [6, 36].

Remember that our target functions $f : \{0,1\}^n \to R$ in this paper have arbitrary ranges R.

As mentioned at the beginning of this paper, the research area of card-based cryptography is very active recently, especially, in the following topics: private-model secure computations [1, 13, 24], zero-knowledge proof protocols [3, 28, 29, 33], novel shuffling operations [16], secure sorting [5], multi-valued protocols with a direction encoding [39], the half-open action [15], standard-deck protocols [4, 11], and applications to private simultaneous messages protocols [38].

2 PRELIMINARIES

In this section, we describe the existing encoding and methods as well as the shuffling operation our protocols use.

Card-based cryptographic protocols are formally defined via abstract machines [8, 21, 22]. Roughly speaking, a protocol consists of three actions, (turn, T), (perm, π), and (shuf, Π), which represent turning over, permuting, and shuffling cards, respectively (where T is a set of positions, π is a permutation, and Π is a set of permutations). In the sequel, for simplicity, instead of giving an abstract machine, we use a natural language to describe a protocol.

2.1 Encoding of Integer by Card Position

As mentioned in Sect. 1.2, the output value of a symmetric Boolean function depends on the sum of the input bits. Thus, to securely compute a symmetric Boolean function, given n commitments

$$\underbrace{?\,?}_{x_1}\ \underbrace{?\,?}_{x_2}\ \cdots\ \underbrace{?\,?}_{x_n}$$

and some helping cards, we want to obtain the sum $\sum_{i=1}^{n} x_i$ being kept secret. For this, we need to encode integers with cards (to maintain the sum).

The existing Ruangwises–Itoh protocols [30, 32] employ the following *integer encoding*. Suppose that $k \geq 2$ and that we want to represent an integer i, $0 \leq i \leq k-1$, with cards. Using k cards consisting of one $\clubsuit$ and $k-1$ $\heartsuit$s, the integer i is encoded by placing the $\clubsuit$ at the $(i+1)$-st as follows:

$$\overset{1}{\heartsuit}\ \overset{2}{\heartsuit}\ \cdots\ \overset{i}{\heartsuit}\ \overset{i+1}{\clubsuit}\ \overset{i+2}{\heartsuit}\ \cdots\ \overset{k}{\heartsuit}.$$

Hereinafter, such a sequence of face-down cards (representing i) is denoted by $E_k^{\clubsuit}(i)$ and written as:

$$\underbrace{?\,?\,\cdots\,?}_{E_k^{\clubsuit}(i)}\,.$$

Exchanging the colors ($\clubsuit$ and $\heartsuit$), $E_k^{\heartsuit}(i)$ is defined in a similar way.

As will be seen later, the existing Ruangwises–Itoh protocol [30, 32] and our proposed protocols perform the addition of input commitments based on the integer encoding above in order to compute symmetric Boolean functions.[iii]

[ii] If we allow protocols to fail with a high probability, there is a helping-card-free protocol for any Boolean function [12].

[iii] This integer encoding is also used in card-based zero-knowledge protocols [27, 31], secure ranking protocols [40], and card-based Yao's millionaire protocols [14].

Table 1: Numbers of helping cards required by the existing and our proposed protocols for I-partially doubly symmetric Boolean functions with n inputs

	I	n	# of helping cards
Ruangwises and Itoh [30, 32]	$\emptyset$	≥ 2	2
Ruangwises and Itoh [30, 32]	$\{0, \ldots, \lfloor \frac{n-1}{2} \rfloor\}$	≥ 2	0
This paper, §4	$\{0\}$	3	0
This paper, §5	$\{0\}$	4	0
This paper, §6	$\{0\}$	≥ 5	1
This paper, §7	$\emptyset$	2	0

Note that a commitment to $a \in \{0,1\}$ together with a free card $\boxed{\heartsuit}$ can be converted to $E_3^{\clubsuit}(a)$, i.e., place the $\boxed{\heartsuit}$ to the right of the commitment and turn it over:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\ \boxed{\heartsuit} \rightarrow \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{E_3^{\clubsuit}(a)},$$

because the commitment satisfies the encoding:

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

Similarly, a commitment to $a \in \{0,1\}$ together with a $\boxed{\clubsuit}$ can be converted to $E_3^{\heartsuit}(a)$ by swapping the two cards of the commitment:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\ \boxed{\clubsuit} \rightarrow \underbrace{\overset{\longleftrightarrow}{\boxed{?}\boxed{?}}}_{\overline{a}}\ \boxed{\clubsuit} \rightarrow \underbrace{\boxed{?}\boxed{?}\boxed{?}}_{E_3^{\heartsuit}(a)}.$$

2.2 Pile-Shifting Shuffle

We describe one of the most commonly used shuffling operations in card-based protocols: the *pile-shifting shuffle* [26, 37]. As an example, let us assume that there are nine cards divided into three piles.

(1) Divide a sequence of nine cards into piles of the same number of cards:

$$\overset{1\ 2\ 3}{\boxed{?}\boxed{?}\boxed{?}}\,\Big|\,\overset{4\ 5\ 6}{\boxed{?}\boxed{?}\boxed{?}}\,\Big|\,\overset{7\ 8\ 9}{\boxed{?}\boxed{?}\boxed{?}}.$$

(2) Cyclically shuffle the three piles without changing the order of the three cards within each pile. The resulting order of the piles will be one of the following three patterns, each with equal probability:

$$\left[\overset{1\ 2\ 3}{\boxed{?}\boxed{?}\boxed{?}}\,\Big|\,\overset{4\ 5\ 6}{\boxed{?}\boxed{?}\boxed{?}}\,\Big|\,\overset{7\ 8\ 9}{\boxed{?}\boxed{?}\boxed{?}}\right] \rightarrow \begin{cases} \overset{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}}, \\ \overset{7\ 8\ 9\ 1\ 2\ 3\ 4\ 5\ 6}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}}, \\ \overset{4\ 5\ 6\ 7\ 8\ 9\ 1\ 2\ 3}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}}. \end{cases}$$

We denote the application of a pile-shifting shuffle by $[\,\cdot \mid \cdots \mid \cdot\,]$.

