New Generation Computing (2021) 39:41–71
https://doi.org/10.1007/s00354-020-00117-9

NEW GENERATION COMPUTING

Card-based Cryptography with Dihedral Symmetry

Kazumasa Shinagawa[1,2]

Received: 6 May 2020 / Accepted: 9 October 2020 / Published online: 4 January 2021

Abstract
It is known that secure computation can be done by using a deck of physical cards. This area is called card-based cryptography. Shinagawa et al. (in: Provable security—9th international conference, ProvSec 2015, Kanazawa, Japan, 2015) proposed regular n-sided polygon cards that enable to compute functions over $\mathbb{Z}/n\mathbb{Z}$. In particular, they designed efficient protocols for linear functions (e.g. addition and constant multiplication) over $\mathbb{Z}/n\mathbb{Z}$. Here, efficiency is measured by the number of cards used in the protocol. In this paper, we propose a new type of cards, dihedral cards, as a natural generalization of regular polygon cards. Based on them, we construct efficient protocols for various interesting functions such as carry of addition, equality, and greater-than, whose efficient construction has not been known before. Beside this, we introduce a new protocol framework that captures a wide class of card types including binary cards, regular polygon cards, dihedral cards, and so on.

Keywords Secure computation · Card-based cryptography · Invisible ink

Introduction

Secure computation enables a set of parties each having inputs to jointly compute a predetermined function of their inputs without revealing their inputs beyond the output. Card-based cryptography (ex. [2, 4, 9]) is secure computation that can be done by using a deck of physical cards, instead of computer devices. This makes people understand the correctness and security of secure computation, even for people who are not familiar with mathematics. Indeed, it is applied to educational situations; some universities

A preliminary conference version appeared at [13]. The main additions from the conference version are Sect. 2 (formal protocol definition) and all security proofs in Sect. 4. This article is a part of my PhD dissertation.

✉ Kazumasa Shinagawa
shinagawakazumasa@uec.ac.jp

[1] The University of Electro-Communications, 1-5-1, Chofugaoka, Chofu, Tokyo 182-8585, Japan

[2] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo Waterfront Bio-IT Research Building 2-4-7 Aomi, Koto City, Tokyo 135-0064, Japan

(e.g., Cornell University [7], University of Waterloo [3], and Tohoku University [8]) adopt card-based cryptography as a teaching material for beginner students.

While most of all existing works [1, 3–6, 9–12, 16] are mainly focused on binary computation only, a lot of secure computation that arises in everyday and classroom situations needs to take multi-valued inputs. For instance, secure computation of the average score, which takes a number of scores and outputs the average of them, is such a canonical example. In order to compute multi-valued functions efficiently, Shinagawa et al. [15] proposed a deck of regular polygon cards, whose shape is a regular n-sided polygon for the base number n. They proposed a two-card addition protocol that outputs $x + y \bmod n$ given two cards having $x, y \in \mathbb{Z}/n\mathbb{Z}$.

Does a deck of regular polygon cards realize sufficiently efficient secure computation for multi-valued functions? Up until now, there exist efficient protocols only for a very restrictive class of functions such as addition and subtraction, however, it requires a large number of cards for computing a function in the outside of the class (in general, it requires $O(n^k)$ cards for k inputs). Unfortunately, there are no efficient protocols even for very simple functions such as addition with carry, where given two integers $x, y \in \{0, 1, \ldots, n-1\}$, it outputs a carry of addition, the predicate "$x + y \geq n$". To compute a carry of addition efficiently is one of the open problems in this area. In this paper, we solve it by designing a new type of cards.

Our Contribution

Dihedral cards We design a new type of cards, dihedral cards, which is based on the use of invisible ink. It enables to construct several efficient protocols. Introducing invisible ink in the area of card-based cryptography is also our contribution. We construct an efficient protocol for computing interesting predicates: a carry of addition "$x + y \geq n$", equality with zero "$x = 0$", equality "$x = y$", and greater than "$x \geq y$". Table 1 shows a comparison between our protocols and the previous protocols [15] with regular polygon cards (RPC). Somewhat surprisingly, our protocols with dihedral cards (DC) for these predicates requires only two cards while all existing RPC-based protocols for the same predicates requires a large number of cards depending on the modulus n.

A unified protocol model We introduce a new protocol model for describing protocols with our new cards (Sect. 2). Our model has somewhat generality. It captures a wide class of protocols not only our dihedral cards but also other type of cards. For example, our model also captures regular polygon cards [14, 15]. See Appendix for the definition of regular polygon cards in our model. We believe that our model will be applied to future works proposing new cards. We left to give concrete definitions for other cards as future works.

A Unified Protocol Model

In this section, we introduce a protocol model for describing not only our dihedral cards but also other cards. Roughly speaking, a card-based protocol can be specified by a deck of cards and a set of operations. Thus in order to describe a

Table 1 Comparison between our protocols and previous protocols: "RPC", and "DC" denote regular polygon cards and dihedral cards, respectively

	Type of cards	Number of cards	Number of shuffles
∘ Addition and Subtraction			
Shinagawa et al. [15]	RPC	2	1
Ours	DC	2	1
∘ Carry: the predicate "$x + y \geq n$"			
Shinagawa et al. [15]	RPC	$n^2 + n + 2$	2
Ours	DC	2	5
∘ Equality with zero: the predicate "$x = 0$"			
Shinagawa et al. [15]	RPC	$2n + 1$	1
Ours	DC	2	4
∘ Equality: the predicate "$x = y$"			
Shinagawa et al. [15]	RPC	$n^2 + n + 2$	2
Ours	DC	2	6
∘ Greater than: the predicate "$x \geq y$"			
Shinagawa et al. [15]	RPC	$n^2 + n + 2$	2
Ours	DC	2	5

new type of cards, we must define a suitable deck of cards and a suitable set of operations. In this section, we explain the model with the case of the standard binary cards ♣♡ in order to make it easier to read for those who are familiar with the ordinary card-based cryptography. We give definitions for dihedral cards in Sect. 3. We also give definitions for other cards in Appendix.

Deck, Sequence, and Visible Sequence

In Mizuki-Shizuya model, a deck is defined by a finite multiset. For example, $\mathcal{D} = \{\clubsuit, \clubsuit, \clubsuit, \heartsuit, \heartsuit, \heartsuit\}$ denotes a deck consists of six cards: three clubs and three hearts. All backsides are assumed to be "**?**". (Thus, it is required the condition that $\mathcal{D} \cap \{?\} = \emptyset$.) Although it captures some class of decks including decks of binary cards ♣♡ and number cards 1 2 3, it is not sufficient if non-standard cards (like dihedral cards) are used.

In our model, we define a deck as follows:

Definition 1 (Deck) A deck $\overline{\mathcal{D}}$ is defined by a five-tuple as follows:

$$\overline{\mathcal{D}} := (\mathcal{C}, \mathcal{T}, \Sigma, \mathsf{vis}, \mathcal{D}),$$

where $\mathcal{C}$ is a finite set called a card set, $\mathcal{T} \subset \{t \mid t : \mathcal{C} \to \mathcal{C}\}$ is called a transformation set, Σ is a finite set called a symbol set, $\mathsf{vis} : \mathcal{C} \to \Sigma$ is a function called a vision function, and $\mathcal{D}$ is a finite multiset called a deck set, where the base set is $\mathcal{C}$. We

assume that $\mathcal{T}$ always contains the identity function $\mathsf{id} : \mathcal{C} \to \mathcal{C}$. The former four-tuple $(\mathcal{C}, \mathcal{T}, \Sigma, \mathsf{vis})$ is called a card specification. ■

Example 1 Consider a deck of cards 🂡🂱🂻🃁🃋 whose back sides are 🂠, which is used by the Five-Card Trick [2]. The deck is described by the following:

- The card set is $\mathcal{C} = \{\clubsuit/\mathbf{?}, \heartsuit/\mathbf{?}, \mathbf{?}/\clubsuit, \mathbf{?}/\heartsuit\}$;
- The symbol set is $\Sigma = \{\clubsuit, \heartsuit, \mathbf{?}\}$;
- The transformation set is $\mathcal{T} = \{\mathsf{id}, \mathsf{turn}\}$, where the function turn is defined by $\mathsf{turn}(X/Y) = Y/X$ for any $X, Y \in \Sigma$;
- The vision function vis is defined by $\mathsf{vis}(X/Y) = X$ for any $X, Y \in \Sigma$;
- The deck set is $\mathcal{D} = \{\clubsuit/\mathbf{?}, \clubsuit/\mathbf{?}, \heartsuit/\mathbf{?}, \heartsuit/\mathbf{?}, \heartsuit/\mathbf{?}\} = \{(\clubsuit/\mathbf{?})^2, (\heartsuit/\mathbf{?})^3\}$.

For the card set $\mathcal{C}$, the element "$\clubsuit/\mathbf{?}$" (resp. "$\heartsuit/\mathbf{?}$") means a face-up card 🂡 (resp. 🂱) and the element "$\mathbf{?}/\clubsuit$" (resp. "$\mathbf{?}/\heartsuit$") means a face-down card 🂠 whose front side is 🂻 (resp. 🃁). The transformation set has a turning transformation turn. By applying turn to a card, a face-up card is changed to a face-down card (and vice versa). The vision function specifies what information is revealed from a card. From face-up cards "$\clubsuit/\mathbf{?}$" and "$\heartsuit/\mathbf{?}$", it reveals the symbols "$\clubsuit$" and "$\heartsuit$", on the other hand, from face-down cards "$\mathbf{?}/\clubsuit$" and "$\mathbf{?}/\heartsuit$", it reveals "$\mathbf{?}$" only. This card specification $(\mathcal{C}, \mathcal{T}, \Sigma, \mathsf{vis})$ is called the binary cards. Hereafter, we denote the binary cards by $\mathsf{Binary} = (\mathcal{C}^{\mathsf{b}}, \mathcal{T}^{\mathsf{b}}, \Sigma^{\mathsf{b}}, \mathsf{vis}^{\mathsf{b}})$. ■

Sequence We define a sequence as follows:

Definition 2 (Sequence) Let $\overline{\mathcal{D}} = (\mathcal{C}, \mathcal{T}, \Sigma, \mathsf{vis}, \mathcal{D})$ be a deck. A sequence s in $\overline{\mathcal{D}}$ is defined as follows:

$$s = (t_1(x_1), t_2(x_2), \ldots, t_{|\mathcal{D}|}(x_{|\mathcal{D}|})),$$

where $t_1, t_2, \ldots, t_{|\mathcal{D}|} \in \mathcal{T}$ and $\mathcal{D} = \{x_1, x_2, \ldots, x_{|\mathcal{D}|}\}$ as a multiset. The set of all sequences in $\overline{\mathcal{D}}$ is denoted by $\mathsf{Seq}^{\overline{\mathcal{D}}}$. ■

Example 2 Let $\overline{\mathcal{D}} = (\mathsf{Binary}, \mathcal{D})$ be the deck in Example 1. An example of a sequence s of $\overline{\mathcal{D}}$ is as follows:

$$s = (\mathbf{?}/\clubsuit, \mathbf{?}/\heartsuit, \heartsuit/\mathbf{?}, \mathbf{?}/\heartsuit, \mathbf{?}/\clubsuit).$$

This is because s is represented as follows:

$$s = (\mathsf{turn}(\clubsuit/\mathbf{?}), \mathsf{turn}(\heartsuit/\mathbf{?}), \mathsf{id}(\heartsuit/\mathbf{?}), \mathsf{turn}(\heartsuit/\mathbf{?}), \mathsf{turn}(\clubsuit/\mathbf{?})).$$

It represents a sequence 🂠🂠🂱🂠🂠. ■

Visible sequence We define a visible sequence as follows:

Definition 3 (Visible sequence) Let $\overline{\mathcal{D}} = (\mathcal{C}, \mathcal{T}, \Sigma, \mathsf{vis}, \mathcal{D})$ be a deck and let $s = (x_1, x_2, \ldots, x_{|\mathcal{D}|}) \in \mathsf{Seq}^{\overline{\mathcal{D}}}$ be a sequence in $\overline{\mathcal{D}}$. The visible sequence of s in $\overline{\mathcal{D}}$ is defined as follows:

$$\mathsf{vis}(s) := (\mathsf{vis}(x_1), \mathsf{vis}(x_2), \ldots, \mathsf{vis}(x_{|\mathcal{D}|})).$$

The set of all visible sequences in $\overline{\mathcal{D}}$ is defined as follows:

$$\mathsf{Vis}^{\overline{\mathcal{D}}} = \{\mathsf{vis}(s) \mid s \in \mathsf{Seq}^{\overline{\mathcal{D}}}\}.$$

■

Example 3 Let s be the sequence in Example 2. The visible sequence of s is $\mathsf{vis}(s) = (?, ?, \heartsuit, ?, ?)$. We sometimes write it by $(?^2, \heartsuit, ?^2)$ or $?^2\heartsuit?^2$. ■

Operation

Let $\overline{\mathcal{D}}$ be a deck. Let $s \in \mathsf{Seq}^{\overline{\mathcal{D}}}$ be a sequence in $\overline{\mathcal{D}}$. We consider two types of operations, conversion and opening, as follows:

- Conversion: It converts s into a new sequence $s' \in \mathsf{Seq}^{\overline{\mathcal{D}}}$. When it is deterministic, it is called a deterministic operation (e.g. permutation and turn). When it is randomized, it is called a probabilistic operation (e.g. shuffle).
- Opening: It reveals some information on s when a visible sequence of the sequence is not changed (e.g. sign opening in Sect. 3.2).

