Single-Shuffle Full-Open Card-Based Protocols for Any Function

Reo Eriguchi[1*] and Kazumasa Shinagawa[2,3,1]

[1]National Institute of Advanced Industrial Science and Technology, Japan
[2]University of Tsukuba, Japan
[3]Kyushu University, Japan
[*]Corresponding author: eriguchi-reo@aist.go.jp

Abstract

A card-based secure computation protocol is a method for n parties to compute a function f on their private inputs $(x_1, \ldots, x_n)$ using physical playing cards, in such a way that the suits of revealed cards leak no information beyond the value of $f(x_1, \ldots, x_n)$. A *single-shuffle full-open* protocol is a minimal model of card-based secure computation in which, after the parties place face-down cards representing their inputs, a single shuffle operation is performed and then all cards are opened to derive the output. Despite the simplicity of this model, the class of functions known to admit single-shuffle full-open protocols has been limited to a few small examples. In this work, we prove for the first time that every function admits a single-shuffle full-open protocol. We present two constructions that offer a trade-off between the number of cards and the complexity of the shuffle operation. These feasibility results are derived from a novel connection between single-shuffle full-open protocols and a cryptographic primitive known as *Private Simultaneous Messages* protocols, which has rarely been studied in the context of card-based cryptography. We also present variants of single-shuffle protocols in which only a subset of cards are revealed. These protocols reduce the complexity of the shuffle operation compared to existing protocols in the same setting.

1 Introduction

To date, various cryptographic techniques have been proposed to achieve useful functionalities and security requirements. However, since these techniques are usually designed for implementation on computers, the algorithms tend to be complicated, leaving a gap in understanding for non-experts. *Card-based cryptography* [2, 3, 15] studies an implementation of cryptographic primitives using physical playing cards. This physical implementation helps make the functionality and security of cryptographic algorithms more accessible and easier to understand. In this work, we focus on binary cards with suits ♡ or ♣, which is the most standard setting in the literature.

Secure computation [33] has been studied actively in card-based cryptography. This primitive enables n parties each holding a private input x_i to evaluate a function f on $(x_1, \ldots, x_n)$ without revealing any extra information other than the value of $f(x_1, \ldots, x_n)$. Many works have proposed different card-based secure computation protocols for general and specific functions, mainly focusing on minimizing the number of cards and shuffle operations required (e.g., [10, 17, 26, 30]).

In this work, we consider a minimal model for card-based secure computation protocols, referred to as *single-shuffle full-open protocols* [27, 28]. These protocols take the following steps:

1. Each party places a sequence of face-down cards representing their private input x_i.
2. A random permutation π sampled from some probability distribution is applied to the entire sequence.
3. All cards are then turned face up, and the output is computed based on the sequence of revealed suits.

Table 1: Comparison of single-shuffle protocols

Protocol	Function	Opening	Shuffle	# cards
[2]	two-input AND	full-open	RC × 1	5
[25]	two-input XOR	full-open	RC × 1	4
[29]	three-input XOR	full-open	RC × 1	8
[6, 25]	three-input equality	full-open	RC × 1	6
[31]	three-input majority	full-open	RC × 1 & RBC × 1	6
[11]	n-input XOR	full-open	RBC × $(n-1)$	$2n$
[11]	n-input XOR	full-open	PSc × 2	$O(n \log n)$
Ours (Corollary 4.5)	any	full-open	PSh × 1 & CS × 1	$2^{O(n^3 2^{n/2})}$
Ours (Corollary 5.7)	any	full-open	PSh × $O(n2^n)$ & PSc × 1	$O(n^2 2^n)$
Ours (Corollary 5.5)	symmetric	full-open	PSh × n	$O(n^3 2^{2n})$
Ours (Corollary 5.4)	n-input AND	full-open	PSh × n	$O(n^2)$
[14]	two-input AND	adaptive-open	RC × 1 & RBC × 1	4
[11]	n-input AND	adaptive-open	RBC × $(n-1)$	$4n-2$
[11]	n-input AND	adaptive-open	PSc × 2	$O(n \log n)$
[19, 26, 32]	any	adaptive-open	PSc × $O(C_f + n)$	$O(C_f + n)$
Ours (Corollary 4.5)	any	static-open	PSh × 1	$2^{O(n^3 2^{n/2})}$
Ours (Corollary 4.5)	any	adaptive-open	RC × 1	$2^{O(n^3 2^{n/2})}$

n refers to the number of inputs. "CS" refers to a complete shuffle, "PSc" refers to a pile-scramble shuffle, "RC" refers to a random cut, "PSh" refers to a pile-shifting shuffle, and "RBC" refers to a random bisection cut, which is a pile-shifting shuffle performed on two piles of cards. C_f refers to the minimum number of gates in any circuit computing f.

The security requirement is that the revealed suits leak no information other than the value of $f(x_1, \ldots, x_n)$. This model assumes the minimum number of shuffle operations and the simplest opening operation (that is, turning over all cards).

Some feasibility results have been shown in this setting. The first card-based protocol, known as "Five-Card Trick" [2], is a single-shuffle full-open protocol that computes the two-input AND function. It is also known that several specific functions such as three-input equality, majority, and XOR functions can be computed by single-shuffle full-open protocols [6,11,25,29,31]. However, the fundamental question of whether every function $f : \{0,1\}^n \to \{0,1\}$ admits a single-shuffle full-open protocol remains open. Surprisingly, even the three-input AND function is not known to have a single-shuffle full-open protocol. Note that if a more advanced operation — namely, opening only a *subset* of cards — is allowed, then every function can be computed with a single shuffle [19,26,32]. However, it is not straightforward to extend such protocols to the full-open setting since revealing even a single additional card may compromise privacy.