2.3 Addition of Encoded Integers

Ruangwises and Itoh [30, 32] proposed the following method for adding two encoded integers. Our proposed protocol also employs this *integer addition* method.

(1) We have sequences of $E_k^{\clubsuit}(a)$ and $E_k^{\heartsuit}(b)$ representing two integers a and b, respectively. For convenience, each card is named as follows:

$$E_k^{\clubsuit}(a): \underset{x_0}{\boxed{?}}\,\underset{x_1}{\boxed{?}} \cdots \underset{x_{k-1}}{\boxed{?}}, \quad E_k^{\heartsuit}(b): \underset{y_0}{\boxed{?}}\,\underset{y_1}{\boxed{?}} \cdots \underset{y_{k-1}}{\boxed{?}}.$$

(2) Rearrange the sequences as follows:

$$\overset{y_{k-1}\ y_{k-2}\ \ y_0}{\underset{x_0\ x_1\ \ x_{k-1}}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}}}.$$

(3) Apply a pile-shifting shuffle as follows:

$$\left[\underset{x_0}{\overset{y_{k-1}}{\boxed{?}\boxed{?}}}\,\Big|\,\underset{x_1}{\overset{y_{k-2}}{\boxed{?}\boxed{?}}}\,\Big|\,\cdots\,\Big|\,\underset{x_{k-1}}{\overset{y_0}{\boxed{?}\boxed{?}}}\right] \rightarrow \underset{x_{0+r}\ \ x_{1+r}\ \ x_{k-1+r}}{\overset{y_{k-1-r}\ y_{k-2-r}\ \ y_{0-r}}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\cdots\boxed{?}\boxed{?}}},$$

where r is a random value.

(4) Rearrange these as they were before, as follows:

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

where a is subtracted by r and b is added by r.

(5) Reveal the sequence of $E_k^{\heartsuit}(b+r)$, and let $s = b + r$. Then, cyclically shift the sequence of $E_k^{\clubsuit}(a-r)$ to the right by s, i.e., s is added to $a - r$:

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

Note that when the sequence of $E_k^{\heartsuit}(b+r)$ is revealed, the value of b does not leak because the random value r was added to b.

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

In the current explanation, $E_k^{\clubsuit}(a)$ and $E_k^{\heartsuit}(b)$ are added; other types of pairs, such as a pair of $E_k^{\heartsuit}(a)$ and $E_k^{\clubsuit}(b)$ and a pair of $E_k^{\clubsuit}(a)$ and $E_k^{\clubsuit}(b)$, can be also added, of course.

2.4 Addition of Two Commitments

Shikata et al. [34] proposed the following *helping-card-free two-commitment addition* that produces $E_3^{\clubsuit}(a+b)$ or $E_3^{\heartsuit}(a+b)$ from two given commitments to $a, b \in \{0,1\}$ (without any helping card).

(1) Apply a pile-shifting shuffle to the commitments to a and b as follows:

$$\underbrace{?\,?}_{a}\ \underbrace{?\,?}_{b} \to [\,?\,?\,|\,?\,?\,] \to ?\,?\,?\,?.$$

(Such a shuffle is also called a *random bisection cut* [23].)

(2) Apply a pile-shifting shuffle to the middle two cards (in this case, it is a normal shuffle):

$$?\ [\,?\,|\,?\,]\ ?\ \to\ ?\,?\,?\,?.$$

(3) Reveal the second card from the left.

(a) If it is $\clubsuit$, we obtain a sequence of $E_3^{\clubsuit}(a+b)$ by rearranging as follows:

$$\overset{1}{?}\,\overset{2}{\clubsuit}\,\overset{3}{?}\,\overset{4}{?} \to \underbrace{\overset{1}{?}\,\overset{3}{?}\,\overset{4}{?}}_{E_3^{\clubsuit}(a+b)}\,\overset{2}{\clubsuit}.$$

(b) If it is $\heartsuit$, we obtain a sequence of $E_3^{\heartsuit}(a+b)$ by rearranging as follows:

$$\overset{1}{?}\,\overset{2}{\heartsuit}\,\overset{3}{?}\,\overset{4}{?} \to \underbrace{\overset{4}{?}\,\overset{3}{?}\,\overset{1}{?}}_{E_3^{\heartsuit}(a+b)}\,\overset{2}{\heartsuit}.$$

In this way, from commitments to a and b, we obtain a sequence of either $E_3^{\clubsuit}(a+b)$ or $E_3^{\heartsuit}(a+b)$ (with a probability of $1/2$) as well as one free card. This subprotocol will be also employed in our proposed protocols.

3 IDEA BEHIND OUR PROPOSED PROTOCOLS

In this section, we describe the idea behind our proposed protocols for $\{0\}$-partially doubly symmetric Boolean functions. We first give an overview of the two existing protocols of Ruangwises and Itoh [30, 32]: one is for symmetric Boolean functions and the other is for doubly symmetric Boolean functions. We then present the properties of $\{0\}$-partially doubly symmetric Boolean functions, and based on them, we show the strategy for reducing the number of helping cards.