Now we define the most standard set of operations (of conversion) for binary cards. Let $\overline{\mathcal{D}} = (\mathsf{Binary}, \underline{\mathcal{D}})$ be a deck of binary cards such that $|\mathcal{D}| = \ell$ and let $s = (c_1, c_2, \ldots, c_\ell) \in \mathsf{Seq}^{\overline{\mathcal{D}}}$ be a sequence in $\overline{\mathcal{D}}$. We define three sets of operations, permutation, turning, and shuffle as follows:

Permutation For $\pi \in S_\ell$ (here S_ℓ denotes the ℓ-th symmetric group), a permutation operation (perm, π) generates a new sequence in $\overline{\mathcal{D}}$ as follows:

$$(c_1, c_2, \ldots, c_\ell) \ \rightarrow \ (c_{\pi^{-1}(1)}, c_{\pi^{-1}(2)}, \ldots, c_{\pi^{-1}(\ell)}).$$

That is, the card in the i-th position in s is moved to the $\pi(i)$-th position in the new sequence. The set of permutations Perm_ℓ for sequences of ℓ cards is defined as follows:

$$\mathsf{Perm}_\ell := \{(\mathsf{perm}, \pi) \mid \pi \in S_\ell\}.$$

Turn For a set of positions $T \subset [\ell]$ (here $[\ell]$ denotes the set $\{1, 2, \ldots, \ell\}$), a turning operation (turn, T) takes s as input and returns a new sequence $s' \in \mathsf{Seq}^{\overline{\mathcal{D}}}$ as follows:

$$(c_1, c_2, \ldots, c_\ell) \ \rightarrow \ (c'_1, c'_2, \ldots, c'_\ell),$$

where for $i \in T$, it holds $c'_i = \mathsf{turn}(c_i)$, where this "turn" is a transformation (i.e., $\mathsf{turn} \in \mathcal{T}^\flat$), and for $i \notin T$, it holds $c'_i = c_i$. The set of turnings Turn_ℓ for sequences of ℓ cards is defined as follows:

$$\mathsf{Turn}_\ell := \{(\mathsf{turn}, T) \mid T \subset [\ell]\}.$$

We note that a turning operation is not an opening but a conversion since it changes the view of a sequence. Opening is used for operations that do not change the view of a sequence.

Shuffle A shuffle operation is defined by a tuple $(\mathsf{shuffle}, \Pi, D)$, where $\Pi \subset S_\ell$ is a subset of permutations and D is a probability distribution on Π. It randomly generates a new sequence $s' \in \mathsf{Seq}^{\mathcal{D}}$ as follows:

$$(c_1, c_2, \ldots, c_\ell) \ \to \ (c_{\pi^{-1}(1)}, c_{\pi^{-1}(2)}, \ldots, c_{\pi^{-1}(\ell)}),$$

where $\pi \in \Pi$ is independently and randomly chosen according to D. The set of shuffles Shuf_ℓ for sequences of ℓ cards is defined as follows:

$$\mathsf{Shuf}_\ell := \{(\mathsf{shuffle}, \Pi, D) \mid \Pi \subset S_\ell, D \text{ is a distribution on } \Pi\}.$$

View

Let $\overline{\mathcal{D}}$ be a deck. Let $\mathcal{O}$ be a set of operations. For a sequence $s \in \mathsf{Seq}^{\overline{\mathcal{D}}}$, an operation $\mathsf{op} \in \mathcal{O}$ converts it into a new sequence $s' \in \mathsf{Seq}^{\overline{\mathcal{D}}}$ with revealed information $r \in \{0,1\}^*$ as follows:

$$s \ \to \ s' \quad \text{revealed information } r,$$

where if op is conversion, revealed information is defined by $r = \bot$, and if op is opening, s' is identical to s. What is revealed from this process to the players? Before applying op, they observe a visible sequence $\mathsf{vis}(s)$. After applying op, they observe a visible sequence $\mathsf{vis}(s')$ and revealed information r. Thus, all information revealed from the above process is $(\mathsf{vis}(s), \mathsf{vis}(s'), r)$. See sign opening and value opening in Sect. 3.2 for concrete example of openings.

Suppose that a list of k operations $\mathsf{op} \in \mathcal{O}^k$ is applied to a sequence s_0 as follows:

$$s_0 \ \to \ s_1 \ \to \ s_2 \ \to \ \cdots \ \to \ s_k.$$

Assume that the i-th operation brings revealed information $r_i \in \{0,1\}^*$. Then, all information revealed from the above process is given as follows:

$$(\mathsf{vis}(s_0), r_0) \to (\mathsf{vis}(s_1), r_1) \to (\mathsf{vis}(s_2), r_2) \to \cdots \to (\mathsf{vis}(s_k), r_k),$$

where $r_0 = \bot$ and $r_i = \bot$ if the i-th operation is conversion. This is called a view of op starting with the sequence s_0. The set of views $\mathsf{View}^{\overline{\mathcal{D}}}$ is defined as follows:

$$\mathsf{View}^{\overline{\mathcal{D}}} = \left(\mathsf{Vis}^{\overline{\mathcal{D}}} \times \{0,1\}^*\right)^*.$$

Example 4 Let $\overline{\mathcal{D}} = (\mathsf{Binary}, \mathcal{D})$ be the deck in Example 1. Let $\mathcal{O}$ be a set of operations $\mathcal{O} = \mathsf{Perm}_5 \cup \mathsf{Turn}_5$. Let op be a list of operations defined as follows:

$$\mathsf{op} = \big((\mathsf{perm}, (1\ 2)), (\mathsf{turn}, \{1, 2\}), (\mathsf{perm}, (1\ 3))\big).$$

When it is applied to a sequence $s_0 = (?/\clubsuit, ?/\heartsuit, ?/\clubsuit)$ as follows:

$$(?/\clubsuit, ?/\heartsuit, ?/\clubsuit) \ \to\ (?/\heartsuit, ?/\clubsuit, ?/\clubsuit) \ \to\ (\heartsuit/?, \clubsuit/?, ?/\clubsuit) \ \to\ (?/\clubsuit, \clubsuit/?, \heartsuit/?),$$

a view of op starting with the sequence s_0 is given as follows:

$$((?, ?, ?), \bot) \ \to\ ((?, ?, ?), \bot) \ \to\ ((\heartsuit, \clubsuit, ?), \bot) \ \to\ ((?, \clubsuit, \heartsuit), \bot).$$

We sometimes omit revealed information it is clear that all operations are conversion as follows:

$$(?, ?, ?) \ \to\ (?, ?, ?) \ \to\ (\heartsuit, \clubsuit, ?) \ \to\ (?, \clubsuit, \heartsuit).$$

We also write the above by $?^3 \to ?^3 \to \heartsuit\clubsuit? \to ?\clubsuit\heartsuit$. ■

Protocol

Protocol We define a protocol as follows:

Definition 4 (Protocol) A protocol $\mathcal{P}$ is defined by a five-tuple as follows:

$$\mathcal{P} = (n, X, \overline{\mathcal{D}}, \mathcal{O}, A),$$

where

- $n \in \mathbb{N}$ is any natural number called the number of inputs;
- X is a finite set called an input domain;
- $\overline{\mathcal{D}} = (\mathcal{C}, \mathcal{T}, \Sigma, \mathsf{vis}, \mathcal{D})$ is a deck;
- $\mathcal{O}$ is a finite set called an operation set;
- $A : \mathsf{View}^{\overline{\mathcal{D}}} \to \mathcal{O} \cup \{\bot\}$ is an action function. ■

Execution of a protocol Let $\mathcal{P} = (n, X, \overline{\mathcal{D}}, \mathcal{O}, A)$ be a protocol. Let $s_0 \in \mathsf{Seq}^{\overline{\mathcal{D}}}$ be a sequence. An execution of $\mathcal{P}$ starting with s_0 proceeds as follows:

1. The initial sequence is set to s_0 as follows:

$$s_0 = \boxed{?}\boxed{?}\boxed{?} \cdots \boxed{?}.$$

Set $s \leftarrow s_0$ and $v \leftarrow (\mathsf{vis}(s_0), \bot)$, where s is a variable of the current sequence and v is a variable of the entire view of an execution.

2. Compute the action function $A(v) = \alpha$; if $\alpha \neq \bot$, apply the operation α to the sequence s; and obtain a new sequence s' with revealed information $r \in \{0,1\}^*$; Set $s \leftarrow s'$ and append "$\rightarrow (\mathsf{vis}(s'), r)$" to v; Repeat this step until it happens $\alpha = \bot$.
3. If $A(v) = \bot$, terminate the execution.

Example 5 We describe a (slightly modified version of) six-card AND protocol by Mizuki and Sone [9] as follows:

$$(2, \{0,1\}, \overline{\mathcal{D}}, \mathcal{O}, A).$$

The deck $\overline{\mathcal{D}}$ is defined by $\overline{\mathcal{D}} = (\mathsf{Binary}, \{(\clubsuit/\mathbf{?})^3, (\heartsuit/\mathbf{?})^3\})$. The operation set $\mathcal{O}$ is defined by $\mathcal{O} = \mathsf{Perm}_6 \cup \mathsf{Turn}_6 \cup \mathsf{Shuf}_6$. The action function A is defined by:

- $A(v_0) = (\mathsf{perm}, (2\ 4\ 3))$;
- $A(v_1) = (\mathsf{shuffle}, \Pi, D)$ where $\Pi = \{\mathsf{id}, (1\ 4)(2\ 5)(3\ 6)\}$ and D is a uniform distribution over Π;
- $A(v_2) = (\mathsf{perm}, (2\ 4\ 3)^{-1})$;
- $A(v_3) = (\mathsf{turn}, \{1,2\})$;
- $A(v_4) = (\mathsf{perm}, (1\ 2)(3\ 5)(4\ 6))$;
- $A(v) = \bot$ for any $v \notin \{v_0, v_1, v_2, v_3, v_4\}$.

where

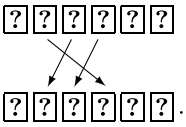
- $v_0 = (\mathbf{?}^6, \bot)$;
- $v_1 = (\mathbf{?}^6, \bot) \rightarrow (\mathbf{?}^6, \bot)$;
- $v_2 = (\mathbf{?}^6, \bot) \rightarrow (\mathbf{?}^6, \bot) \rightarrow (\mathbf{?}^6, \bot)$;
- $v_3 = (\mathbf{?}^6, \bot) \rightarrow (\mathbf{?}^6, \bot) \rightarrow (\mathbf{?}^6, \bot) \rightarrow (\mathbf{?}^6, \bot)$;
- $v_4 = (\mathbf{?}^6, \bot) \rightarrow (\mathbf{?}^6, \bot) \rightarrow (\mathbf{?}^6, \bot) \rightarrow (\mathbf{?}^6, \bot) \rightarrow (\heartsuit\clubsuit\mathbf{?}^4, \bot)$.

We describe an execution of this protocol starting with an initial sequence $s_0 = (\mathsf{com}(x_1), \mathsf{com}(x_2), \mathsf{com}(1))$ as follows:

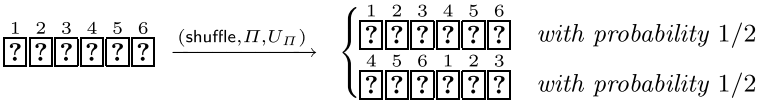
$$s_0 = \underbrace{\boxed{?}\boxed{?}}_{x_1}\underbrace{\boxed{?}\boxed{?}}_{x_2}\underbrace{\boxed{?}\boxed{?}}_{1},$$

where the commitment $\mathsf{com}(b)$ ($b \in \{0,1\}$) be two face-down cards whose front sides are $\boxed{\clubsuit}\boxed{\heartsuit}$ if $b = 0$ and $\boxed{\heartsuit}\boxed{\clubsuit}$ otherwise. The protocol proceeds as follows:

1. $(\mathsf{perm}, (2\ 4\ 3))$: Rearrange the order of the sequence as follows:

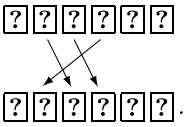


2. (shuffle, Π, D): Apply the shuffle:

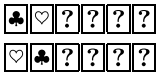


This shuffle is called a random bisection cut.