1.1 Our Results

In this work, we prove that every function $f : \{0,1\}^n \to \{0,1\}$ admits a single-shuffle full-open protocol. Our constructions are based on a novel connection between single-shuffle full-open protocols and a cryptographic primitive known as *Private Simultaneous Messages* (PSM) protocols, which has rarely been studied in the context of card-based cryptography. We also present variants of single-shuffle protocols in which only a subset of cards are opened. These protocols reduce the complexity of the shuffle operation compared to the previous single-shuffle protocols [19, 26, 32]. We show a comparison of single-shuffle protocols in Table 1. Below, we elaborate on our results.

Single-Shuffle Full-Open Protocols for Any Function. We present two constructions of single-shuffle full-open protocols that offer a trade-off between the number of cards and the complexity of the shuffle operation. To describe our constructions, we introduce basic shuffle operations.

- A *complete shuffle* applies a uniformly random permutation to a sequence of cards.
- A *pile-scramble shuffle* divides a sequence of cards into multiple piles and applies a complete shuffle to these piles.
- A *random cut* applies a uniformly random shift to a sequence of cards.
- A *pile-shifting shuffle* divides a sequence of cards into piles and applies a random cut to the piles.

All our protocols use a composition of the above four shuffle operations[1]. We consider that a complete shuffle (resp. a random cut) is preferable to a pile-scramble (resp. a pile-shifting) shuffle since the latter requires additional items such as envelopes or sleeves to make piles.

For any function $f : \{0,1\}^n \to \{0,1\}$, our first construction requires $2^{O(n^3 2^{n/2})}$ cards and uses a composition of one pile-shifting shuffle and one complete shuffle. Our second construction reduces the number of cards to $O(n^2 2^n)$, but at the cost of a more complex shuffle: a composition of $O(n2^n)$ pile-shifting shuffles and one pile-scramble shuffle.

Techniques: Implication from Private Simultaneous Messages. Our constructions are based on a novel connection between single-shuffle full-open protocols and PSM protocols. This primitive was originally proposed by Feige et al. [4] as a non-interactive model of secure computation, in a setting unrelated to card-based cryptography. In this model, each party computes a message $m_i^{x_i}(\rho)$ based on their private input x_i and shared randomness $\rho \in \{0,1\}^r$, and then an external party can compute $f(x_1, \ldots, x_n)$ from the collection of messages $(m_i^{x_i}(\rho))_{i=1,\ldots,n}$. The security requirement is that the messages reveal no information beyond the output of f without the knowledge of the shared randomness ρ.

We present two transformations from PSM protocols to single-shuffle full-open protocols, corresponding to the above two constructions. In retrospect, both transformations are quite simple. In the first construction, we enumerate all possible collections of messages $M(\rho) = (m_i^{x_i}(\rho))_{i=1,\ldots,n}$ for every possible value of randomness $\rho \in \{0,1\}^r$. Each such $M(\rho)$ is encoded into a pile of cards, resulting in 2^r piles in total. A pile-shifting shuffle is applied to select a pile $M(\rho)$ corresponding to a uniformly random ρ, and then a complete shuffle is applied to the remaining piles to "erase" the other messages $M(\rho')$ for $\rho' \neq \rho$. Finally, all cards are revealed and $f(x_1, \ldots, x_n)$ is obtained by applying the decoding algorithm of the PSM protocol to the selected pile $M(\rho)$. Since every function admits a PSM protocol, this transformation leads to a single-shuffle full-open protocol for any function f. However, the number of piles is exponential in the randomness complexity r of the PSM protocol. Since the currently best known PSM protocol has $r = O(n^3 2^{n/2})$ [1], a doubly exponential number of cards $2^{O(n^3 2^{n/2})}$ are required.

We propose an alternative transformation applicable to a subclass of PSM protocols that satisfy a certain algebraic property, which we refer to as *additive PSM*. This transformation reduces the number of cards compared to our first transformation, which applies to arbitrary PSM protocols. Since several functions admit efficient additive PSM protocols [7,23], it derives single-shuffle full-open protocols with fewer cards for those functions including the AND function and any symmetric function[2]. In particular, we obtain a single-shuffle full-open protocol for the n-input AND function using $O(n^2)$ cards. We can extend this protocol to general functions, by leveraging the fact that any function can be computed by evaluating at most 2^n AND functions. This yields our second construction, which uses $O(n^2 2^n)$ cards. See Section 2 for more details.

Notably, a converse implication was shown in [28]: Any single-shuffle full-open protocol for a function f can be transformed into a PSM protocol for f. Combining the result of [28] with ours, we can relate the minimum number of cards required by any single-shuffle full-open protocol for f, to the minimum communication complexity and the minimum randomness complexity of any PSM protocol for f. See Section 6 for details.

Variants: Single-Shuffle Protocols with Partial Opening. We show variants of our first construction that reduce the complexity of the shuffle operation by relaxing the requirement that all cards are revealed. When a subset of cards are opened, we distinguish two types of opening: (1) *static opening*, where a fixed

[1] We treat any composition of shuffles as a single shuffle.