3.1 Existing Protocols

Let $f : \{0,1\}^n \to R$ be a symmetric Boolean function, and let $g : \{0,1,\ldots,n\} \to R$ be the function satisfying Eq. (2). The existing protocol [30, 32] takes commitments to $x_1, x_2, \ldots, x_n \in \{0,1\}$ and two helping cards as input and uses the integer addition described in Sect. 2.3 repeatedly to obtain a sequence encoding $\sum_{i=1}^n x_i$, i.e., $E_{n+1}^{\clubsuit}(\sum_{i=1}^n x_i)$:

$$\underbrace{?\,?}_{x_1}\ \underbrace{?\,?}_{x_2}\ \cdots\ \underbrace{?\,?}_{x_n}\ \clubsuit\,\heartsuit \to \cdots \to \underbrace{?\,?\,?\,\cdots\,?}_{E_{n+1}^{\clubsuit}(\sum_{i=1}^n x_i)}.$$

Notice that if the $(i+1)$-st face-down card in $E_{n+1}^{\clubsuit}(\sum_{i=1}^n x_i)$ is $\clubsuit$, then $f(x_1, x_2, \ldots, x_n) = g(i)$. Therefore, for each $j \in R$, we collect all $(i+1)$-st cards such that $g(i) = j$, shuffle all the collected cards, and reveal them; if a $\clubsuit$ appears, then it implies $f(x_1, x_2, \ldots, x_n) = j$.

As an example, let f be the parity function, i.e., if i is even, $g(i) = 0$; otherwise, $g(i) = 1$. Then, after collecting all cards whose positions are odd numbers in $E_{n+1}^{\clubsuit}(\sum_{i=1}^n x_i)$, we shuffle all the collected cards and reveal them. If a $\clubsuit$ appears, then $f(x_1, x_2, \ldots, x_n) = 0$; otherwise, $f(x_1, x_2, \ldots, x_n) = 1$.

For another example, let f be the function h defined in Eq. (4) in Sect. 1.2. Then, we reveal the fifth and ninth cards to see if a $\clubsuit$ appears, i.e., if $h(x_1, x_2, \ldots, x_9) = 1$ and $h(x_1, x_2, \ldots, x_9) = 2$, respectively (here, we do not have to shuffle a single card).

In this way, from $E_{n+1}^{\clubsuit}(\sum_{i=1}^n x_i)$, we can know only the value of $f(x_1, x_2, \ldots, x_n)$, i.e., we can securely compute $f(x_1, x_2, \ldots, x_n)$.

For the case of doubly symmetric Boolean functions, Ruangwises and Itoh [30, 32] pointed out that two cards can be revealed by adding a common random bit to all input commitments, so that they become two free cards. The obtained two free cards can be used in the integer addition. Based on these ideas, they proposed a helping-card-free protocol for any doubly symmetric Boolean function.

3.2 Our Approach

As mentioned before, this paper focuses on $\{0\}$-partially doubly symmetric Boolean functions $f : \{0,1\}^n \to R$. In this case, the function g satisfying Eq. (2) has the property that $g(0) = g(n)$. Therefore, the function g also satisfies

$$g\left(\sum_{i=1}^n x_i\right) = g\left(\sum_{i=1}^n x_i \bmod n\right).$$

Because $0 \le (\sum_{i=1}^n x_i \bmod n) \le n-1$, this value is represented by n cards. This implies that instead of $E_{n+1}^{\clubsuit}(\sum_{i=1}^n x_i)$, it suffices to obtain $E_n^{\clubsuit}(\sum_{i=1}^n x_i \bmod n)$:

$$\underbrace{?\,?\,\cdots\,?}_{E_n^{\clubsuit}(\sum_{i=1}^n x_i \bmod n)},$$

That is, one less card can be used for encoding, which contributes to reducing the number of required helping cards, as will be seen in the next sections.

4 PROPOSED PROTOCOL FOR $n = 3$

In this section, we construct a helping-card-free protocol for an arbitrary $\{0\}$-partially doubly symmetric Boolean function for the case of $n = 3$.

Let $f : \{0,1\}^3 \to R$ be a $\{0\}$-partially doubly symmetric Boolean function to be securely computed. Since the proposed protocol requires no helping card, the input to the protocol is

$$\underbrace{?\,?}_{x_1}\ \underbrace{?\,?}_{x_2}\ \underbrace{?\,?}_{x_3}.$$

Our protocol proceeds as follows.

(1) For the commitments to x_1 and x_2, apply the helping-card-free two-commitment addition described in Sect. 2.4. Assume

Table 2: Value of $x_1 + x_2 + x_3$ and the sequence of cards

$x_1 + x_2 + x_3$	Sequence of cards
0	♣♡♡
1	♡♣♡
2	♡♡♣
3	♣♡♡

without loss of generality that $E_3^{\clubsuit}(x_1 + x_2)$ and a free black card ♣ are obtained:

$$\underbrace{??}_{x_1}\ \underbrace{??}_{x_2}\ \underbrace{??}_{x_3} \to \underbrace{???}_{E_3^{\clubsuit}(x_1+x_2)}\ \clubsuit\ \underbrace{??}_{x_3}.$$

(2) Convert the commitment to x_3 together with the free card ♣ to $E_3^{\heartsuit}(x_3)$ by the method mentioned in Sect. 2.1:

$$\underbrace{???}_{E_3^{\clubsuit}(x_1+x_2)}\ \underbrace{??}_{x_3}\ \clubsuit \to \underbrace{???}_{E_3^{\clubsuit}(x_1+x_2)}\ \underbrace{???}_{E_3^{\heartsuit}(x_3)}.$$

(3) Apply the integer addition described in Sect. 2.3 to $E_3^{\clubsuit}(x_1+x_2)$ and $E_3^{\heartsuit}(x_3)$:

$$\underbrace{???}_{E_3^{\clubsuit}(x_1+x_2)}\ \underbrace{???}_{E_3^{\heartsuit}(x_3)} \to \cdots \to \underbrace{???}_{E_3^{\clubsuit}(x_1+x_2+x_3 \bmod 3)}\ \clubsuit\clubsuit\heartsuit.$$

Here, as shown in Table 2, if $x_1 + x_2 + x_3 = 3$, the resulting three-card sequence becomes $E_3^{\clubsuit}(0)$. Therefore, we have $E_3^{\clubsuit}(x_1 + x_2 + x_3 \bmod 3)$.