3. (perm, $(2\ 4\ 3)^{-1}$): Rearrange the order of the sequence as follows:

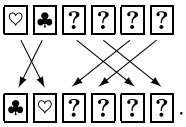


4. (turn, $\{1, 2\}$): Turn the leftmost commitment as follows:

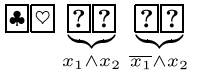


If it is the former case, i.e., the opened symbols are ♣♡, the protocol terminates. Otherwise, it proceeds to the next Step.

5. (perm, $(1\ 2)(3\ 5)(4\ 6)\}$): Rearrange the order of the sequence as follows:



After Steps 4 and 5, the protocol terminates. Then, the finial sequence is given as follows:



Since it contains a commitment to $x_1 \wedge x_2$, it is said to be an AND protocol. ■

Functionality

In order to define the correctness and the security of protocols, we introduce a notion of functionality. Informally speaking, a functionality is a pair of sequences parametrized by input variables $\mathbf{x} \in X^n$. For example, the following is the functionality $\mathcal{F}_{\text{AND}}$ of Mizuki-Sone's AND protocol (See Example 5).

$$\mathcal{F}_{\mathsf{AND}} : \underbrace{?\,?}_{x_1}\ \underbrace{?\,?}_{x_2}\ \underbrace{?\,?}_{1} \Rightarrow \clubsuit\heartsuit\ \underbrace{?\,?}_{x_1\wedge x_2}\ \underbrace{?\,?}_{\overline{x_1}\wedge x_2}.$$

It is also described as follows:

$$\mathcal{F}_{\mathsf{AND}} : (\mathsf{com}(x_1), \mathsf{com}(x_2), \mathsf{com}(1)) \Rightarrow (\clubsuit\heartsuit, \mathsf{com}(x_1 \wedge x_2), \mathsf{com}(\overline{x_1} \wedge x_2)).$$

When some part of input/output sequences in a functionality are not important, $\perp$ is used. For example, when the AND protocol does not care about the rightmost commitment in the output sequence, it is described as follows:

$$\mathcal{F}'_{\mathsf{AND}} : (\mathsf{com}(x_1), \mathsf{com}(x_2), \mathsf{com}(1)) \Rightarrow (\clubsuit\heartsuit, \mathsf{com}(x_1 \wedge x_2), \perp^2).$$

Sequence with a dummy symbol Let $\overline{\mathcal{D}} = (\mathcal{C}, \mathcal{T}, \Sigma, \mathsf{vis}, \mathcal{D})$ be a deck with $\mathcal{C} \cap \{\perp\} = \emptyset$, where $\perp$ is a dummy symbol. Let $s = (c_1, c_2, \ldots, c_\ell) \in \mathsf{Seq}^{\overline{\mathcal{D}}}$ be a sequence. A sequence $s' = (c'_1, c'_2, \ldots, c'_\ell) \in (\mathcal{C} \cup \{\perp\})^\ell$ is said to be a dummy sequence of s if $c'_i \in \{c_i, \perp\}$ for all $i \in [\ell]$. Thus, there exist 2^ℓ dummy sequences of any sequence of ℓ cards. The set of dummy sequences of s is denoted by $\mathsf{Seq}_\perp(s)$. The set of dummy sequences of $\overline{\mathcal{D}}$ is defined by

$$\mathsf{Seq}_\perp^{\overline{\mathcal{D}}} = \bigcup_{s \in \mathsf{Seq}^{\overline{\mathcal{D}}}} \mathsf{Seq}_\perp(s).$$

We say that $s \in \mathsf{Seq}^{\overline{\mathcal{D}}}$ is matched with $s' \in \mathsf{Seq}_\perp^{\overline{\mathcal{D}}}$ if $s' \in \mathsf{Seq}_\perp(s)$.

Example 6 For a sequence $s = (c_1, c_2, c_3)$, $\mathsf{Seq}_\perp(s)$ is given as follows:

$$\begin{aligned}\mathsf{Seq}_\perp(s) = \{&(c_1, c_2, c_3), (\perp, c_2, c_3), (c_1, \perp, c_3), (c_1, c_2, \perp),\\ &(\perp, \perp, c_3), (c_1, \perp, \perp), (\perp, c_2, \perp), (\perp, \perp, \perp)\}.\end{aligned}$$

For a sequence $s' = (c_1, c_2, c'_3)$ with $c'_3 \neq c_3$, s' is matched with $(c_1, c_2, \perp)$. ∎

Variable sequence Let $\overline{\mathcal{D}}$ be a deck, X be an input domain, and n be the number of inputs. A variable sequence s over $\mathsf{Seq}^{\overline{\mathcal{D}}}$ is defined by a function $s : X^n \to \mathsf{Seq}^{\overline{\mathcal{D}}}$. A variable dummy sequence s over $\mathsf{Seq}_\perp^{\overline{\mathcal{D}}}$ is defined by a function $s : X^n \to \mathsf{Seq}_\perp^{\overline{\mathcal{D}}}$.

Example 7 An input sequence $s(x)$ of Mizuki-Sone's AND protocol is a variable sequence $s : \{0,1\}^2 \to \mathsf{Seq}^{\overline{\mathcal{D}}}$ defined as follows:

$$s(x) = \begin{cases} (?/\clubsuit, ?/\heartsuit, ?/\clubsuit, ?/\heartsuit, ?/\clubsuit, ?/\heartsuit) & \text{if } x = (0,0)\\ (?/\clubsuit, ?/\heartsuit, ?/\heartsuit, ?/\clubsuit, ?/\clubsuit, ?/\heartsuit) & \text{if } x = (0,1)\\ (?/\heartsuit, ?/\clubsuit, ?/\clubsuit, ?/\heartsuit, ?/\clubsuit, ?/\heartsuit) & \text{if } x = (1,0)\\ (?/\heartsuit, ?/\clubsuit, ?/\heartsuit, ?/\clubsuit, ?/\clubsuit, ?/\heartsuit) & \text{otherwise.}\end{cases}$$

An output sequence $s'(x)$ of Mizuki-Sone's AND protocol is a variable dummy sequence $s' : \{0,1\}^2 \to \mathsf{Seq}_\perp^{\overline{\mathcal{D}}}$ defined as follows:

$$s'(x) = \begin{cases} (\clubsuit/?, \heartsuit/?, ?/\heartsuit, ?/\clubsuit, \perp^2) & \text{if } x = (1,1) \\ (\clubsuit/?, \heartsuit/?, ?/\clubsuit, ?/\heartsuit, \perp^2) & \text{otherwise.} \end{cases}$$

■

Functionality A functionality is defined as follows:

Definition 5 (Functionality) Let $\overline{\mathcal{D}}$ be a deck, X be an input domain, and n be the number of inputs. A functionality $\mathcal{F}$ is defined by a pair:

$$\mathcal{F} = (s_{\mathsf{in}}, s_{\mathsf{out}}),$$

where $s_{\mathsf{in}} : X^n \to \mathsf{Seq}^{\overline{\mathcal{D}}}$ is a variable sequence over $\mathsf{Seq}^{\overline{\mathcal{D}}}$ and $s_{\mathsf{out}} : X^n \to \mathsf{Seq}_{\perp}^{\overline{\mathcal{D}}}$ is a variable dummy sequence over $\mathsf{Seq}_{\perp}^{\overline{\mathcal{D}}}$. ■

Correctness

Correctness The correctness of protocols is defined as follows:

Definition 6 (Correctness) Let $\mathcal{P} = (n, X, \overline{\mathcal{D}}, \mathcal{O}, A)$ be a protocol. Let $\mathcal{F} = (s_{\mathsf{in}}, s_{\mathsf{out}})$ be a functionality. We say that $\mathcal{P}$ correctly realizes $\mathcal{F}$ if for any input $\mathbf{x} \in X^n$, any execution of $\mathcal{P}$ starting with $s_{\mathsf{in}}(\mathbf{x})$ terminates with a sequence s that is matched with $s_{\mathsf{out}}(\mathbf{x})$. ■

The correctness of protocols in a committed format is defined as follows:

Definition 7 (Correctness in a committed format) Let $\overline{\mathcal{D}} = (\mathcal{C}, \mathcal{T}, \Sigma, \mathsf{vis}, \mathcal{D})$ and $\overline{\mathcal{D}'} = (\mathcal{C}, \mathcal{T}, \Sigma, \mathsf{vis}, \mathcal{D}')$ be decks such that $\mathcal{D}$ contains n copies of $\mathcal{D}'$ as multi-set. ($\mathcal{C}, \mathcal{T}, \Sigma$, and vis are common.) Let $\mathcal{P} = (n, X, \overline{\mathcal{D}}, \mathcal{O}, A)$ be a protocol. Let $\mathcal{F} = (s_{\mathsf{in}}, s_{\mathsf{out}})$ be a functionality. Let $f : X^n \to X$ be a function. Let $\mathsf{com} : X \to \mathsf{Seq}^{\overline{\mathcal{D}}}$ be a function that takes an input and returns a sequence. We say that $\mathcal{P}$ correctly computes f if it satisfies the following:

- $\mathcal{P}$ correctly realizes $\mathcal{F}$;
- $s_{\mathsf{in}} = (\mathsf{com}(x_1), \mathsf{com}(x_2), \ldots, \mathsf{com}(x_n), s)$ where s is a (possibly empty) fixed sequence;
- s_{out} contains $\mathsf{com}(f(x_1, x_2, \ldots, x_n))$. ■

Security

The probability distribution of a view Let $\mathcal{P} = (n, X, \overline{\mathcal{D}}, \mathcal{O}, A)$ be a protocol. Let $s_0 \in \mathsf{Seq}^{\overline{\mathcal{D}}}$ be a sequence and let $x \in X^n$ be an input. The probability distribution of a view of $\mathcal{P}$ with input x and starting with sequence s_0 is denoted by $\mathsf{view}_{\mathcal{P}}(s_0)$, where randomness comes from probability operations (e.g., shuffles).

Security The security of protocols is defined as follows:

Definition 8 (Security) Let $\mathcal{P} = (n, X, \overline{\mathcal{D}}, \mathcal{O}, A)$ be a protocol. Let $\mathcal{F} = (s_{\mathsf{in}}, s_{\mathsf{out}})$ be a functionality. We say that $\mathcal{P}$ securely realizes $\mathcal{F}$ if for every $x, x' \in X^n$, it holds $\mathsf{view}_{\mathcal{P}}(s_{\mathsf{in}}(x)) = \mathsf{view}_{\mathcal{P}}(s_{\mathsf{in}}(x'))$. ∎

Example 8 Let us prove that the protocol given in Example 5 securely realizes the functionality $\mathcal{F}_{\mathrm{AND}} = (s_{\mathsf{in}}, s_{\mathsf{out}})$ defined as follows:

$$\mathcal{F}_{\mathrm{AND}} : (\mathsf{com}(x_1), \mathsf{com}(x_2), \mathsf{com}(1)) \Rightarrow (\clubsuit\heartsuit, \mathsf{com}(x_1 \wedge x_2), \mathsf{com}(\overline{x_1} \wedge x_2)).$$

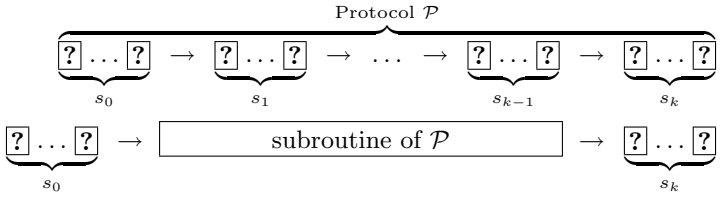
Let $x \in \{0,1\}^2$ be any input. The probability distribution of a view of the protocol starting with the sequence $s_{\mathsf{in}}(x) = (\mathsf{com}(x_1), \mathsf{com}(x_2), \mathsf{com}(1))$ is given as follows:

$$\mathsf{view}(s_{\mathsf{in}}(x)) = \begin{cases} v \to (\clubsuit\heartsuit\mathbf{?}^4, \bot) & \text{with probability } 1/2 \\ v \to (\heartsuit\clubsuit\mathbf{?}^4, \bot) \to (\clubsuit\heartsuit\mathbf{?}^4, \bot) & \text{with probability } 1/2 \end{cases}$$

where $v = (\mathbf{?}^6, \bot) \to (\mathbf{?}^6, \bot) \to (\mathbf{?}^6, \bot) \to (\mathbf{?}^6, \bot)$. Due to the random bisection cut, the above probability distribution $\mathsf{view}(s_{\mathsf{in}}(x))$ is the same for any $x \in \{0,1\}^2$. Therefore, it securely realizes the functionality. ∎

Composition of Protocols

Subroutine operation Let $\mathcal{P} = (n, X, \overline{\mathcal{D}}, \mathcal{O}, A)$ be a protocol. A subroutine of $\mathcal{P}$ is a "magical box" that executes the protocol $\mathcal{P}$ in a single step: it takes a sequence $s_0 \in \mathsf{Seq}^{\overline{\mathcal{D}}}$ as an input and outputs a final sequence of $\mathcal{P}$ when the initial sequence is s_0 as follows:



Formally, a *subroutine operation for a protocol* $\mathcal{P}$ is defined as follows:

$$(\mathsf{subroutine}, \mathcal{P}, T),$$

where $T \subset [\ell]$ is a subset of positions such that $|T|$ is the number of cards of $\mathcal{P}$. (We assume that the number of cards of $\mathcal{P}$ is equal to or less than ℓ.) The set of subroutine operations with $\mathcal{P}$ is denoted as follows:

$$\mathsf{Subroutine}_{\ell}[\mathcal{P}] = \{(\mathsf{subroutine}, \mathcal{P}, T) \mid T \subset [\ell]\}.$$

For protocols $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k$, we define the set of subroutine operations as follows:

$$\mathsf{Subroutine}_{\ell}[\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k] = \mathsf{Subroutine}_{\ell}[\mathcal{P}_1] \cup \mathsf{Subroutine}_{\ell}[\mathcal{P}_2] \cup \cdots \cup \mathsf{Subroutine}_{\ell}[\mathcal{P}_k].$$

We define an subroutine-respecting protocol as follows:

Definition 9 (Subroutine-respecting protocol) Let $\mathcal{F}_{\mathrm{sub}} = (s_{\mathsf{in}}, s_{\mathsf{out}})$ be a functionality using ℓ_{sub} cards. Let $\mathcal{P}_{\mathrm{sub}} = (n_{\mathrm{sub}}, X_{\mathrm{sub}}, \overline{\mathcal{D}}_{\mathrm{sub}}, \mathcal{O}_{\mathrm{sub}}, A_{\mathrm{sub}})$ be a protocol using ℓ_{sub} cards. Let $\mathcal{P} = (n, X, \overline{\mathcal{D}}, \mathcal{O}, A)$ be a protocol using ℓ cards ($\ell \geq \ell_{\mathrm{sub}}$). We say that $\mathcal{P}$ is subroutine-respecting for $\mathcal{P}_{\mathrm{sub}}$ and $\mathcal{F}_{\mathrm{sub}}$ if it satisfies as follows:

- $\mathsf{Subroutine}_{\ell}[\mathcal{P}_{\mathrm{sub}}] \subset \mathcal{O}$;
- For any input $x \in \{0,1\}^n$, whenever $\mathcal{P}$ enters an operation $(\mathsf{subroutine}, \mathcal{P}_{\mathrm{sub}}, T)$, the cards on positions T in the current sequence is identical to $s_{\mathsf{in}}(x')$ for some input $x' \in X_{\mathrm{sub}}$. Here, the input x' for $\mathcal{P}_{\mathrm{sub}}$ can be varied for each call of the subroutine for $\mathcal{P}_{\mathrm{sub}}$. ■

Example 9 Let $\mathcal{P}_{\mathrm{AND2}}$ be a two-bit AND protocol defined as follows:

$$\mathcal{P}_{\mathrm{AND2}} = (2, \{0,1\}, (\mathsf{Binary}, \{(\clubsuit/\mathbf{?})^3, (\heartsuit/\mathbf{?})^3\}), \mathsf{Perm}_6 \cup \mathsf{Turn}_6 \cup \mathsf{Shuf}_6, A),$$

that correctly and securely realizes a functionality $\mathcal{F}_{\mathrm{AND2}}$ as follows:

$$\mathcal{F}_{\mathrm{AND2}} : \underbrace{?\,?}_{x_1}\ \underbrace{?\,?}_{x_2}\ \underbrace{?\,?}_{1} \Rightarrow \underbrace{\bot\,\bot}\ \underbrace{?\,?}_{x_1 \wedge x_2}\ \underbrace{?\,?}_{1}$$

This is obtained from Mizuki and Sone's AND protocol in Example 5 with a small modification. By using the subroutine of $\mathcal{P}_{\mathrm{AND2}}$, we construct an eight-card three-bit AND protocol $\mathcal{P}_{\mathrm{AND3}}$ defined as follows:

$$\mathcal{P}_{\mathrm{AND3}} = (3, \{0,1\}, (\mathsf{Binary}, \{(\clubsuit/\mathbf{?})^4, (\heartsuit/\mathbf{?})^4\}), \mathsf{Subroutine}_8[\mathcal{P}_{\mathrm{AND2}}], A').$$

that realizes a functionality $\mathcal{F}_{\mathrm{AND3}} = (s_{\mathsf{in}}, s_{\mathsf{out}})$ as follows:

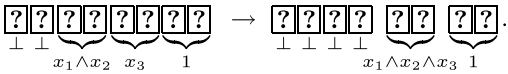
$$\mathcal{F}_{\mathrm{AND3}} : \underbrace{?\,?}_{x_1}\ \underbrace{?\,?}_{x_2}\ \underbrace{?\,?}_{x_3}\ \underbrace{?\,?}_{1} \Rightarrow \underbrace{\bot\,\bot\,\bot\,\bot}\ \underbrace{?\,?}_{x_1 \wedge x_2 \wedge x_3}\ \underbrace{?\,?}_{1}.$$

It proceeds as follows:

1. $(\mathsf{subroutine}, \mathcal{P}_{\mathrm{AND2}}, \{1,2,3,4,7,8\})$: Apply the two-bit AND protocol for cards on $\{1,2,3,4,7,8\}$ as follows:

$$\underbrace{?\,?}_{x_1}\ \underbrace{?\,?}_{x_2}\ \underbrace{?\,?}_{x_3}\ \underbrace{?\,?}_{1} \rightarrow \underbrace{\bot\,\bot}\ \underbrace{?\,?}_{x_1 \wedge x_2}\ ?\,?\ ?\,?.$$

2. $(\mathsf{subroutine}, \mathcal{P}_{\mathrm{AND2}}, \{3,4,5,6,7,8\})$: Apply the two-bit AND protocol for cards on $\{3,4,5,6,7,8\}$ as follows:



We can observe that the protocol $\mathcal{P}_{\mathrm{AND3}}$ is subroutine-respecting for $\mathcal{P}_{\mathrm{AND2}}$ and $\mathcal{F}_{\mathrm{AND2}}$: the first condition in Definition 9 is satisfied since the operation set of $\mathcal{P}_{\mathrm{AND3}}$ is $\mathsf{Subroutine}_8[\mathcal{P}_{\mathrm{AND2}}]$; and, the second condition in Definition 9 is satisfied since for each call of the subroutine $\mathcal{P}_{\mathrm{AND2}}$, the cards on positions T in the sequence is identical to $s_{\mathsf{in}}(x')$ for some $x' \in \{0,1\}^2$. ■

Proposition 1 (*Composition theorem*) *Let* $\mathcal{P}_i = (n_i, X_i, \overline{\mathcal{D}}_i, \mathcal{O}_i, A_i)$ $(i \in [k])$ *be a protocol that correctly and securely realizes a functionality* $\mathcal{F}_i$. *Let* $\mathcal{P} = (n, X, \overline{\mathcal{D}}, \mathcal{O} \cup \mathsf{Subroutine}_\ell[\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k], A)$ *be a protocol that is subroutine-respecting for* $\mathcal{P}_i$ *and* $\mathcal{F}_i$, *and* $\mathcal{O}$ *is upward compatible with* $\mathcal{O}_i$ *for every* $i \in [k]$. *If* $\mathcal{P}$ *correctly and securely realizes a functionality* $\mathcal{F}$, *then there exists a protocol* $\mathcal{P}' = (n, X, \overline{\mathcal{D}}, \mathcal{O}, A)$ *that correctly and securely realizes* $\mathcal{F}$. ■

Proof The protocol $\mathcal{P}'$ is obtained from the protocol $\mathcal{P}$ by replacing all subroutine calls of $\mathcal{P}_i$ with the protocols $\mathcal{P}_i$ for all $i \in [k]$. We can observe that the final sequence of $\mathcal{P}$ and that of $\mathcal{P}'$ are the same since $\mathcal{P}$ is subroutine-respecting. Thus, $\mathcal{P}'$ correctly realizes $\mathcal{F}$. We can also observe that a view of $\mathcal{P}'$ is obtained from a view of $\mathcal{P}$ by replacing all subroutine calls of $\mathcal{P}_i$ with a view of $\mathcal{P}_i$ for all $i \in [k]$. Since $\mathcal{P}$ and $\mathcal{P}_i$ securely realize $\mathcal{F}$ and $\mathcal{F}_i$, respectively, for all $i \in [k]$. Thus, $\mathcal{P}'$ also securely realizes $\mathcal{F}$. ■

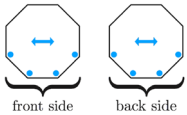
Dihedral Cards

Dihedral Cards

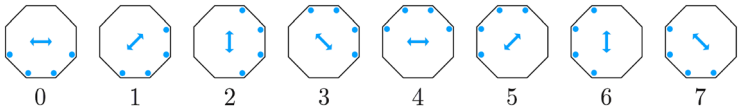
Let $m \geq 2$ be any integer. A dihedral card of modulus m is a card as follows:

- It holds a non-binary value $x \in \mathbb{Z}_{2m}$;
- A transformation from x to $x + c$ (for any constant $c \in \mathbb{Z}_{2m}$) is allowed;
- A transformation from x to $-x + c$ (for any constant $c \in \mathbb{Z}_{2m}$) is allowed;
- For a card holding x, it is possible to observe whether $x \geq m$ only;
- For a card holding x, it is possible to observe x mod m only.

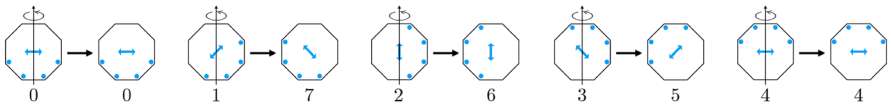
Thus, the shape of dihedral cards of modulus m is a regular $2m$-sided polygon. For example, a dihedral card of modulus 4 is implemented as follows:



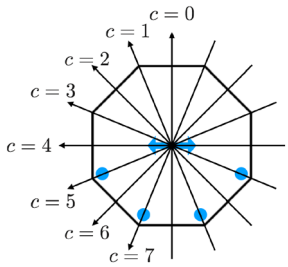
Four vertices among eight vertices have blue dots and an arrow is written on the center. The front side and the back side are the same pattern satisfying that any vertex having a blue dot in the front side also has a dot in the back side. Here, all blue circles and arrows are written by invisible ink[1] in order to hide a value of a card. Since it is a hexagon, it can hold a value $x \in \mathbb{Z}_8$ as follows:



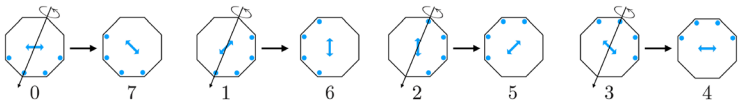
The first transformation from x to $x + c$ is done by a rotation with $(360c/2m)°$ as in the case of cyclic cards. A nontrivial property is to allow the second transformation from x to $-x + c$. This is done by a flipping. Say $c = 0$. A transformation from x to $-x$ is done by a flipping with a vertical line as follows:



For $m = 4$, each axis of line symmetry corresponds to some $c \in \mathbb{Z}_8$ as follows:

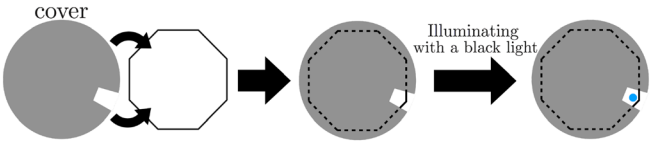


Indeed, a transformation from x to $-x + 7$ is done by a flipping as follows:

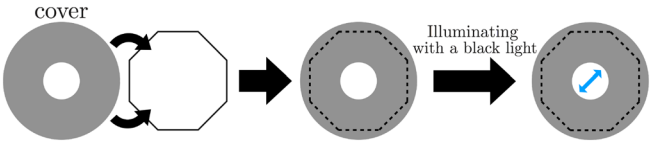


[1] Invisible ink is used for writing, which is invisible but can be made visible with illuminating a black light. It can be used for steganography, which hides the existence of plain texts while cryptography hides the contents of plain texts.

For a general modulus m, an axis of line symmetry rotated by $(180c/2m)°$ from the vertical line corresponds to $c \in \mathbb{Z}_{2m}$. Finally, we need to open a bit $\mathsf{p}(x \geq m)$ and a value x mod m. Here, p(statement) is a predicate that outputs 1 if the statement is true and 0 false. Thanks to the property of invisible ink, this is done by illuminating a black light with a cover. For a card holding x, it is possible to observe $\mathsf{p}(x \geq m)$ only as follows:



In the above case, since the vertex has a blue dot, the predicate $\mathsf{p}(x \geq m)$ is 0. (We can observe that for a card holding x, the vertex has a blue dot if and only if $x < 4$.) Similarly, it is possible to observe the value x mod m only as follows:



In the above case, since the card holds either 1 or 5, the value x mod m is 1. For $x \in \mathbb{Z}_{2m}$, $\mathsf{p}(x \geq m)$ is called a sign of x and x mod m is called a value of x.