[2] We say that a function is symmetric if its output is independent of the order of inputs.

subset of cards are revealed; (2) *adaptive opening*, where the next card to reveal is determined by the suits of the cards that have already been revealed. We consider that static opening is preferable to the adaptive one in terms of simplicity.

Our full-open protocol requires a composition of a pile-shifting shuffle and a complete shuffle. The complete shuffle is required to "erase" information encoded in a specific subset of cards. The first variant reveals all cards except those in that subset, resulting in a single-shuffle static-open protocol for any function using only a pile-shifting shuffle.

Note that the pile-shifting shuffle is used to reveal one pile selected uniformly at random from multiple piles. We propose an adaptive-open protocol to realize such a functionality using a random cut and some additional cards, which may be of independent interest. As a result, we obtain another variant: a single-shuffle adaptive-open protocol for any function using a random cut. Both variants require asymptotically the same number of cards as our first construction, i.e., $2^{O(n^3 2^{n/2})}$.

We compare our protocols with the previous single-shuffle protocols in [19, 26, 32], which require adaptive opening. Our protocols reduce the complexity of the shuffle operation: the protocols in [19, 26, 32] require a composition of $O(C_f + n)$ pile-scramble shuffles, where C_f is the minimum number of gates in any circuit computing $f : \{0,1\}^n \to \{0,1\}$. As a trade-off, the number of cards required by our protocols is larger than those of the previous protocols [19, 26, 32] with $O(C_f + n)$ cards, which are upper bounded by $2^{O(n)}$ in the worst case.

1.2 Related Works

We provide a brief summary of existing works on card-based secure computation protocols.

Single-Shuffle Protocols. Prior to this work, the class of functions known to admit single-shuffle full-open protocols were limited to a few specific examples: two-input AND [2], multi-input XOR [11, 25, 29], three-input equality [6, 25], and three-input majority [31]. We are the first to prove that every function can be computed by a single-shuffle full-open protocol. The protocols in [19, 26, 32] can compute any function with a single shuffle but they require a more advanced operation when opening cards, namely adaptive opening. Furthermore, the complexity of the shuffle operation is sub-optimal: the protocols in [19, 26, 32] require a composition of many pile-scramble shuffles proportional to the number of gates plus the number of input wires in a circuit computing the function. In contrast, our single-shuffle adaptive-open protocol uses only one random cut.

Protocols Using More Than One Shuffles. There are card-based secure computation protocols that apply multiple rounds of shuffle and opening operations. In this model, Nishida, Mizuki, and Sone [17] showed that any function $f : \{0,1\}^n \to \{0,1\}$ can be computed using $O(2^n)$ rounds of shuffle and opening operations. The protocol achieves a constant number of cards (i.e., six) in addition to those representing inputs. Shinagawa and Nuida [26] also proposed a protocol for f using $O(C_f + n)$ cards and two rounds of shuffle and opening operations. For symmetric functions, tailor-made constructions by [17, 22, 30] reduce the number of operations and the number of cards. Several works have also focused on minimizing the number of cards and shuffle operations for small functions (e.g., [13, 18, 20, 21]).

Applications. Shinagawa and Nuida [28] showed that a certain type of card-based secure computation protocols can be transformed into PSM protocols for the same function. In particular, a single-shuffle full-open protocol using ℓ cards implies a PSM protocol with communication complexity ℓn and randomness complexity $\ell \log \ell + \ell n$. Prior to our work, this was the only known result demonstrating a connection between card-based protocols and PSM protocols. Our work complements their result by showing the converse implication from PSM protocols to single-shuffle full-open protocols. Card-based protocols have also found applications beyond secure computation such as in zero-knowledge proofs [5, 12, 16] and in simulating players of card games [24].

2 Overview of Our Techniques

We here provide an overview of our techniques. More detailed descriptions and proofs are given in the following sections.

2.1 Single-Shuffle Full-Open Protocols from PSM

We denote $[n] = \{1, \ldots, n\}$ for $n \in \mathbb{N}$. In this paper, we use binary cards whose front sides have a suit $\heartsuit$ or $\clubsuit$. A single-shuffle full-open protocol for a function $f : \{0,1\}^n \to \{0,1\}$ specifies $2n$ sequences of suits $(L_i^b)_{i\in[n],b\in\{0,1\}}$. Each party with input $x_i \in \{0,1\}$ puts face-down cards whose suits correspond to $L_i^{x_i}$ on the table. Then, a random permutation sampled from a probability distribution $\mathcal{D}$ is applied to the face-down cards, where the permutation is unknown to any party. Finally, parties turn over all cards and obtain an output $y = f(x_1, \ldots, x_n)$ from the revealed suits. The privacy requirement is that the sequence of the revealed suits leaks nothing about $(x_i)_{i\in[n]}$ beyond what follows from y. While we can extend the notion into the setting where only a subset of cards are turned over, we focus on protocols where all cards are opened in this section.