This is how $E_3^{\clubsuit}(x_1+x_2+x_3 \bmod 3)$ is generated for $n = 3$, and as explained in Sect. 3, it is possible to securely compute $f(x_1, x_2, x_3)$ from this.

5 PROPOSED PROTOCOL FOR $n = 4$

In this section, we construct a helping-card-free protocol for an arbitrary $\{0\}$-partially doubly symmetric Boolean function for the case of $n = 4$.

Let $f : \{0,1\}^4 \to R$ be a $\{0\}$-partially doubly symmetric Boolean function to be securely computed. Since the proposed protocol requires no helping card, the input to the protocol is

$$\underbrace{??}_{x_1}\ \underbrace{??}_{x_2}\ \underbrace{??}_{x_3}\ \underbrace{??}_{x_4}.$$

We first present a subprotocol in Sect. 5.1 and then show the main protocol in Sect. 5.2.

5.1 Color Conversion Subprotocol

In our protocol presented in this section, we have a probabilistic case where an integer encoding based on some color must be converted to the other color. For this, we propose a subprotocol to convert a three-card encoding of either color obtained by the helping-card-free two-commitment addition described in Sect. 2.3 to the other color with one helping card.

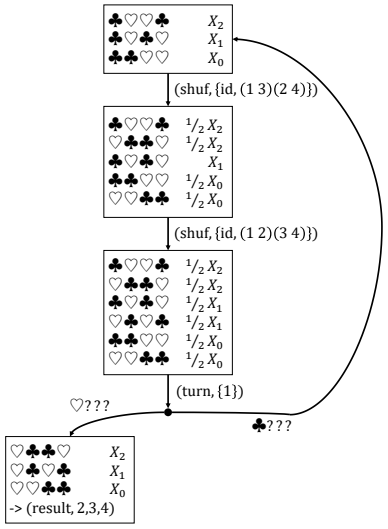


Figure 1: KWH-tree for the color conversion subprotocol

Here, we explain the case of converting $E_3^{\clubsuit}(x)$ to $E_3^{\heartsuit}(x)$.

(1) Place a helping card and turn it over as follows:

$$\clubsuit\ \underbrace{???}_{E_3^{\clubsuit}(x)} \to ????.$$

(2) Apply a random bisection cut as follows:

$$[??|??] \to ????.$$

(3) After rearranging the middle two cards, apply a random bisection cut, and then rearrange the middle two cards again:

$$\overleftrightarrow{????} \to [??|??] \to \overleftrightarrow{????}.$$

(4) Reveal the left-most card:

$$\underset{\text{turn}}{?}???.$$

(a) If it is ♡, then the remaining face-down cards represent $E_3^{\heartsuit}(x)$:

$$\heartsuit\ \underbrace{???}_{E_3^{\heartsuit}(x)}.$$

(b) If it is ♣, then turn over the revealed card again, and return to Step 2.

This is the color conversion subprotocol, converting $E_3^{\clubsuit}(x)$ to $E_3^{\heartsuit}(x)$. If we want to convert $E_3^{\heartsuit}(x)$ to $E_3^{\clubsuit}(x)$, then it suffices to just reverse ♡ and ♣ and perform the same operations.

The security and correctness of this subprotocol is proved by its KWH-tree [12] depicted in Fig. 1. Note that this subprotocol has a loop and is a Las Vegas algorithm.

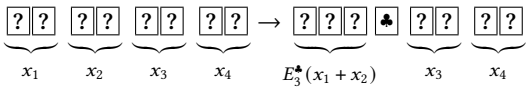
5.2 Protocol Description

The proposed protocol for $n = 4$ proceeds as follows.

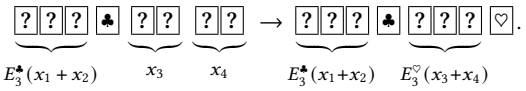
Table 3: Value of $x_1 + x_2 + x_3 + x_4$ and the sequence of cards

$x_1 + x_2 + x_3 + x_4$	Sequence of cards
0	♣ ♡ ♡ ♡
1	♡ ♣ ♡ ♡
2	♡ ♡ ♣ ♡
3	♡ ♡ ♡ ♣
4	♣ ♡ ♡ ♡

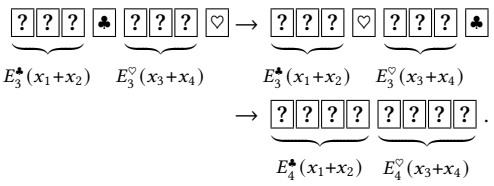
(1) For the commitments to x_1 and x_2, apply the helping-card-free two-commitment addition described in Sect. 2.4. Assume without loss of generality that we obtain $E_3^{\clubsuit}(x_1 + x_2)$ and a free black card ♣:



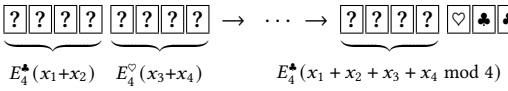
(2) For the commitments to x_3 and x_4, apply the helping-card-free two-commitment addition described in Sect. 2.4. Here, we want to obtain the addition of x_3+x_4 with a different color encoding to the addition of $x_1 + x_2$ obtained in the previous step. Therefore, after the addition of x_3 and x_4, we proceed to the next step if $E_3^{\heartsuit}(x_3 + x_4)$ is obtained. If $E_3^{\clubsuit}(x_3 + x_4)$ is obtained, then apply the color conversion subprotocol described in Sect. 5.1 to it. In any case, the resulting sequence after the operation is as follows:



(3) Convert $E_3^{\clubsuit}(x_1 + x_2)$ and $E_3^{\heartsuit}(x_3 + x_4)$ together with the free cards ♡ and ♣ to $E_4^{\clubsuit}(x_1 + x_2)$ and $E_4^{\heartsuit}(x_3 + x_4)$, respectively, in a similar way to the method mentioned in Sect. 2.1:



(4) Apply the integer addition described in Sect. 2.3 to $E_4^{\clubsuit}(x_1+x_2)$ and $E_4^{\heartsuit}(x_3 + x_4)$:

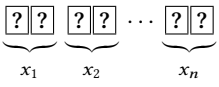


Here, as shown in Table 3, if $x_1 + x_2 + x_3 + x_4 = 4$, the resulting four-card sequence becomes $E_4^{\clubsuit}(0)$. Therefore, we have $E_4^{\clubsuit}(x_1 + x_2 + x_3 + x_4 \bmod 4)$.