A card specification of dihedral cards For $x \in \mathbb{Z}_{2m}$, we denote a card holding x by $[\![x]\!]$. The card set of dihedral cards of modulus m, denoted by $\mathcal{C}_m^{\mathsf{d}}$, is defined as follows:

$$\mathcal{C}_m^{\mathsf{d}} = \{[\![0]\!], [\![1]\!], \ldots, [\![2m-1]\!]\}.$$

Let $[\![x]\!] \in \mathcal{C}_m^{\mathsf{d}}$ be a card holding a value $x \in \mathbb{Z}_{2m}$. For any constant $a \in \mathbb{Z}_{2m}$, a rotation operation with a degree a is defined as follows:

$$\mathsf{rot}^a([\![x]\!]) = [\![x + a]\!]$$

For any constant $a \in \mathbb{Z}_{2m}$, a flipping operation with an axis a is defined as follows:

$$\mathsf{flip}_a([\![x]\!]) = [\![-x + a]\!].$$

The transformation set of dihedral cards of modulus m, denoted by $\mathcal{T}_m^{\mathsf{d}}$, is defined as follows:

$$\mathcal{T}_m^{\mathsf{d}} = \{\mathsf{id}, \mathsf{rot}, \mathsf{rot}^2, \ldots, \mathsf{rot}^{2m-1}, \mathsf{flip}_0, \mathsf{flip}_1, \ldots, \mathsf{flip}_{2m-1}\}.$$

The symbol set of dihedral cards of modulus m, denoted by Σ_m^{d}, is defined as follows:

$$\Sigma_m^{\mathsf{d}} = \{?\}.$$

The vision function $\mathsf{vis}_m^{\mathsf{d}} : \mathcal{C}_m^{\mathsf{d}} \to \Sigma_m^{\mathsf{d}}$ of dihedral cards of modulus m is defined as follows:

$$\mathsf{vis}_m^{\mathsf{d}}([\![x]\!]) = ? \quad \text{for any } \ x \in \mathbb{Z}_{2m}.$$

A card specification of dihedral cards of modulus m, denoted by $\mathsf{Dihedral}_m$, is defined as follows:

$$\mathsf{Dihedral}_m = (\mathcal{C}_m^{\mathsf{d}}, \mathcal{T}_m^{\mathsf{d}}, \Sigma_m^{\mathsf{d}}, \mathsf{vis}_m^{\mathsf{d}}).$$

Commitment A commitment to $x \in \mathbb{Z}_{2m}$ is defined by $[\![x]\!]$.

Operations for Dihedral Cards

For dihedral cards, we introduce eight operations: permutation, rotation, rotation shuffle, flipping, flipping shuffle, two-sided rotation shuffle, sign opening, and value opening.

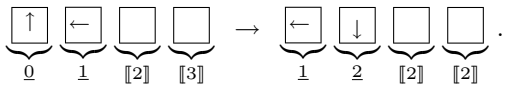
Permutation This operation is the same as permutation for binary cards in Sect. 2.2. For modulus m, the set of permutations $\mathsf{Perm}_{m,\ell}$ for sequences of ℓ dihedral cards with modulus m is defined as follows:

$$\mathsf{Perm}_{m,\ell} := \{(\mathsf{perm}, \pi) \mid \pi \in S_\ell\}.$$

Rotation For $T \subset [\ell]$ and $a \in \mathbb{Z}_m$, a rotation operation is defined as follows:

$$(\mathsf{rot}, T, a).$$

For a sequence $s = (c_1, c_2, \ldots, c_\ell) \in \mathsf{Seq}^{\overline{\mathcal{D}}}$, by applying a rotation operation (rot, T, a), it is transformed into a new sequence $s' = (c'_1, c'_2, \ldots, c'_\ell) \in \mathsf{Seq}^{\overline{\mathcal{D}}}$ such that $c'_i = \mathsf{rot}^a(c_i)$ for all $i \in T$ and $c'_i = c_i$ for all $i \notin T$. For example, for a sequence $s = (\underline{0}, \underline{1}, [\![2]\!], [\![3]\!])$ with modulus $m = 4$, a rotation operation $(\mathsf{rot}, \{1, 2, 4\}, 1)$ transforms it into a new sequence $s' = (\underline{1}, \underline{2}, [\![2]\!], [\![2]\!])$ as follows:



The set of rotations $\mathsf{Rot}_{m,\ell}$ is defined as follows:

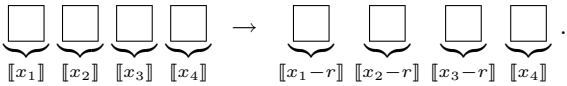
$$\mathsf{Rot}_{m,\ell} = \{(\mathsf{rot}, T, a) \mid T \subset [\ell], a \in \mathbb{Z}_m\}.$$

Rotation shuffle For $T \subset [\ell]$, a rotation shuffle is defined as follows:

$$(\mathsf{rotshuf}, T).$$

For all $i \in T$, the i-th card in the sequence is rotated with a degree $r \in \mathbb{Z}_m$, here r is uniformly and randomly chosen from $\mathbb{Z}_m$ and this r is common for all $i \in T$.

The other cards are unchanged. For example, for a sequence $([\![x_1]\!], [\![x_2]\!], [\![x_3]\!], [\![x_4]\!])$ with modulus $m = 4$, a rotation shuffle $(\mathsf{rotshuf}, \{1, 2, 3\})$ generates a sequence $([\![x_1 - r]\!], [\![x_2 - r]\!], [\![x_3 - r]\!], [\![x_4]\!])$ for a random $r \in \mathbb{Z}/4\mathbb{Z}$ as follows:



The set of rotation shuffles is defined as follows:

$$\mathsf{RotShuf}_{m,\ell} = \{(\mathsf{rotshuf}, T) \mid T \subset [\ell]\}.$$

Flipping A flipping operation is defined as follows:

$$(\mathsf{flip}, a, T),$$

where $a \in \mathbb{Z}_{2m}$ is an axis of flipping and $T \subset [\ell]$ is a subset of positions. By applying a flipping operation (flip, a, T), a sequence is converted as follows:

$$([\![x_1]\!], [\![x_2]\!], \ldots, [\![x_\ell]\!]) \rightarrow ([\![x'_1]\!], [\![x'_2]\!], \ldots, [\![x'_\ell]\!]),$$

where $x'_i = -x_i + a$ for all $i \in T$ and $x'_i = x_i$ for all $i \notin T$. For example, for a sequence $([\![0]\!], [\![2]\!], [\![5]\!], [\![7]\!])$ of modulus $m = 4$, a flipping operation $(\mathsf{flip}, 0, \{1, 2, 3, 4\})$ converts it into a new sequence $([\![0]\!], [\![6]\!], [\![3]\!], [\![1]\!])$. The set of flipping operations $\mathsf{Flip}_{m,\ell}$ is defined as follows:

$$\mathsf{Flip}_{m,\ell} = \{(\mathsf{flip}, j, T) \mid j \in \mathbb{Z}_{2m}, T \subset [\ell]\}.$$

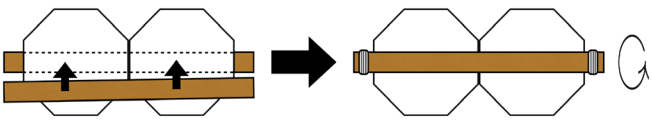
Flipping shuffle A flipping shuffle is defined as follows:

$$(\mathsf{flipshuf}, (a_1, a_2, \ldots, a_k), T_1, T_2, \ldots, T_k),$$

where $k \in [\ell]$ is the number of axes, $a_1, a_2, \ldots, a_k \in \mathbb{Z}_{2m}$ are axes of flipping and $T_1, T_2, \ldots, T_k \subset [\ell]$ are disjoint subsets of positions. For all $1 \leq i \leq k$, all cards on T_i are flipped (by flip_{a_i}) randomly and simultaneously. Here, the random bit designating whether flipped or not is common for all i. The other cards are unchanged. For example, for a sequence $([\![0]\!], [\![2]\!], [\![5]\!], [\![7]\!])$ of modulus $m = 4$, a flipping shuffle $(\mathsf{flipshuf}, (0, 1), \{1, 2\}, \{3, 4\})$ generates a new sequence:

$$([\![0]\!], [\![2]\!], [\![5]\!], [\![7]\!]) \rightarrow \begin{cases} ([\![0]\!], [\![2]\!], [\![5]\!], [\![7]\!]) & \text{with probability } 1/2 \\ ([\![0]\!], [\![6]\!], [\![4]\!], [\![2]\!]) & \text{with probability } 1/2 \end{cases}$$

A flipping shuffle is implemented by using two wooden boards as follows:



The set of flipping shuffles is defined as follows:

$$\begin{aligned}\mathsf{FlipShuf}_{m,\ell} = \{(\mathsf{flipshuf}, (a_1, a_2, \ldots, a_k), T_1, T_2, \ldots, T_k) \mid \\ k \in [\ell],\ a_1, a_2, \ldots, a_k \in \mathbb{Z}_{2m}, \\ T_1, T_2, \ldots, T_k \subset [\ell] \text{ s.t. } \forall a, b \in [k], T_a \cap T_b = \emptyset\}.\end{aligned}$$

Two-sided rotation shuffle A two-sided rotation shuffle is defined by:

$$(\mathsf{twoshuf}, T),$$

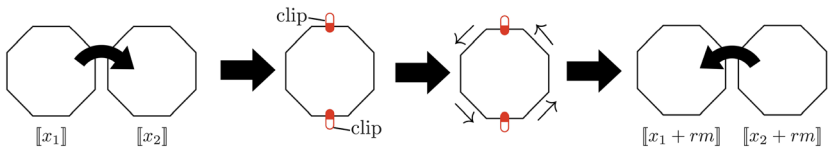
where $T \subset [\ell]$ is a subset of positions. By applying a two-sided rotation shuffle $(\mathsf{twoshuf}, T)$, a sequence is converted as follows:

$$([\![x_1]\!], [\![x_2]\!], \ldots, [\![x_\ell]\!]) \ \rightarrow \ ([\![x'_1]\!], [\![x'_2]\!], \ldots, [\![x'_\ell]\!]),$$

where $x'_i = x_i + rm$ for a random bit $r \in \{0, 1\}$ if $i \in T$ and $x'_i = x_i$ otherwise. Note that the random bit r is common for all $i \in T$. For example, for a sequence $([\![0]\!], [\![2]\!], [\![5]\!], [\![7]\!])$ of modulus $m = 4$, a two-sided rotation shuffle $(\mathsf{twoshuf}, \{1, 2, 3, 4\})$ generates a new sequence as follows:

$$([\![0]\!], [\![2]\!], [\![5]\!], [\![7]\!]) \ \rightarrow \ \begin{cases} ([\![0]\!], [\![2]\!], [\![5]\!], [\![7]\!]) & \text{with probability } 1/2 \\ ([\![4]\!], [\![6]\!], [\![1]\!], [\![3]\!]) & \text{with probability } 1/2 \end{cases}$$

A two-sided rotation shuffle is implemented by using two clips as follows:



The set of two-sided rotation shuffles is defined as follows:

$$\mathsf{TwoShuf}_{m,\ell} = \{(\mathsf{twoshuf}, T) \mid T \subset [\ell]\}.$$

Sign opening A sign opening is defined as follows:

$$(\mathsf{sgnopen}, i),$$

where $i \in [\ell]$ is a position. For a sequence $([\![x_1]\!], [\![x_2]\!], \ldots, [\![x_\ell]\!])$, it publicly reveals a bit value $\mathsf{p}(x_i \geq m) \in \{0, 1\}$. It is treated as revealed information. That is, it outputs revealed information $r = \mathsf{p}(x_i \geq m)$ without changing the sequence. For example, for a sequence $([\![0]\!], [\![2]\!], [\![5]\!], [\![7]\!])$ of modulus $m = 4$, a sign opening $(\mathsf{sgnopen}, 3)$ outputs the sign of the third card "1" $(\mathsf{p}(5 \geq 4))$ as revealed information. The set of sign openings is defined as follows:

$$\mathsf{SgnOpen}_{m,\ell} = \{(\mathsf{sgnopen}, i) \mid i \subset [\ell]\}.$$

Value opening A value opening is defined as follows:

$$(\mathsf{valopen}, i),$$

where $i \in [\ell]$ is a position. For a sequence $([\![x_1]\!], [\![x_2]\!], \ldots, [\![x_\ell]\!])$, it publicly reveals a value $x_i \bmod m \in \mathbb{Z}_m$. It is treated as revealed information. That is, it outputs revealed information $r = (x_i \bmod m)$ without changing the sequence. For example, for a sequence $([\![0]\!], [\![2]\!], [\![5]\!], [\![7]\!])$ of modulus $m = 4$, a value opening $(\mathsf{valopen}, 4)$ outputs the value of the fourth card "3" $(= 7 \bmod 4)$ as revealed information. The set of value openings is defined as follows:

$$\mathsf{ValOpen}_{m,\ell} = \{(\mathsf{valopen}, i) \mid i \subset [\ell]\}.$$

Full opening A full opening is defined as follows:

$$(\mathsf{open}, i),$$

where $i \in [\ell]$ is a position. For a sequence $([\![x_1]\!], [\![x_2]\!], \ldots, [\![x_\ell]\!])$, it publicly reveals a value $x_i \in \mathbb{Z}_{2m}$. It is treated as revealed information. Note that it is equivalent to applying a sign opening and a value opening successively. Thus, the full opening can be viewed as a syntax sugar of applying a sign opening and a value opening successively.