As mentioned earlier, our first construction is based on a cryptographic primitive known as PSM protocols [4, 8]. This primitive enables n parties each holding an input x_i to reveal $f(x_1, \ldots, x_n)$ to an external party called a referee without revealing any extra information. Specifically, it consists of three algorithms: A randomness generation algorithm Gen takes no input and outputs a random string ρ, which is shared among all parties except for the referee; An encoding algorithm Enc takes a party's index $i \in [n]$, a party's input x_i, and shared randomness ρ as inputs, and outputs a message μ_i, which is sent to the referee; A decoding algorithm Dec takes n messages $(\mu_i)_{i\in[n]}$ as input and outputs $y = f(x_1, \ldots, x_n)$. The privacy requirement is that the messages $(\mu_i)_{i\in[n]}$ reveal nothing about $(x_i)_{i\in[n]}$ other than $y = f(x_1, \ldots, x_n)$. We define randomness complexity as the bit-length of ρ, denoted by r, and communication complexity as the total bit-length of $(\mu_i)_{i\in[n]}$, denoted by c. To simplify the exposition, we here assume that ρ is uniformly distributed over $\{0,1\}^r$, while our full construction allows ρ to be sampled from a subset of $\{0,1\}^r$.

We follow an encoding rule $\heartsuit\clubsuit = 1$, $\clubsuit\heartsuit = 0$ and encode a bit string into a sequence of suits bit by bit. For example, a string 101 is encoded into $\heartsuit\clubsuit\clubsuit\heartsuit\heartsuit\clubsuit$. Let Σ be a PSM protocol for a function $f : \{0,1\}^n \to \{0,1\}$. For each $i \in [n]$, each $b \in \{0,1\}$, and each random string $\rho \in \{0,1\}^r$, we define a sequence of suits $m_i^b(\rho)$ as the encoding of a party i's message $\mathsf{Enc}(i, b, \rho)$ with input b and shared randomness ρ. Let L_i^b be a sequence of suits obtained by concatenating the $m_i^b(\rho)$'s over all possible ρ's. Here, we fix an order on the set of all ρ's (e.g., a lexicographical order) and enumerate them as $\{\rho_1, \ldots, \rho_{2^r}\}$.

Suppose that parties place face-down cards whose suits correspond to $L_1^{x_1}, \ldots, L_n^{x_n}$. This sequence can be rearranged into the concatenation of $M(\rho) := (m_1^{x_1}(\rho), \ldots, m_n^{x_n}(\rho))$ over all ρ's. Note that it suffices to open only the set of cards corresponding to $M(\rho)$ for a uniformly random ρ, since this is equivalent to revealing the messages that the referee receives in the PSM protocol Σ. We apply a pile-shifting shuffle to obtain a sequence of cards with suits $M(\rho_s), M(\rho_{s+1}), \ldots, M(\rho_{s-1})$, where s is a uniformly random shift. To eliminate all information except that of $M(\rho_s)$, we then apply a complete shuffle to the remaining cards $M(\rho_{s+1}), \ldots, M(\rho_{s-1})$. As a result, we obtain a sequence of face-down cards whose suits consist of $M(\rho_s)$ followed by a uniformly random arrangement of suits. Finally, we turn over all cards and compute $y = f(x_1, \ldots, x_n)$ by applying the decoding algorithm of Σ to the bit string $(\mathsf{Enc}(i, x_i, \rho_s))_{i\in[n]}$ encoded in $M(\rho_s)$. If the communication and randomness complexity of Σ are c and r, respectively, then the number of cards required to encode each $M(\rho)$ is $2c$ and hence the total number of cards is $2c \cdot 2^r = c2^{r+1}$. The state-of-the-art PSM protocol for a general function has communication and randomness complexity $O(n^3 2^{n/2})$ [1]. Therefore, the resulting single-shuffle full-open protocol uses $2^{O(n^3 2^{n/2})}$ cards.

2.2 Reducing the Number of Cards

Next, we show a different construction of single-shuffle full-open protocols that uses a smaller number of cards at the cost of requiring more complicated shuffles.

Expressing General Functions by AND Functions. Let $f : \{0,1\}^n \to \{0,1\}$ be a function and $f^{-1}(1) = \{\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(N)}\}$, where $\boldsymbol{a}^{(j)} = (a_1^{(j)}, \ldots, a_n^{(j)}) \in \{0,1\}^n$ is an input on which f evaluates to 1. For each $i \in [n]$, each $j \in [N]$, and each $x_i \in \{0,1\}$, we define $b_i^{(j)}(x_i) \in \{0,1\}$ as $b_i^{(j)}(x_i) = 1$ if and only if $x_i = a_i^{(j)}$. For any $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ with $f(\boldsymbol{x}) = 1$, there exists a unique $j \in [N]$ such that $\boldsymbol{x} = \boldsymbol{a}^{(j)}$; and if $f(\boldsymbol{x}) = 0$, then $\boldsymbol{x} \neq \boldsymbol{a}^{(j)}$ for all $j \in [N]$. Thus, it holds that

$$f(x_1, \ldots, x_n) = \sum_{j \in [N]} \mathsf{AND}_n(b_1^{(j)}(x_1), \ldots, b_n^{(j)}(x_n))$$

for any $(x_1, \ldots, x_n) \in \{0,1\}^n$, where AND_n is the n-input AND function, i.e., $\mathsf{AND}_n(y_1, \ldots, y_n) = 1$ if and only if $y_1 = \cdots = y_n = 1$.