This is how $E_4^{\clubsuit}(x_1 + x_2 + x_3 + x_4 \bmod 4)$ is generated for $n = 4$, and as explained in Sect. 3, it is possible to securely compute $f(x_1, x_2, x_3, x_4)$ from this.

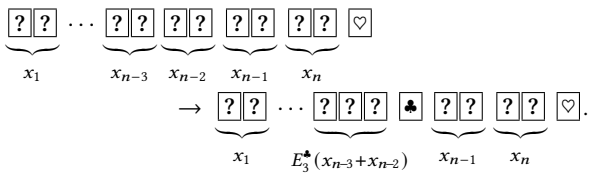
6 PROPOSED PROTOCOL FOR $n \geq 5$

In this section, we construct a protocol for an arbitrary n-input $\{0\}$-partially doubly symmetric Boolean function such that $n \geq 5$. The proposed protocol requires one helping card ♡, and hence, the input to the protocol is

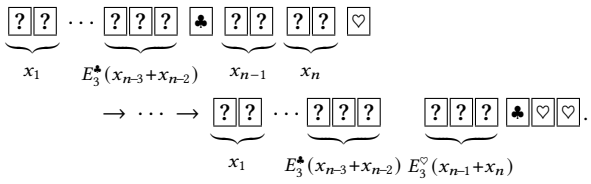


The protocol proceeds as follows.

(1) For the commitments to x_{n-3} and x_{n-2}, apply the helping-card-free two-commitment addition described in Sect. 2.4. Assume without loss of generality that $E_3^{\clubsuit}(x_{n-3} + x_{n-2})$ and a free black card ♣ are obtained:

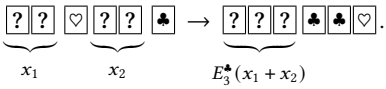


(2) To the commitments to x_{n-1} and x_n together with the free cards ♣ and ♡, apply the integer addition described in Sect. 2.3:

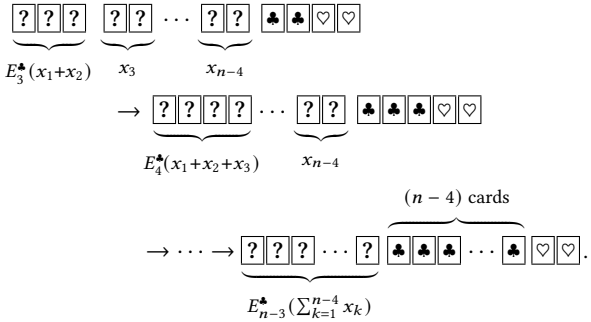


Note that we apply the addition so that $E_3^{\heartsuit}(x_{n-1} + x_n)$ is obtained.

(3) Convert the commitments to x_1 and x_2 together with ♡ and ♣ to $E_3^{\clubsuit}(x_1)$ and $E_3^{\heartsuit}(x_2)$, and apply the integer addition described in Sect. 2.3, obtaining $E_3^{\clubsuit}(x_1 + x_2)$:

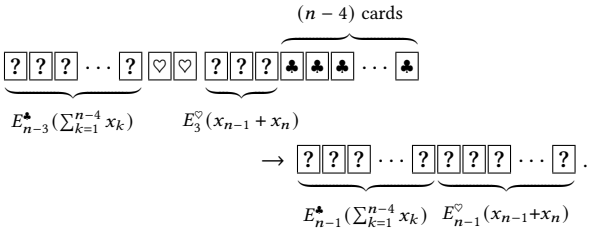


(4) Similar to the previous step, add the commitments to x_3, x_4, …, x_{n-4} to $E_3^{\clubsuit}(x_1+x_2)$ one by one, obtaining $E_{n-3}^{\clubsuit}(\sum_{k=1}^{n-4} x_k)$:

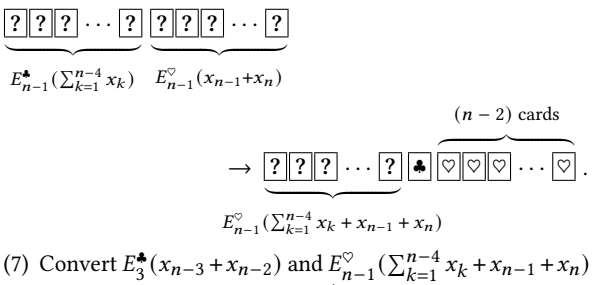


(5) Convert $E_{n-3}^{\clubsuit}(\sum_{k=1}^{n-4} x_k)$ and $E_3^{\heartsuit}(x_{n-1}+x_n)$ to $E_{n-1}^{\clubsuit}(\sum_{k=1}^{n-4} x_k)$ and $E_{n-1}^{\heartsuit}(x_{n-1} + x_n)$, respectively, in a similar way to the

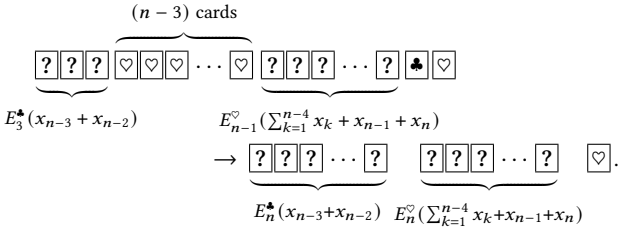
method described in Sect. 2.1:



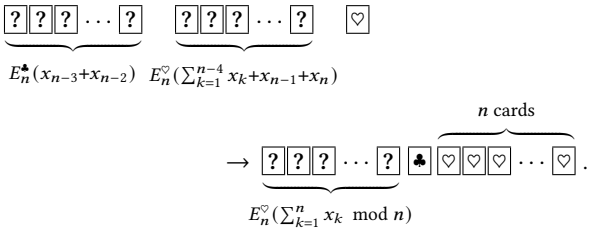
(6) Apply the integer addition presented in Sect. 2.3 to $E^{\clubsuit}_{n-1}(\sum_{k=1}^{n-4} x_k)$ and $E^{\heartsuit}_{n-1}(x_{n-1}+x_n)$ derived in the previous step:



(7) Convert $E^{\clubsuit}_{3}(x_{n-3}+x_{n-2})$ and $E^{\heartsuit}_{n-1}(\sum_{k=1}^{n-4} x_k + x_{n-1} + x_n)$ to $E^{\clubsuit}_{n}(x_{n-3}+x_{n-2})$ and $E^{\heartsuit}_{n}(\sum_{k=1}^{n-4} x_k + x_{n-1} + x_n)$, respectively, in a similar way to the method described in Sect. 2.1:



(8) Apply the integer addition described in Sect. 2.3 to $E^{\clubsuit}_{n}(x_{n-3}+x_{n-2})$ and $E^{\heartsuit}_{n}(\sum_{k=1}^{n-4} x_k + x_{n-1} + x_n)$:



Here, as shown in Table 4, if $\sum_{k=1}^{n} x_k = n$, the resulting sequence becomes $E^{\heartsuit}_{n}(0)$. Therefore, we have $E^{\heartsuit}_{n}(\sum_{k=1}^{n} x_k \mod n)$.

This is how $E^{\heartsuit}_{n}(\sum_{k=1}^{n} x_k \mod n)$ is obtained for $n \geq 5$, and as explained in Sect. 3, it is possible to compute any $\{0\}$-partially doubly symmetric Boolean function from this sequence.

7 CONCLUSION

In this paper, we aimed to construct new card-based protocols requiring fewer helping cards within the class of symmetric Boolean functions. To this end, we considered the class of $\{0\}$-partially doubly symmetric Boolean functions as a subclass, and designed generic protocols for them using less than two helping cards. Specifically, our results show that no helping card is required for $n = 3, 4$, and only one helping card suffices for $n \geq 5$.

As for the case of $n = 2$, given commitments to x_1, x_2, the helping-card-free two-commitment addition described in Sect. 2.4 produces $E^{\clubsuit}_{3}(x_1 + x_2)$ or $E^{\heartsuit}_{3}(x_1 + x_2)$, from which we can securely compute any two-input symmetric Boolean function having an arbitrary range. Therefore, for the case of $n = 2$, without the restriction, there is a helping-card-free protocol for any symmetric Boolean function; see Table 1 again.

For future work, we will address whether there exists a protocol with less than two helping cards for any n-input (unrestricted) symmetric Boolean function having an arbitrary range such that $n \geq 3$.

Table 4: Value of $\sum_{k=1}^{n} x_k$ and the sequence of cards

Value of $\sum_{k=1}^{n} x_k$	Sequence of cards
0	♡ ♣ ♣ ··· ♣ ♣
1	♣ ♡ ♣ ··· ♣ ♣
2	♣ ♣ ♡ ··· ♣ ♣
⋮	⋮
$n-1$	♣ ♣ ♣ ··· ♣ ♡
n	♡ ♣ ♣ ··· ♣ ♣

ACKNOWLEDGMENTS

We thank the anonymous referees, whose comments have helped us to improve the presentation of the paper. The method in Sect. 5.1 was established with the help of Kodai Toyoda. This work was supported in part by JSPS KAKENHI Grant Numbers JP21K11881 and JP23H00479.

REFERENCES

[1] Yoshiki Abe, Takeshi Nakai, Yoshihisa Kuroki, Shinnosuke Suzuki, Yuta Koga, Yohei Watanabe, Mitsugu Iwamoto, and Kazuo Ohta. 2022. Efficient Card-Based Majority Voting Protocols. *New Gener. Comput.* 40 (2022), 173–198. https://doi.org/10.1007/s00354-022-00161-7

[2] Claude Crépeau and Joe Kilian. 1994. Discreet Solitary Games. In *Advances in Cryptology—CRYPTO' 93 (LNCS, Vol. 773)*, Douglas R. Stinson (Ed.). Springer, Berlin, Heidelberg, 319–330. https://doi.org/10.1007/3-540-48329-2_27

[3] Takuro Fukasawa and Yoshifumi Manabe. 2022. Card-Based Zero-Knowledge Proof for the Nearest Neighbor Property: Zero-Knowledge Proof of ABC End View. In *Security, Privacy, and Applied Cryptography Engineering (Lecture Notes in Computer Science, Vol. 13783)*, Lejla Batina, Stjepan Picek, and Mainack Mondal (Eds.). Springer Nature Switzerland, Cham, 147–161.

[4] Rikuo Haga, Yuichi Hayashi, Daiki Miyahara, and Takaaki Mizuki. 2022. Card-Minimal Protocols for Three-Input Functions with Standard Playing Cards. In *Progress in Cryptology—AFRICACRYPT 2022 (LNCS)*. Springer, Cham. to appear.