Notations

Hereafter, we use notations as follows.

Operations We assume that the set of operations is $\mathcal{O}^{\mathrm{d}}_{m,\ell}$ defined as follows:

$$\begin{aligned}\mathcal{O}^{\mathrm{d}}_{m,\ell} =& \mathsf{Perm}_{m,\ell} \cup \mathsf{Rot}_{m,\ell} \cup \mathsf{RotShuf}_{m,\ell} \cup \mathsf{Flip}_{m,\ell} \cup \mathsf{FlipShuf}_{m,\ell} \\ & \cup \mathsf{TwoShuf}_{m,\ell} \cup \mathsf{SgnOpen}_{m,\ell} \cup \mathsf{ValOpen}_{m,\ell}.\end{aligned}$$

Protocols with Dihedral Cards

Initialization Protocol

Functionality A functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{init}}$ is defined as follows:

$$\mathcal{F}^{\mathrm{d}}_{\mathrm{init}} : [\![x]\!] \Rightarrow [\![0]\!].$$

where $x \in \mathbb{Z}_{2m}$.

Protocol An initialization protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{init}}$ is defined as follows:

$$\mathcal{P}^{\mathrm{d}}_{\mathrm{init}} = (1, \mathbb{Z}_{2m}, (\mathsf{Dihedral}_m, \{[\![0]\!]\}), \mathcal{O}^{\mathrm{d}}_{m,1}, A).$$

It proceeds as follows:

1. $(\mathsf{rotshuf}, \{1\})$: Apply a rotation shuffle to it:

$$[\![x]\!] \ \rightarrow \ [\![x']\!].$$

2. (open, 1): Apply a full opening operation to it. Let $x' \in \mathbb{Z}_{2m}$ be the opened value, which is treated as revealed information.

$$\text{revealed information } x'.$$

3. (rot, $\{1\}, -x'$): Rotate it with a degree $-x'$ as follows:

$$[\![x']\!] \ \rightarrow \ [\![0]\!]$$

The protocol terminates.

Correctness The correctness is trivial.

Security Let $x \in \mathbb{Z}_{2m}$ be any input. The probability distribution of a view of the protocol starting with the sequence $s_{\mathsf{in}}(x) = [\![x]\!]$ is given as follows:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{init}}}(s_{\mathsf{in}}(x)) = \big((?, \perp) \rightarrow (?, \perp) \rightarrow (?, x') \rightarrow (?, \perp)\big),$$

where $x' = x + r$ for a uniform random value $r \in \mathbb{Z}_{2m}$. This is equivalent to a probability distribution view^* defined as follows:

$$\mathsf{view}^* = \big((?, \perp) \rightarrow (?, \perp) \rightarrow (?, r') \rightarrow (?, \perp)\big),$$

where $r' \in \mathbb{Z}_{2m}$ is a uniform random value. The distribution view^* does not depend on x. Thus, for every $x, x' \in \mathbb{Z}_{2m}$, the following holds:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{init}}}(s_{\mathsf{in}}(x)) = \mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{init}}}(s_{\mathsf{in}}(x')) = \mathsf{view}^*.$$

Therefore, $\mathcal{P}^{\mathrm{d}}_{\mathrm{init}}$ securely realizes $\mathcal{F}^{\mathrm{d}}_{\mathrm{init}}$.

Efficiency The number of cards is one. Note that this is the minimum number of cards. The number of probabilistic operations is one (one rotation shuffle).

Addition Protocol

Functionality A functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{add}}$ is defined as follows:

$$\mathcal{F}^{\mathrm{d}}_{\mathrm{add}} : ([\![x_1]\!], [\![x_2]\!]) \ \Rightarrow \ ([\![0]\!], [\![x_1 + x_2]\!]).$$

where $x_1, x_2 \in \mathbb{Z}_{2m}$.

Protocol An addition protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{add}}$ is defined as follows:

$$\mathcal{P}^{\mathrm{d}}_{\mathrm{add}} = (2, \mathbb{Z}_{2m}, (\mathsf{Dihedral}_m, \{[\![0]\!], [\![0]\!]\}), \mathcal{O}^{\mathrm{d}}_{m,2}, A).$$

It proceeds as follows:

1. (flip, 0, $\{1\}$): Flip the left card along with the 0-axis as follows:

$$([\![x_1]\!], [\![x_2]\!]) \ \rightarrow \ ([\![-x_1]\!], [\![x_2]\!]).$$

2. $(\mathsf{rotshuf}, \{1,2\})$: Apply a rotation shuffle to them:

$$([\![-x_1]\!], [\![x_2]\!]) \ \rightarrow \ ([\![x'_1]\!], [\![x'_2]\!]).$$

3. $(\mathsf{open}, 1)$: Apply a full opening operation to the left card. Let $x'_1 \in \mathbb{Z}_{2m}$ be the opened value, which is treated as revealed information.

$$\text{revealed information } x'_1.$$

4. $(\mathsf{rot}, \{1,2\}, -x'_1)$: Rotate them so that they are added by $-x'_1$:

$$([\![x'_1]\!], [\![x'_2]\!]) \ \rightarrow \ ([\![0]\!], [\![x'_2 - x'_1]\!])$$

Correctness By the rotation shuffle, $x'_1 = -x_1 + r$ and $x'_2 = x_2 + r$ for a uniform random value $r \in \mathbb{Z}_{2m}$. The right card in the final sequence is $[\![x'_2 - x'_1]\!] = [\![(x_2 + r) - (-x_1 + r)]\!] = [\![x_1 + x_2]\!]$. Therefore, the above protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{add}}$ correctly realizes the functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{add}}$.

Security Let $x = (x_1, x_2) \in (\mathbb{Z}_{2m})^2$ be any input. The probability distribution of a view of the protocol starting with the sequence $s_{\mathsf{in}}(x) = ([\![x_1]\!], [\![x_2]\!])$ is given as follows:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{add}}}(s_{\mathsf{in}}(x)) = \big((?^2, \bot) \rightarrow (?^2, \bot) \rightarrow (?^2, \bot) \rightarrow (?^2, x'_1) \rightarrow (?^2, \bot)\big),$$

Since $x'_1 = x_1 + r$ for a uniform random value $r \in \mathbb{Z}_{2m}$ is distributed uniformly randomly, the above distribution is equivalent to a probability distribution view^* defined as follows:

$$\mathsf{view}^* = \big((?^2, \bot) \rightarrow (?^2, \bot) \rightarrow (?^2, \bot) \rightarrow (?^2, r') \rightarrow (?^2, \bot)\big).$$

where $r' \in \mathbb{Z}_{2m}$ is a uniform random value. The distribution view^* does not depend on x. Thus, for every $x, x' \in \mathbb{Z}_{2m}$, the following holds:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{add}}}(s_{\mathsf{in}}(x)) = \mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{add}}}(s_{\mathsf{in}}(x')) = \mathsf{view}^*.$$

Therefore, $\mathcal{P}^{\mathrm{d}}_{\mathrm{add}}$ securely realizes $\mathcal{F}^{\mathrm{d}}_{\mathrm{add}}$.

Efficiency The number of cards is two. Note that this is the minimum number of cards since the number of inputs is two. The number of probabilistic operations is one (one rotation shuffle).

Sign Normalization Protocol

Functionality A functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{sign}}$ is defined as follows:

$$\mathcal{F}^{\mathrm{d}}_{\mathrm{sign}} \ : \ [\![x]\!] \ \Rightarrow \ [\![x \bmod m]\!],$$

where $x \in \mathbb{Z}_{2m}$.

Protocol A protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{sign}}$ is defined as follows:

$$\mathcal{P}^{\mathrm{d}}_{\mathrm{sign}} = (1, \mathbb{Z}_{2m}, (\mathsf{Dihedral}_m, \{[\![0]\!]\}), \mathcal{O}^{\mathrm{d}}_{m,1}, A).$$

It proceeds as follows:

1. $(\mathsf{twoshuf}, \{1\})$: Apply a two-sided rotation shuffle to the input card as follows:

$$[\![x]\!] \to [\![x']\!],$$

   where $x' = x + rm$ for a uniform random bit $r \in \{0, 1\}$.
2. $(\mathsf{sgnopen}, 1)$: Apply the sign opening to the card. Let $s' \in \{0, 1\}$ be the sign of the card, which is treated as revealed information.

$$[\![x']\!] \to [\![x']\!], \quad \text{revealed information } s'.$$

3. $(\mathsf{rot}, \{1\}, s'm)$: Rotate the card with a degree $s'm$:

$$[\![x']\!] \to [\![x' + s'm]\!].$$

Correctness Let $x = v + sm$ for $v \in \mathbb{Z}_m$ and $s \in \{0, 1\}$. Due to the property of a two-sided rotation shuffle, x' is represented by $x' = v + (s \oplus r)m$ and s' is represented by $s' = s \oplus r$. Thus, the card in the final sequence is $[\![x' + s'm]\!] = [\![v + (s \oplus r)m + s'm]\!] = [\![v + (s \oplus r)m + (s \oplus r)m]\!] = [\![v]\!]$. (Note that every computation is done over $\mathbb{Z}_{2m}$.) Therefore, the above protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{sign}}$ correctly realizes the functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{sign}}$.

Security Let $x = v + sm \in \mathbb{Z}_{2m}$ ($v \in \mathbb{Z}_m$ and $s \in \{0, 1\}$) be any input. The probability distribution of a view of the protocol starting with the sequence $s_{\mathsf{in}}(x) = [\![x]\!]$ is given as follows:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{sign}}}(s_{\mathsf{in}}(x)) = \big((?, \bot) \to (?, s') \to (?, \bot) \to (?, \bot)\big),$$

where $s' = s \oplus r \in \{0, 1\}$ for a uniform random bit r. It is equivalent to a probability distribution view^* defined as follows:

$$\mathsf{view}^* = \big((?, \bot) \to (?, r') \to (?, \bot) \to (?, \bot)\big).$$

where $r' \in \{0, 1\}$ is a uniform random value. Thus, for every $x, x' \in \mathbb{Z}_{2m}$, the following holds:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{sign}}}(s_{\mathsf{in}}(x)) = \mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{sign}}}(s_{\mathsf{in}}(x')) = \mathsf{view}^*.$$

Therefore, $\mathcal{P}^{\mathrm{d}}_{\mathrm{sign}}$ securely realizes $\mathcal{F}^{\mathrm{d}}_{\mathrm{sign}}$.

Efficiency The number of cards is one. Note that this is the minimum number of cards. The number of probabilistic operations is one (one two-sided rotation shuffle).

Sign-to-Value Protocol

Functionality A functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{sv}}$ is defined as follows:

$$\mathcal{F}^{\mathrm{d}}_{\mathrm{sv}} : ([\![x]\!], [\![0]\!]) \Rightarrow ([\![\mathsf{p}(x \geq m)]\!], [\![0]\!]),$$

where $x \in \mathbb{Z}_{2m}$.