Assume that we are given a single-shuffle full-open protocol Π_0 for AND_n. Then, we can construct a single-shuffle full-open protocol Π for f as follows. Each party i with input x_i computes bits $b_i^{(j)}(x_i)$ for all $j \in [N]$. Then, for each $j \in [N]$, parties execute Π_0 on inputs $(b_i^{(j)}(x_i))_{i \in [n]}$. That is, each party i places a sequence of face-down cards corresponding to $b_i^{(j)}(x_i)$ following the specification of Π_0, and then apply a shuffle specified by Π_0. At this point, if $f(\boldsymbol{x}) = 1$, then the resulting card sequence consists of a single pile of cards corresponding to an output 1 and $N-1$ piles of cards corresponding to an output 0, where the position of the unique pile is determined by $j \in [N]$ such that $\boldsymbol{a}^{(j)} = \boldsymbol{x}$. If $f(\boldsymbol{x}) = 0$, then the card sequence consists of N piles of cards corresponding to an output 0. Thus, we can determine $f(\boldsymbol{x})$ from the revealed cards. Note that before turning over cards, we need to apply a pile-scramble shuffle to ensure that the position of the pile corresponding to 1 is independent of $\boldsymbol{x}$. Then, the privacy of Π directly follows from that of Π_0. If Π_0 uses ℓ cards, then Π uses $\ell N \leq \ell 2^n$ cards.

If we instantiate Π_0 with our general protocol in the previous section, then $\ell = (n \log n) 2^{O(n \log n)} = 2^{O(n \log n)}$ since there exists a PSM protocol for AND_n with communication and randomness complexity $O(n \log n)$ [7, Claim 4.3]. Thus, the total number of cards is $2^{O(n \log n)} \cdot 2^n = 2^{O(n \log n)}$, which reduces the number of cards $2^{O(n^3 2^{n/2})}$ in our previous protocol. In the following, we further improve the number of cards by showing a single-shuffle full-open protocol tailored to computing AND_n that uses only $\ell = O(n^2)$ cards. This leads to a protocol for general f using $O(n^2 2^n)$ cards.

A Single-Shuffle Full-Open Protocol for AND_n. A key observation is that the PSM protocol for AND_n in [7] has a special algebraic structure, which we refer to as *additive PSM*. Let m be a prime, $\mathbb{Z}_m$ denote the ring of integers modulo m, and $\mathbb{Z}_m^*$ denote its multiplicative group. We call a PSM protocol additive over $\mathbb{Z}_m$ if shared randomness consists of an element u sampled uniformly at random from a subgroup $U \subseteq \mathbb{Z}_m^*$ and additive shares $(r_i)_{i \in [n]} \in \mathbb{Z}_m^n$ of zero, which means that $(r_i)_{i \in [n]}$ are sampled uniformly at random conditioned on $\sum_{i \in [n]} r_i = 0 \bmod m$. Furthermore, a message of each party i with input $x_i \in \{0,1\}$ has the form of $uy_i - r_i$, where $y_i \in \mathbb{Z}_m$ is determined by x_i. The output is computed based on $\sum_{i \in [n]}(uy_i - r_i) = u \sum_{i \in [n]} y_i$.

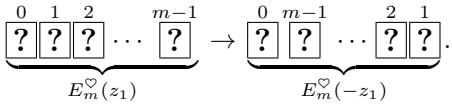
We show a construction of single-shuffle full-open protocols from additive PSM protocols. For several functions including AND_n, this can improve the number of cards in our previous construction from arbitrary PSM protocols. Our construction takes two steps: First, we transform n piles of face-down cards each encoding y_i into n piles each encoding $z_i := uy_i$ for a uniformly random element $u \in U$, which is unknown to any party. Second, we transform the resulting n piles into n piles each encoding $z_i - r_i \bmod m$, where $(r_i)_{i \in [n]}$ are random additive shares of zero. Here, we use a different encoding rule: $x \in \mathbb{Z}_m$ is encoded into a sequence $E_m^{\heartsuit}(x)$ of suits such that $\heartsuit$ is placed at x-th position and $\clubsuit$ is placed elsewhere, i.e.,

$$E_m^{\heartsuit}(x) := \overset{0}{\boxed{\clubsuit}} \cdots \overset{x}{\boxed{\heartsuit}} \cdots \overset{m-1}{\boxed{\clubsuit}}.$$

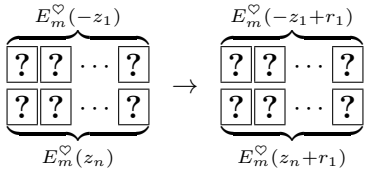
To describe the first procedure, we here assume for simplicity that $U = \mathbb{Z}_m^* = \{1, 2, \ldots, m-1\}$. Since m is a prime, U is a cyclic group with a generator α and we can write $U = \{\alpha, \alpha^2, \ldots, \alpha^{m-1}\}$. Each party with input x_i computes $y_i \in \mathbb{Z}_m$ and places face-down cards with suits $E_m^{\heartsuit}(y_i)$. For each $\gamma \in \mathbb{Z}_m$, let $c_{i,\gamma}$ denote the γ-th suit of $E_m^{\heartsuit}(y_i)$, that is, $c_{i,\gamma} = \heartsuit$ if and only if $\gamma = y_i$. We rearrange the face-down cards and