[5] Rikuo Haga, Kodai Toyoda, Yuto Shinoda, Daiki Miyahara, Kazumasa Shinagawa, Yuichi Hayashi, and Takaaki Mizuki. 2022. Card-Based Secure Sorting Protocol. In *Advances in Information and Computer Security (LNCS, Vol. 13504)*, Chen-Mou Cheng and Mitsuaki Akiyama (Eds.). Springer, Cham, 224–240. https://doi.org/10.1007/978-3-031-15255-9_12

[6] James Heather, Steve Schneider, and Vanessa Teague. 2014. Cryptographic protocols with everyday objects. *Formal Aspects of Computing* 26, 1 (2014), 37–62. https://doi.org/10.1007/s00165-013-0274-7

[7] Raimu Isuzugawa, Kodai Toyoda, Yu Sasaki, Daiki Miyahara, and Takaaki Mizuki. 2021. A Card-Minimal Three-Input AND Protocol Using Two Shuffles. In *Computing and Combinatorics (LNCS, Vol. 13025)*, Chi-Yeh Chen, Wing-Kai Hon, Ling-Ju Hung, and Chia-Wei Lee (Eds.). Springer, Cham, 668–679. https://doi.org/10.1007/978-3-030-89543-3_55

[8] Julia Kastner, Alexander Koch, Stefan Walzer, Daiki Miyahara, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. 2017. The Minimum Number of Cards in Practical Card-Based Protocols. In *Advances in Cryptology—ASIACRYPT 2017 (LNCS, Vol. 10626)*, Tsuyoshi Takagi and Thomas Peyrin (Eds.). Springer, Cham, 126–155. https://doi.org/10.1007/978-3-319-70700-6_5

[9] Alexander Koch. 2019. *Cryptographic Protocols from Physical Assumptions.* Ph. D. Dissertation. Karlsruhe Institute of Technology. https://doi.org/10.5445/IR/1000097756

[10] Alexander Koch. 2021. The Landscape of Security from Physical Assumptions. In *IEEE Information Theory Workshop*. IEEE, NY, 1–6. https://doi.org/10.1109/ITW48936.2021.9611501

[11] Alexander Koch, Michael Schrempp, and Michael Kirsten. 2019. Card-Based Cryptography Meets Formal Verification. In *Advances in Cryptology–ASIACRYPT 2019 (LNCS, Vol. 11921)*, Steven D. Galbraith and Shiho Moriai (Eds.). Springer, Cham, 488–517. https://doi.org/10.1007/978-3-030-34578-5_18

[12] Alexander Koch, Stefan Walzer, and Kevin Härtel. 2015. Card-Based Cryptographic Protocols Using a Minimal Number of Cards. In *Advances in Cryptology—ASIACRYPT 2015 (LNCS, Vol. 9452)*, Tetsu Iwata and Jung Hee Cheon (Eds.). Springer, Berlin, Heidelberg, 783–807. https://doi.org/10.1007/978-3-662-48797-6_32

[13] Yoshifumi Manabe and Hibiki Ono. 2022. Card-Based Cryptographic Protocols with Malicious Players Using Private Operations. *New Gener. Comput.* 40 (2022), 67–93. https://doi.org/10.1007/s00354-021-00148-w

[14] Daiki Miyahara, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. 2020. Practical card-based implementations of Yao's millionaire protocol. *Theor. Comput. Sci.* 803 (2020), 207–221. https://doi.org/10.1016/j.tcs.2019.11.005

[15] Daiki Miyahara and Takaaki Mizuki. 2022. Secure Computations through Checking Suits of Playing Cards. In *Frontiers in Algorithmics (Lecture Notes in Computer Science, Vol. 13461)*. Springer, Cham, 110–128.

[16] Kengo Miyamoto and Kazumasa Shinagawa. 2022. Graph Automorphism Shuffles from Pile-Scramble Shuffles. *New Gener. Comput.* 40 (2022), 199–223. https://doi.org/10.1007/s00354-022-00164-4

[17] Takaaki Mizuki. 2016. Card-based Protocols for Securely Computing the Conjunction of Multiple Variables. *Theor. Comput. Sci.* 622, C (2016), 34–44. https://doi.org/10.1016/j.tcs.2016.01.039

[18] Takaaki Mizuki. 2021. Preface: Special Issue on Card-Based Cryptography. *New Gener. Comput.* 39 (2021), 1–2. https://doi.org/10.1007/s00354-021-00127-1

[19] Takaaki Mizuki. 2022. Preface: Special Issue on Card-Based Cryptography 2. *New Gener. Comput.* 40 (2022), 47–48. https://doi.org/10.1007/s00354-022-00170-6

[20] Takaaki Mizuki, Michihito Kumamoto, and Hideaki Sone. 2012. The Five-Card Trick Can Be Done with Four Cards. In *Advances in Cryptology—ASIACRYPT 2012 (LNCS, Vol. 7658)*, Xiaoyun Wang and Kazue Sako (Eds.). Springer, Berlin, Heidelberg, 598–606. https://doi.org/10.1007/978-3-642-34961-4_36

[21] Takaaki Mizuki and Hiroki Shizuya. 2014. A formalization of card-based cryptographic protocols via abstract machine. *Int. J. Inf. Secur.* 13, 1 (2014), 15–23. https://doi.org/10.1007/s10207-013-0219-4

[22] Takaaki Mizuki and Hiroki Shizuya. 2017. Computational Model of Card-Based Cryptographic Protocols and Its Applications. *IEICE Trans. Fundam.* E100.A, 1 (2017), 3–11. https://doi.org/10.1587/transfun.E100.A.3

[23] Takaaki Mizuki and Hideaki Sone. 2009. Six-Card Secure AND and Four-Card Secure XOR. In *Frontiers in Algorithmics (LNCS, Vol. 5598)*, Xiaotie Deng, John E. Hopcroft, and Jinyun Xue (Eds.). Springer, Berlin, Heidelberg, 358–369. https://doi.org/10.1007/978-3-642-02270-8_36