Protocol A protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}$ is defined as follows:

$$\mathcal{P}^{\mathrm{d}}_{\mathrm{sv}} = (1, \mathbb{Z}_{2m}, (\mathsf{Dihedral}_m, \{[\![0]\!], [\![0]\!]\}), \mathcal{O}^{\mathrm{d}}_{m,2} \cup \mathsf{Subroutine}[\mathcal{P}^{\mathrm{d}}_{\mathrm{init}}], A).$$

It proceeds as follows:

1. $(\mathsf{twoshuf}, \{1\})$: Apply a two-sided rotation shuffle to the input card as follows:

$$([\![x]\!], [\![0]\!]) \rightarrow ([\![x + r_1 m]\!], [\![r_1 m]\!]),$$

   where $r_1 \in \{0, 1\}$ is a uniform random bit.
2. $(\mathsf{sgnopen}, 1)$: Apply the sign opening to the left card. Let $s_1 \in \{0, 1\}$ be the sign of the left card, which is treated as revealed information. (We can observe that $s_1 = \mathsf{p}(x \geq m) \oplus r_1$.)
3. $(\mathsf{rot}, \{2\}, s_1 m)$: Rotate the right card with a degree $s_1 m$:

$$([\![x + r_1 m]\!], [\![r_1 m]\!]) \rightarrow ([\![x + r_1 m]\!], [\![(r_1 \oplus s_1) m]\!]).$$

4. $(\mathsf{subroutine}, \mathcal{P}^{\mathrm{d}}_{\mathrm{init}}, \{1\})$: Apply the initialization protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{init}}$ as follows:

$$([\![x + r_1 m]\!], [\![(r_1 \oplus s_1) m]\!]) \rightarrow ([\![0]\!], [\![(r_1 \oplus s_1) m]\!]).$$

5. $(\mathsf{flipshuf}, (\mathsf{flip}_1, \mathsf{flip}_m), (1, 2))$: Apply a flipping shuffle as follows:

$$([\![0]\!], [\![(r_1 \oplus s_1) m]\!]) \rightarrow ([\![r_2]\!], [\![(r_1 \oplus s_1 \oplus r_2) m]\!]),$$

   where $r_2 \in \{0, 1\}$ is a uniform random bit.
6. $(\mathsf{sgnopen}, 2)$: Apply the sign opening to the right card. Let $s_2 \in \{0, 1\}$ be the sign of the right card, which is treated as revealed information. (We can observe that $s_2 = r_1 \oplus s_1 \oplus r_2$.) If $s_2 = 0$, the protocol terminates.
7. $(\mathsf{rot}, \{2\}, m)$: If $s_2 = 1$, rotate the right card with a degree m:

$$([\![r_2]\!], [\![m]\!]) \rightarrow ([\![r_2]\!], [\![0]\!]).$$

8. $(\mathsf{flip}, 1, \{1\})$: If $s_2 = 1$, apply a flipping with an axis 1 as follows:

$$([\![r_2]\!], [\![0]\!]) \rightarrow ([\![-r_2 + 1]\!], [\![0]\!]).$$

   The protocol terminates.

Correctness If $s_2 = 0$ at Step 6, the protocol terminates. In this case, the left card in the final sequence is given as follows:

$$[\![r_2]\!] = [\![r_1 \oplus s_1]\!] = [\![\mathsf{p}(x \geq m)]\!].$$

If $s_2 = 1$ at Step 6, the protocol proceeds to Step 8. In this case, the left card in the final sequence is given as follows:

$$[\![-r_2 + 1]\!] = [\![-(1 - r_1 \oplus s_1) + 1]\!] = [\![r_1 \oplus s_1]\!] = [\![\mathsf{p}(x \geq m)]\!].$$

Therefore, the above protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}$ correctly realizes the functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{sv}}$.

Security Let $x = v + sm \in \mathbb{Z}_{2m}$ ($v \in \mathbb{Z}_m$ and $s \in \{0,1\}$) be any input. The probability distribution of a view of the protocol starting with the sequence $s_{\mathsf{in}}(x) = ([\![x]\!], [\![0]\!])$ is given as follows:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}}(s_{\mathsf{in}}(x)) = \Big((?^2,\bot) \to (?^2,\bot) \to (?^2,s_1) \to (?^2,\bot) \to (?^2,\bot) \to (?^2,\bot) \to (?^2,s_2)\big[\to (?^2,\bot) \to (?^2,\bot)\big]^{s_2}\Big),$$

where $s_1 = \mathsf{p}(x \geq m) \oplus r_1 \in \{0,1\}$ for a uniform random bit r_1, $s_2 = r_1 \oplus s_1 \oplus r_2 \in \{0,1\}$ for a uniform random bit r_2, and the last two components "$\to (?^2,\bot) \to (?^2,\bot)$" appears only when $s_2 = 0$. It is equivalent to a probability distribution view^* defined as follows:

$$\mathsf{view}^* = \Big((?^2,\bot) \to (?^2,\bot) \to (?^2,r'_1) \to (?^2,\bot) \to (?^2,\bot) \to (?^2,\bot) \to (?^2,r'_2)\big[\to (?^2,\bot) \to (?^2,\bot)\big]^{r'_2}\Big),$$

where $r'_1, r'_2 \in \{0,1\}$ are uniform random bits and the last two components appears only when $r'_2 = 0$. Thus, for every $x, x' \in \mathbb{Z}_{2m}$, the following holds:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}}(s_{\mathsf{in}}(x)) = \mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}}(s_{\mathsf{in}}(x')) = \mathsf{view}^*.$$

Therefore, $\mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}$ securely realizes $\mathcal{F}^{\mathrm{d}}_{\mathrm{sv}}$.

Efficiency The number of cards is two. The number of subroutine calls is one (one call of the initialization protocol). From Proposition 1, a sign-to-value protocol without subroutines can be obtained. The number of probabilistic operations is three (one rotation shuffle, one two-sided rotation shuffle, and one flipping shuffle).

Carry Protocol

Functionality A functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{carry}}$ is defined as follows:

$$\mathcal{F}^{\mathrm{d}}_{\mathrm{carry}} = ([\![x_1]\!], [\![x_2]\!]) \;\Rightarrow\; ([\![\mathsf{p}(x_1 + x_2 \geq m)]\!], [\![0]\!]),$$

where $x_1, x_2 \in \mathbb{Z}_m$.

Protocol A protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{carry}}$ is defined as follows:

$$\mathcal{P}^{\mathrm{d}}_{\mathrm{carry}} = (2, \mathbb{Z}_m, (\mathsf{Dihedral}_{2m}, \{[\![0]\!], [\![0]\!]\}), \mathcal{O}^{\mathrm{d}}_{2m,2} \cup \mathsf{Subroutine}[\mathcal{P}^{\mathrm{d}}_{\mathrm{add}}, \mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}], A).$$

It proceeds as follows:

1. $(\mathsf{subroutine}, \mathcal{P}^{\mathrm{d}}_{\mathrm{add}}, \{1,2\})$: Apply the addition protocol in Sect. 4.2 to the sequence as follows:

$$([\![x_1]\!], [\![x_2]\!]) \;\to\; ([\![x_1 + x_2]\!], [\![0]\!]).$$

2. $(\mathsf{subroutine}, \mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}, \{1\})$: Apply the sign-to-value protocol in Sect. 4.4 to the first card as follows:

$$([\![x_1 + x_2]\!], [\![0]\!]) \ \rightarrow \ ([\![\mathsf{p}(x_1 + x_2 \geq m)]\!], [\![0]\!]).$$

Correctness The correctness is trivial.

Security Let $x = (x_1, x_2) \in (\mathbb{Z}_m)^2$ be any input. The probability distribution of a view of the protocol starting with the sequence $s_{\mathsf{in}}(x) = ([\![x_1]\!], [\![x_2]\!])$ is given as follows:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{carry}}}(s_{\mathsf{in}}(x)) = \big((?^2, \bot) \rightarrow (?^2, \bot) \rightarrow (?^2, \bot)\big).$$

It does not depend on x since it is just a fixed sequence. Thus, for every $x, x' \in (\mathbb{Z}_m)^2$, the following holds:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{carry}}}(s_{\mathsf{in}}(x)) = \mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{carry}}}(s_{\mathsf{in}}(x')).$$

Therefore, $\mathcal{P}^{\mathrm{d}}_{\mathrm{carry}}$ securely realizes $\mathcal{F}^{\mathrm{d}}_{\mathrm{carry}}$.

Efficiency The number of cards is two. The number of subroutine calls is two (one call of the addition protocol and one call of the sign-to-value protoocol). From Proposition 1, a carry protocol without subroutines can be obtained. The number of probabilistic operations is four (two rotation shuffles, one two-sided rotation shuffle, and one flipping shuffle).

Equality with Zero Protocol

Functionality A functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{zero}}$ is defined as follows:

$$\mathcal{F}^{\mathrm{d}}_{\mathrm{zero}} = ([\![x]\!], [\![0]\!]) \ \Rightarrow \ ([\![\mathsf{p}(x = 0)]\!], [\![0]\!]),$$

where $x \in \mathbb{Z}_m$.

Protocol A protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{zero}}$ is defined as follows:

$$\mathcal{P}^{\mathrm{d}}_{\mathrm{zero}} = (1, \mathbb{Z}_m, (\mathsf{Dihedral}_{2m}, \{[\![0]\!], [\![0]\!]\}), \mathcal{O}^{\mathrm{d}}_{2m,2} \cup \mathsf{Subroutine}[\mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}], A).$$

It proceeds as follows:

1. $(\mathsf{flip}, m, \{1\})$: Flip the first card along with the axis m as follows:

$$([\![x]\!], [\![0]\!]) \ \rightarrow \ ([\![m - x]\!], [\![0]\!]).$$

2. $(\mathsf{subroutine}, \mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}, \{1\})$: Apply the sign-to-value protocol in Sect. 4.4 to the first card as follows:

$$([\![m - x]\!], [\![0]\!]) \ \rightarrow \ ([\![s]\!], [\![0]\!]),$$

where $s = \mathsf{p}(m - x \geq m)$.

3. $(\mathsf{flip}, 1, \{1\})$: Flip the first card along with the axis 1 as follows:

$$([\![s]\!], [\![0]\!]) \ \rightarrow \ ([\![-s+1]\!], [\![0]\!]).$$

The protocol terminates.

Correctness For any $x \in \mathbb{Z}_m$, it holds $\mathsf{p}(m - x \geq m) = 0$ if and only if $x = 0$. Thus, the above protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{zero}}$ correctly realizes the functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{zero}}$.

Security Let $x \in \mathbb{Z}_m$ be any input. The probability distribution of a view of the protocol starting with the sequence $s_{\mathsf{in}}(x) = ([\![x]\!], [\![0]\!])$ is given as follows:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{zero}}}(s_{\mathsf{in}}(x)) = \big((?^2, \bot) \rightarrow (?^2, \bot) \rightarrow (?^2, \bot)\big).$$

It does not depend on x since it is just a fixed sequence. Thus, for every $x, x' \in (\mathbb{Z}_m)^2$, the following holds:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{zero}}}(s_{\mathsf{in}}(x)) = \mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{zero}}}(s_{\mathsf{in}}(x')).$$

Therefore, $\mathcal{P}^{\mathrm{d}}_{\mathrm{zero}}$ securely realizes $\mathcal{F}^{\mathrm{d}}_{\mathrm{zero}}$.

Efficiency The number of cards is two. The number of subroutine calls is one (one call of the sign-to-value protocol). From Proposition 1, an equality with zero protocol without subroutines can be obtained. The number of probabilistic operations is three (one rotation shuffle, one two-sided rotation shuffle, and one flipping shuffle).

Equality Protocol

Functionality A functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{equal}}$ is defined as follows:

$$\mathcal{F}^{\mathrm{d}}_{\mathrm{equal}} = ([\![x_1]\!], [\![x_2]\!]) \ \Rightarrow \ ([\![\mathsf{p}(x_1 = x_2)]\!], [\![0]\!]),$$

where $x_1, x_2 \in \mathbb{Z}_m$.

Protocol A protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{equal}}$ is defined as follows:

$$\mathcal{P}^{\mathrm{d}}_{\mathrm{equal}} = (2, \mathbb{Z}_m, (\mathsf{Dihedral}_{2m}, \{[\![0]\!], [\![0]\!]\}), \mathcal{O}^{\mathrm{d}}_{2m,2} \cup \mathsf{Subroutine}[\mathcal{P}^{\mathrm{d}}_{\mathrm{sub}}, \mathcal{P}^{\mathrm{d}}_{\mathrm{sign}}, \mathcal{P}^{\mathrm{d}}_{\mathrm{zero}}], A).$$

It proceeds as follows:

1. $(\mathsf{subroutine}, \mathcal{P}^{\mathrm{d}}_{\mathrm{sub}}, \{1\})$: Apply the subtraction protocol to the sequence as follows:
$$([\![x_1]\!], [\![x_2]\!]) \ \rightarrow \ ([\![x_2 - x_1]\!], [\![0]\!]).$$
2. $(\mathsf{subroutine}, \mathcal{P}^{\mathrm{d}}_{\mathrm{sign}}, \{1\})$: Apply the sign normalization protocol in Sect. 4.3 to the first card as follows:
$$([\![x_2 - x_1]\!], [\![0]\!]) \ \rightarrow \ ([\![z]\!], [\![0]\!]).$$
3. $(\mathsf{subroutine}, \mathcal{P}^{\mathrm{d}}_{\mathrm{zero}}, \{1, 2\})$: Apply the equality with zero protocol in Sect. 4.6 as follows:
$$([\![z]\!], [\![0]\!]) \ \rightarrow \ ([\![\mathsf{p}(z = 0)]\!], [\![0]\!]).$$

Correctness By the sign normalization protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{sign}}$, $z = x_2 - x_1 \bmod m$. Thus, the sequence $([\![z]\!], [\![0]\!])$ is matched with a subroutine of $\mathcal{P}^{\mathrm{d}}_{\mathrm{zero}}$. We can also observe that $z = 0$ if and only if $x_1 = x_2$. Thus, the above protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{equal}}$ correctly realizes the functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{equal}}$.