construct m piles $p(1), \ldots, p(m-1)$, where each $p(j)$ consists of the α^j-th cards of $E_m^{\heartsuit}(y_i)$ for all $i \in [n]$, i.e., $p(j) = (c_{i,\alpha^j})_{i\in[n]}$. We apply a pile-shifting shuffle to $p(1), \ldots, p(m-1)$ and denote the resulting piles by $p'(1), \ldots, p'(m-1)$. Then, there exists a uniformly random shift Δ such that $p'(j) = p(j-\Delta)$ for all j, where indices are modulo $m-1$. We define $c'_{i,0} = c_{i,0}$ and for $\gamma \neq 0$, define $c'_{i,\gamma}$ as the i-th card of $p'(\log_\alpha \gamma)$. Finally, we output $(c'_{1,\gamma})_{\gamma\in\mathbb{Z}_m}, \ldots, (c'_{n,\gamma})_{\gamma\in\mathbb{Z}_m}$ with all cards face down. Here, we define $\log_\alpha \gamma$ for $\gamma \in \mathbb{Z}_m \setminus \{0\}$ as j such that $\gamma = \alpha^j$. It holds that $c'_{i,\gamma} = c_{i,\alpha^{\log_\alpha \gamma - \Delta}} = c_{i,\gamma u^{-1}}$, where $u := \alpha^\Delta$. Thus, $c'_{i,\gamma} = \heartsuit$ if and only if $\gamma = y_i u$, which means that each output pile $(c'_{i,\gamma})_{\gamma\in\mathbb{Z}_m}$ is equal to $E_m^{\heartsuit}(y_i u)$. Since Δ is uniformly distributed over $\{1, \ldots, m-1\}$, u is uniformly distributed over $U = \mathbb{Z}_m^*$.

The second procedure transforms n piles of cards each with suits $E_m^{\heartsuit}(z_i)$ into n piles each with suits $E_m^{\heartsuit}(z_i - r_i)$, where $(r_i)_{i\in[n]}$ are random additive shares of zero. We first transform $E_m^{\heartsuit}(z_1)$ into $E_m^{\heartsuit}(-z_1)$ simply by reversing the order of the last $m-1$ cards:



We then pair two sequences $E_m^{\heartsuit}(-z_1)$ and $E_m^{\heartsuit}(z_n)$, and apply a pile-shifting shuffle:



Here, r_1 is a uniformly random shift. Finally, we reverse the order of the last $m-1$ cards of $E_m^{\heartsuit}(-z_1+r_1)$. As a result, we obtain two sequences $E_m^{\heartsuit}(z_1-r_1)$ and $E_m^{\heartsuit}(z_n+r_1)$. We apply the same procedure to a pair of $E_m^{\heartsuit}(z_2)$ and $E_m^{\heartsuit}(z_n+r_1)$, and obtain $E_m^{\heartsuit}(z_2-r_2)$ and $E_m^{\heartsuit}(z_n+r_1+r_2)$, where r_2 is another uniformly random shift. By continuing these procedures, we finally obtain $E_m^{\heartsuit}(z_1-r_1), \ldots, E_m^{\heartsuit}(z_n-r_n)$, where $r_1, \ldots, r_{n-1}$ are uniformly at random and $r_n = -\sum_{i=1}^{n-1} r_i$.

Now, we can transform an additive PSM protocol Σ for f into a single-shuffle full-open protocol Π for f. After performing the above two procedures, we obtain face-down cards with suits $E_m^{\heartsuit}(uy_1-r_1), \ldots, E_m^{\heartsuit}(uy_n-r_n)$, where u is a uniformly random element of U and $(r_i)_{i\in[n]}$ are random additive shares of zero. We turn over all cards and run the decoding algorithm of Σ on $(uy_i-r_i)_{i\in[n]}$. The correctness of Σ ensures that the output is $f(x_1, \ldots, x_n)$ and the privacy of Σ ensures that the revealed suits leak nothing but $f(x_1, \ldots, x_n)$. The number of cards in Π is mn. Since AND_n admits an additive PSM protocol over $\mathbb{Z}_m$ such that $m = O(n)$ [7], we obtain a single-shuffle full-open protocol for AND_n using $O(n^2)$ cards.

Putting It All Together. By plugging the protocol for AND_n into the above construction, we obtain a single-shuffle full-open protocol for any function $f : \{0,1\}^n \to \{0,1\}$ using at most $O(n^2) \cdot 2^n = O(n^2 2^n)$ cards. This improves upon our previous construction, which requires $2^{O(n^3 2^{n/2})}$ cards. However, it comes at the cost of more complicated shuffles: this protocol requires a composition of $O(n2^n)$ pile-shifting shuffles followed by one pile-scramble shuffle, while our previous protocol needs only one pile-shifting shuffle and one complete shuffle.

3 Preliminaries

Notations. For a natural number $n \in \mathbb{N}$, we denote $[n] = \{1, 2, \ldots, n\}$. Let $\mathfrak{S}_N$ denote the set of all permutations over $[N]$. For a probability distribution $\mathcal{D}$, we write $\rho \leftarrow \mathcal{D}$ if an element ρ is sampled according to $\mathcal{D}$. We denote the support of $\mathcal{D}$, i.e., the set of all possible outcomes, by $\mathrm{supp}(\mathcal{D})$. For a set X, we write $\rho \leftarrow\!\!\$\, X$ if an element ρ is sampled uniformly at random from X. For an integer $m \geq 2$, we denote the ring of integers modulo m by $\mathbb{Z}_m$ and identify it with $\{0, 1, \ldots, m-1\}$ as a set. Let $\mathbb{Z}_m^*$ denote the multiplicative

group of integers modulo m and naturally identify it with a subset of $\{0, 1, \ldots, m-1\}$. We call a function $f : \{0,1\}^n \to \{0,1\}$ symmetric if $f(x_{\sigma(1)}, \ldots, x_{\sigma(n)}) = f(x_1, \ldots, x_n)$ for any input $(x_i)_{i\in[n]} \in \{0,1\}^n$ and any permutation $\sigma \in \mathfrak{S}_n$. We define the n-input AND function $\mathsf{AND}_n : \{0,1\}^n \to \{0,1\}$ as $\mathsf{AND}_n(x_1, \ldots, x_n) = 1$ if and only if $x_1 = \cdots = x_n = 1$.