[24] Takeshi Nakai, Yuto Misawa, Yuuki Tokushige, Mitsugu Iwamoto, and Kazuo Ohta. 2022. Secure Computation for Threshold Functions with Physical Cards: Power of Private Permutations. *New Gener. Comput.* 40 (2022), 95–113. https://doi.org/10.1007/s00354-022-00153-7

[25] Takuya Nishida, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. 2015. Card-Based Protocols for Any Boolean Function. In *Theory and Applications of Models of Computation (LNCS, Vol. 9076)*, Rahul Jain, Sanjay Jain, and Frank Stephan (Eds.). Springer, Cham, 110–121. https://doi.org/10.1007/978-3-319-17142-5_11

[26] Akihiro Nishimura, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. 2018. Pile-shifting scramble for card-based protocols. *IEICE Trans. Fundam.* 101, 9 (2018), 1494–1502. https://doi.org/10.1587/transfun.E101.A.1494

[27] Léo Robert, Daiki Miyahara, Pascal Lafourcade, Luc Libralesso, and Takaaki Mizuki. 2022. Physical zero-knowledge proof and NP-completeness proof of Suguru puzzle. *Inf. Comput.* 285 (2022), 1–14. https://doi.org/10.1016/j.ic.2021.104858

[28] Léo Robert, Daiki Miyahara, Pascal Lafourcade, and Takaaki Mizuki. 2022. Card-Based ZKP for Connectivity: Applications to Nurikabe, Hitori, and Heyawake. *New Gener. Comput.* 40 (2022), 149–171. https://doi.org/10.1007/s00354-022-00155-5

[29] Léo Robert, Daiki Miyahara, Pascal Lafourcade, and Takaaki Mizuki. 2022. Hide a Liar: Card-Based ZKP Protocol for Usowan. In *Theory and Applications of Models of Computation (LNCS)*. Springer, Cham. to appear.

[30] Suthee Ruangwises and Toshiya Itoh. 2020. Securely Computing the *n*-Variable Equality Function with 2*n* Cards. In *Theory and Applications of Models of Computation (LNCS, Vol. 12337)*, Jianer Chen, Qilong Feng, and Jinhui Xu (Eds.). Springer, Cham, 25–36. https://doi.org/10.1007/978-3-030-59267-7_3

[31] Suthee Ruangwises and Toshiya Itoh. 2021. Physical ZKP for Connected Spanning Subgraph: Applications to Bridges Puzzle and Other Problems. In *Unconventional Computation and Natural Computation*, Irina Kostitsyna and Pekka Orponen (Eds.). Springer, Cham, 149–163.

[32] Suthee Ruangwises and Toshiya Itoh. 2021. Securely computing the n-variable equality function with 2n cards. *Theor. Comput. Sci.* 887 (2021), 99–110. https://doi.org/10.1016/j.tcs.2021.07.007

[33] Suthee Ruangwises and Toshiya Itoh. 2022. Physical ZKP for Makaro Using a Standard Deck of Cards. In *Theory and Applications of Models of Computation (LNCS)*. Springer, Cham. to appear.

[34] Hayato Shikata, Kodai Toyoda, Daiki Miyahara, and Takaaki Mizuki. 2022. Card-minimal Protocols for Symmetric Boolean Functions of More Than Seven Inputs. In *Theoretical Aspects of Computing – ICTAC 2022 (LNCS, Vol. 13572)*, Helmut Seidl, Zhiming Liu, and Corina S. Pasareanu (Eds.). Springer, Cham, 388–406. https://doi.org/10.1007/978-3-031-17715-6_25

[35] Kazumasa Shinagawa. 2020. *On the Construction of Easy to Perform Card-Based Protocols.* Ph. D. Dissertation. Tokyo Institute of Technology.

[36] Kazumasa Shinagawa and Takaaki Mizuki. 2019. The Six-Card Trick: Secure Computation of Three-Input Equality. In *Information Security and Cryptology (LNCS, Vol. 11396)*, Kwangsu Lee (Ed.). Springer, Cham, 123–131. https://doi.org/10.1007/978-3-030-12146-4_8

[37] Kazumasa Shinagawa, Takaaki Mizuki, Jacob Schuldt, Koji Nuida, Naoki Kanayama, Takashi Nishide, Goichiro Hanaoka, and Eiji Okamoto. 2017. Card-Based Protocols Using Regular Polygon Cards. *IEICE Trans. Fundam.* E100.A, 9 (2017), 1900–1909. https://doi.org/10.1587/transfun.E100.A.1900

[38] Kazumasa Shinagawa and Koji Nuida. 2022. Single-shuffle Full-open Card-based Protocols Imply Private Simultaneous Messages Protocols. Cryptology ePrint Archive, Paper 2022/1306. https://eprint.iacr.org/2022/1306 https://eprint.iacr.org/2022/1306.

[39] Yuji Suga. 2022. A classification proof for commutative three-element semigroups with local AND structure and its application to card-based protocols. In *2022 IEEE International Conference on Consumer Electronics - Taiwan*. IEEE, NY, 171–172. https://doi.org/10.1109/ICCE-Taiwan55306.2022.9869063

[40] Ken Takashima, Yuta Abe, Tatsuya Sasaki, Daiki Miyahara, Kazumasa Shinagawa, Takaaki Mizuki, and Hideaki Sone. 2020. Card-based protocols for secure ranking computations. *Theor. Comput. Sci.* 845 (2020), 122–135. https://doi.org/10.1016/j.tcs.2020.09.008

[41] Kodai Toyoda, Daiki Miyahara, and Takaaki Mizuki. 2021. Another Use of the Five-Card Trick: Card-Minimal Secure Three-Input Majority Function Evaluation. In *Progress in Cryptology—INDOCRYPT 2021 (LNCS, Vol. 13143)*, Avishek Adhikari, Ralf Küsters, and Bart Preneel (Eds.). Springer, Cham, 536–555. https://doi.org/10.1007/978-3-030-92518-5_24