Security Let $x = (x_1, x_2) \in (\mathbb{Z}_m)^2$ be any input. The probability distribution of a view of the protocol starting with the sequence $s_{\mathsf{in}}(x) = ([\![x_1]\!], [\![x_2]\!])$ is given as follows:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{equal}}}(s_{\mathsf{in}}(x)) = \big((?^2, \bot) \to (?^2, \bot) \to (?^2, \bot) \to (?^2, \bot)\big).$$

It does not depend on x since it is just a fixed sequence. Thus, for every $x, x' \in (\mathbb{Z}_m)^2$, the following holds:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{equal}}}(s_{\mathsf{in}}(x)) = \mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{equal}}}(s_{\mathsf{in}}(x')).$$

Therefore, $\mathcal{P}^{\mathrm{d}}_{\mathrm{equal}}$ securely realizes $\mathcal{F}^{\mathrm{d}}_{\mathrm{equal}}$.

Efficiency The number of cards is two. The number of subroutine calls is three (one call of the subtraction protocol, one call of the sign normalization protocol, and one call of the equality with zero protocol). From Proposition 1, an equality protocol without subroutines can be obtained. The number of probabilistic operations is five (two rotation shuffles, two two-sided rotation shuffles, and one flipping shuffle).

Greater-than Protocol

Functionality A functionality $\mathcal{F}^{\mathrm{d}}_{\mathrm{gr}}$ is defined as follows:

$$\mathcal{F}^{\mathrm{d}}_{\mathrm{gr}} = ([\![x_1]\!], [\![x_2]\!]) \;\Rightarrow\; ([\![\mathsf{p}(x_2 \geq x_1)]\!], [\![0]\!]),$$

where $x_1, x_2 \in \mathbb{Z}_m$.

Protocol A protocol $\mathcal{P}^{\mathrm{d}}_{\mathrm{gr}}$ is defined as follows:

$$\mathcal{P}^{\mathrm{d}}_{\mathrm{gr}} = (2, \mathbb{Z}_m, (\mathsf{Dihedral}_{2m}, \{[\![0]\!], [\![0]\!]\}), \mathcal{O}^{\mathrm{d}}_{2m,2} \cup \mathsf{Subroutine}[\mathcal{P}^{\mathrm{d}}_{\mathrm{sub}}, \mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}], A).$$

It proceeds as follows:

1. $(\mathsf{subroutine}, \mathcal{P}^{\mathrm{d}}_{\mathrm{sub}}, \{1, 2\})$: Apply the subtraction protocol in Sect. 4.2 to the sequence as follows:

$$([\![x_1]\!], [\![x_2]\!]) \;\to\; ([\![x_2 - x_1]\!], [\![0]\!]).$$

2. $(\mathsf{subroutine}, \mathcal{P}^{\mathrm{d}}_{\mathrm{sv}}, \{1, 2\})$: Apply the sign-to-value protocol in Sect. 4.4 as follows:

$$([\![x_2 - x_1]\!], [\![0]\!]) \;\to\; ([\![1 - \mathsf{p}(x_2 \geq x_1)]\!], [\![0]\!]).$$

3. $(\mathsf{flip}, 1, \{1\})$: Flip the first card along with the axis 1 as follows:

$$([\![1 - \mathsf{p}(x_2 \geq x_1)]\!], [\![0]\!]) \;\to\; ([\![\mathsf{p}(x_2 \geq x_1)]\!], [\![0]\!]).$$

The protocol terminates.

Correctness The correctness is trivial.

Security Let $x = (x_1, x_2) \in (\mathbb{Z}_m)^2$ be any input. The probability distribution of a view of the protocol starting with the sequence $s_{\mathsf{in}}(x) = (\llbracket x_1 \rrbracket, \llbracket x_2 \rrbracket)$ is given as follows:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{gr}}}(s_{\mathsf{in}}(x)) = \big((?^2, \bot) \to (?^2, \bot) \to (?^2, \bot) \to (?^2, \bot)\big).$$

It does not depend on x since it is just a fixed sequence. Thus, for every $x, x' \in (\mathbb{Z}_m)^2$, the following holds:

$$\mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{gr}}}(s_{\mathsf{in}}(x)) = \mathsf{view}_{\mathcal{P}^{\mathrm{d}}_{\mathrm{gr}}}(s_{\mathsf{in}}(x')).$$

Therefore, $\mathcal{P}^{\mathrm{d}}_{\mathrm{gr}}$ securely realizes $\mathcal{F}^{\mathrm{d}}_{\mathrm{gr}}$.

Efficiency The number of cards is two. The number of subroutine calls is two (one call of the subtraction protocol and one call of the sign-to-value protocol). From Proposition 1, a greater than protocol without subroutines can be obtained. The number of probabilistic operations is four (two rotation shuffles, one two-sided rotation shuffle, and one flipping shuffle).

Conclusion and Future Work

In this paper, we designed a new type of cards, dihedral cards, with invisible ink, and constructed efficient protocols for various interesting predicates. We believe that the use of invisible ink makes it easier to design a new type of cards that enable to construct efficient secure computation protocols. An interesting research direction is to find such a new type of cards and objects, e.g., polyhedron.

Appendix

Definition for Regular Polygon Cards

We define the card specification of regular polygon cards. Regular polygon cards are also known as cyclic cards. Hereafter, we call them cyclic cards. The card specification of cyclic cards is given as follows.

For $x \in \mathbb{Z}_m$, we denote a face-up card having x by $\underline{x}$ and a face-down card having x by $[\![x]\!]$. The card set of cyclic cards of modulus m, denoted by $\mathcal{C}^{\mathsf{c}}_m$, is defined as follows:

$$\mathcal{C}^{\mathsf{c}}_m = \{\underline{0}, \underline{1}, \ldots, \underline{m-1}, [\![0]\!], [\![1]\!], \ldots, [\![m-1]\!]\}.$$

For a card $c \in \mathcal{C}^{\mathsf{c}}_m$, we define two types of transformations: rotation and turning. For any $j \in \mathbb{Z}_m$, a rotation operation with a degree j is defined as follows:

$$\mathsf{rot}^j(c) = \begin{cases} \underline{i+j} & \text{if } c = \underline{i} \text{ for some } i \in \mathbb{Z}_m \\ [\![i-j]\!] & \text{if } c = [\![i]\!] \text{ for some } i \in \mathbb{Z}_m \end{cases}$$

Physically, this is a rotation with $(360/m)°$. Note that a face-down card $[\![i]\!]$ is transformed into a face-down card $[\![i-j]\!]$ since a rotation of face-down cards is a backward rotation of face-up cards. A turning operation is defined as follows:

$$\mathsf{turn}(c) = \begin{cases} [\![i]\!] & \text{if } c = \underline{i} \text{ for some } i \in \mathbb{Z}_m \\ \underline{i} & \text{if } c = [\![i]\!] \text{ for some } i \in \mathbb{Z}_m \end{cases}$$

The transformation set of cyclic cards of modulus m, denoted by $\mathcal{T}^{\mathsf{c}}_m$, is defined as follows:

$$\mathcal{T}^{\mathsf{c}}_m = \{\mathsf{id}, \mathsf{rot}, \mathsf{rot}^2, \ldots, \mathsf{rot}^{m-1}, \mathsf{turn}\}.$$

The symbol set of cyclic cards of modulus m, denoted by Σ^{c}_m, is defined as follows:

$$\Sigma^{\mathsf{c}}_m = \{0, 1, 2, \ldots, m-1, \mathbf{?}\}.$$

The vision function $\mathsf{vis}^{\mathsf{c}}_m : \mathcal{C}^{\mathsf{c}}_m \to \Sigma^{\mathsf{c}}_m$ of cyclic cards of modulus m is defined as follows:

$$\mathsf{vis}^{\mathsf{c}}_m(c) = \begin{cases} i & \text{if } c = \underline{i} \text{ for } 0 \le i \le m-1 \\ \mathbf{?} & \text{otherwise.} \end{cases}$$

A card specification of cyclic cards of modulus m, denoted by Cyclic_m, is defined as follows:

$$\mathsf{Cyclic}_m = (\mathcal{C}^{\mathsf{c}}_m, \mathcal{T}^{\mathsf{c}}_m, \Sigma^{\mathsf{c}}_m, \mathsf{vis}^{\mathsf{c}}_m).$$

Operations for cyclic cards are defined similarly to operations for binary cards and dihedral cards. Specifically, permutations and turnings are defined almost the same as binary cards, and rotations and rotation shuffles are defined almost the same as dihedral cards.

References

1. Abe, Y., Hayashi, Y., Mizuki, T., Sone, H.: Five-card AND protocol in committed format using only practical shuffles. In: Proceedings of the 5th ACM on ASIA Public-Key Cryptography Workshop, APKC@AsiaCCS, Incheon, Republic of Korea, June 4, 2018, pp. 3–8 (2018). https://doi.org/10.1145/3197507.3197510
2. den Boer, B.: More efficient match-making and satisfiability: *The Five Card Trick*. In: Advances in Cryptology—EUROCRYPT '89, Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, April 10–13, 1989, Proceedings, pp. 208–217 (1989). https://doi.org/10.1007/3-540-46885-4_23
3. Cheung, E., Hawthorne, C., Lee, P.: Cs 758 project: Secure computation with playing cards (2013). https://csclub.uwaterloo.ca/~cdchawth/files/papers/secure_playing_cards.pdf
4. Crépeau, C., Kilian, J.: Discreet solitary games. In: Advances in Cryptology—CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings, pp. 319–330 (1993). https://doi.org/10.1007/3-540-48329-2_27
5. Kastner, J., Koch, A., Walzer, S., Miyahara, D., Hayashi, Y., Mizuki, T., Sone, H.: The minimum number of cards in practical card-based protocols. In: Advances in Cryptology—ASIACRYPT 2017—23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3–7, 2017, Proceedings, Part III, pp. 126–155 (2017). https://doi.org/10.1007/978-3-319-70700-6_5
6. Koch, A., Walzer, S., Härtel, K.: Card-based cryptographic protocols using a minimal number of cards. In: Advances in Cryptology—ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29–December 3, 2015, Proceedings, Part I, pp. 783–807 (2015). https://doi.org/10.1007/978-3-662-48797-6_32
7. Marcedone, A., Wen, Z., Shi, E.: Secure dating with four or fewer cards. Cryptology ePrint Archive, Report 2015/1031 (2015)
8. Mizuki, T.: Applications of card-based cryptography to education. IEICE Tech. Rep. **116**(289), 13–17 (2016). (In Japanese)
9. Mizuki, T., Sone, H.: Six-card secure AND and four-card secure XOR. In: Frontiers in Algorithmics, Third International Workshop, FAW 2009, Hefei, China, June 20-23, 2009. Proceedings, pp. 358–369 (2009). https://doi.org/10.1007/978-3-642-02270-8_36
10. Mizuki, T., Kumamoto, M., Sone, H.: The five-card trick can be done with four cards. In: Advances in Cryptology - ASIACRYPT 2012—18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2–6, 2012. Proceedings, pp. 598–606 (2012). https://doi.org/10.1007/978-3-642-34961-4_36
11. Mizuki, T., Uchiike, F., Sone, H.: Securely computing XOR with 10 cards. Austral. J. Combinator. **36**, 279–293 (2006)
12. Niemi, V., Renvall, A.: Secure multiparty computations without computers. Theor. Comput. Sci. **191**(1–2), 173–183 (1998). https://doi.org/10.1016/S0304-3975(97)00107-2
13. Shinagawa, K.: Card-based cryptography with invisible ink. In: T.V. Gopal, J. Watada (eds.) Theory and Applications of Models of Computation—15th Annual Conference, TAMC 2019, Kitakyushu, Japan, April 13-16, 2019, Proceedings, *Lecture Notes in Computer Science*, vol. 11436, pp. 566–577. Springer (2019). https://doi.org/10.1007/978-3-030-14812-6_35
14. Shinagawa, K., Mizuki, T., Schuldt, J.C.N., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Multi-party computation with small shuffle complexity using regular polygon cards. In: Provable Security—9th International Conference, ProvSec 2015, Kanazawa, Japan, November 24–26, 2015, Proceedings, pp. 127–146 (2015). https://doi.org/10.1007/978-3-319-26059-4_7
15. Shinagawa, K., Mizuki, T., Schuldt, J.C.N., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Card-based protocols using regular polygon cards. IEICE Transactions **100-A**(9), 1900–1909 (2017). http://search.ieice.org/bin/summary.php?id=e100-a_9_1900
16. Stiglic, A.: Computations with a deck of cards. Theor. Comput. Sci. **259**(1–2), 671–678 (2001). https://doi.org/10.1016/S0304-3975(00)00409-6