3.1 Card-Based Protocols

Cards. In this paper, we use *binary cards* whose front sides are either $\boxed{\clubsuit}$ or $\boxed{\heartsuit}$ and back sides are both $\boxed{?}$. We assume that two cards with the same suit are indistinguishable. For $x \in \mathbb{Z}_m$, let $E_m^{\heartsuit}(x) \in \{\heartsuit, \clubsuit\}^m$ denote the sequence of suits such that $\heartsuit$ is placed at x-th position and $\clubsuit$ is placed elsewhere.

Shuffles. A shuffle is an operation that applies a random permutation to a sequence of ℓ face-down cards, where the permutation is chosen from some probability distribution over $\mathfrak{S}_\ell$. We allow a shuffle to be applied not only to the entire sequence of cards but also to a subsequence of cards. We assume that no party guesses which permutation is chosen during the shuffle. Below, we introduce basic shuffles.

A *complete shuffle* is a shuffle that applies a uniformly random permutation to a sequence of ℓ face-down cards, which is denoted by $[\cdot]$. For example, a complete shuffle for a sequence of three cards results in one of the six sequences each with probability 1/6 as follows:

$$\left[\begin{smallmatrix}1&2&3\\ ?&?&?\end{smallmatrix}\right] \to \begin{smallmatrix}1&2&3\\ ?&?&?\end{smallmatrix} \text{ or } \begin{smallmatrix}1&3&2\\ ?&?&?\end{smallmatrix} \text{ or } \begin{smallmatrix}2&3&1\\ ?&?&?\end{smallmatrix} \text{ or } \begin{smallmatrix}2&1&3\\ ?&?&?\end{smallmatrix} \text{ or } \begin{smallmatrix}3&1&2\\ ?&?&?\end{smallmatrix} \text{ or } \begin{smallmatrix}3&2&1\\ ?&?&?\end{smallmatrix}.$$

A *pile-scramble shuffle* [9] is a shuffle that divides a sequence of cards into multiple *piles* of the same number of cards and applies a random permutation to a sequence of piles, which is denoted by $[\cdot|\cdots|\cdot]$. For example, a pile-scramble shuffle for a sequence of three piles each having two cards results in one of the six sequences each with probability 1/6 as follows:

$$\left[\begin{smallmatrix}1&2\\ ?&?\end{smallmatrix}\middle|\begin{smallmatrix}3&4\\ ?&?\end{smallmatrix}\middle|\begin{smallmatrix}5&6\\ ?&?\end{smallmatrix}\right] \to \left\{\begin{array}{llllll} \begin{smallmatrix}1&2\\ ?&?\end{smallmatrix} & \begin{smallmatrix}3&4\\ ?&?\end{smallmatrix} & \begin{smallmatrix}5&6\\ ?&?\end{smallmatrix} \\ \begin{smallmatrix}1&2\\ ?&?\end{smallmatrix} & \begin{smallmatrix}5&6\\ ?&?\end{smallmatrix} & \begin{smallmatrix}3&4\\ ?&?\end{smallmatrix} \\ \begin{smallmatrix}3&4\\ ?&?\end{smallmatrix} & \begin{smallmatrix}5&6\\ ?&?\end{smallmatrix} & \begin{smallmatrix}1&2\\ ?&?\end{smallmatrix} \\ \begin{smallmatrix}3&4\\ ?&?\end{smallmatrix} & \begin{smallmatrix}1&2\\ ?&?\end{smallmatrix} & \begin{smallmatrix}5&6\\ ?&?\end{smallmatrix} \\ \begin{smallmatrix}5&6\\ ?&?\end{smallmatrix} & \begin{smallmatrix}1&2\\ ?&?\end{smallmatrix} & \begin{smallmatrix}3&4\\ ?&?\end{smallmatrix} \\ \begin{smallmatrix}5&6\\ ?&?\end{smallmatrix} & \begin{smallmatrix}3&4\\ ?&?\end{smallmatrix} & \begin{smallmatrix}1&2\\ ?&?\end{smallmatrix} \end{array}\right. .$$

A *random cut* is a shuffle that applies a cyclic shift to a sequence of cards a uniformly random number of times, which is denoted by $\langle\cdot\rangle$. For example a random cut for a sequence of four cards results in one of the four sequences each with probability 1/4 as follows:

$$\left\langle\begin{smallmatrix}1&2&3&4\\ ?&?&?&?\end{smallmatrix}\right\rangle \to \begin{smallmatrix}1&2&3&4\\ ?&?&?&?\end{smallmatrix} \text{ or } \begin{smallmatrix}2&3&4&1\\ ?&?&?&?\end{smallmatrix} \text{ or } \begin{smallmatrix}3&4&1&2\\ ?&?&?&?\end{smallmatrix} \text{ or } \begin{smallmatrix}4&1&2&3\\ ?&?&?&?\end{smallmatrix}.$$

A *pile-shifting shuffle* is a shuffle that divides a sequence of cards into multiple piles of the same number of cards and applies a cyclic shift to a sequence of cards a uniformly random number of times, which is denoted by $\langle\cdot|\cdots|\cdot\rangle$. For example, a pile-shifting shuffle for a sequence of three piles each having two cards results in one of the three sequences each with probability 1/3 as follows:

$$\left\langle\begin{smallmatrix}1&2\\ ?&?\end{smallmatrix}\middle|\begin{smallmatrix}3&4\\ ?&?\end{smallmatrix}\middle|\begin{smallmatrix}5&6\\ ?&?\end{smallmatrix}\right\rangle \to \begin{smallmatrix}1&2&3&4&5&6\\ ?&?&?&?&?&?\end{smallmatrix} \text{ or } \begin{smallmatrix}3&4&5&6&1&2\\ ?&?&?&?&?&?\end{smallmatrix} \text{ or } \begin{smallmatrix}5&6&1&2&3&4\\ ?&?&?&?&?&?\end{smallmatrix}.$$

A complete shuffle is preferable to a pile-scramble shuffle since the latter requires envelopes or sleeves to make piles. For the same reason, a random cut is preferable to a pile-shifting shuffle. We consider that a complete shuffle and a random cut are incomparable.

A *composition* of two shuffles is an operation that applies two shuffles to a sequence consecutively. We regard any composition of shuffles as a single shuffle. Formally, let $\mathcal{F}_1, \mathcal{F}_2$ be the probability distributions of the first and the second shuffles, respectively. Then the probability distribution $\mathcal{F}$ of the composed shuffle is defined as

$$\Pr[\sigma \leftarrow \mathcal{F}] = \sum_{\substack{\sigma_1,\sigma_2:\\ \sigma_2\sigma_1=\sigma}} \Pr[\sigma_1 \leftarrow \mathcal{F}_1] \cdot \Pr[\sigma_2 \leftarrow \mathcal{F}_2].$$

Protocols. We follow the formalization of card-based protocols in [15]. Since we focus on protocols with a single shuffle, the formalization is simplified as follows.

Definition 3.1. *A* single-shuffle card-based protocol Π *for a function* $f : \{0,1\}^n \to \{0,1\}$ *is specified by* $2n$ *sequences of suits* $(L_i^b)_{i\in[n],b\in\{0,1\}}$ *with* $L_i^b \in \{\heartsuit,\clubsuit\}^{\ell_i}$*, a probability distribution* $\mathcal{D}$ *over* $\mathfrak{S}_\ell$*, and an index set* $R \subseteq [\ell]$ *of size* k*, where* $\ell = \sum_{i\in[n]} \ell_i$*. For an input* $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$*, a final state* $\mathsf{st}_{\boldsymbol{x}} \in \{\heartsuit,\clubsuit\}^k$ *of* Π *is defined as a sequence of suits obtained through the following procedures:*

1. *Prepare a sequence of* ℓ *face-down cards with suits* $(L_1^{x_1}, \ldots, L_n^{x_n})$*.*
2. *Sample a permutation* $\pi \leftarrow \mathcal{D}$ *and permute the sequence of the face-down cards according to* π*.*
3. *Reveal the cards at positions specified by the index set* R*.*
4. *Let* $\mathsf{st}_{\boldsymbol{x}}$ *be the sequence of suits of the revealed cards.*

Π *is required to satisfy the following properties:*

Correctness. *There exists a function* $\mathsf{Output} : \{\heartsuit,\clubsuit\}^k \to \{0,1\}$ *such that for any input* $\boldsymbol{x} \in \{0,1\}^n$*, it holds that* $\Pr[\mathsf{Output}(\mathsf{st}_{\boldsymbol{x}}) = f(\boldsymbol{x})] = 1$*, where the randomness is taken over the choice of* $\pi \leftarrow \mathcal{D}$*.*

Privacy. *For any pair of inputs* $\boldsymbol{x}, \boldsymbol{x}' \in \{0,1\}^n$ *with* $f(\boldsymbol{x}) = f(\boldsymbol{x}')$*, the distribution of* $\mathsf{st}_{\boldsymbol{x}}$ *is identical to that of* $\mathsf{st}_{\boldsymbol{x}'}$*.*

We define the number of cards *of* Π *as the minimum value of* $\ell_\heartsuit + \ell_\clubsuit$ *over all* $(\ell_\heartsuit, \ell_\clubsuit)$*'s such that for any* $(x_i)_{i\in[n]}$*, the sequence of suits* $(L_1^{x_1}, \ldots, L_n^{x_n})$ *contains at most* $\ell_\heartsuit$ $\heartsuit$*'s and* $\ell_\clubsuit$ $\clubsuit$*'s*[3]*. The* shuffle *of* Π *refers to the distribution* $\mathcal{D}$*.*

If $R = [\ell]$, then we say that Π is a single-shuffle *full-open* protocol. If R is a strict subset of $[\ell]$, then we call Π a single-shuffle *static-open* protocol. We also introduce a single-shuffle *adaptive-open* protocol, in which the position of the next card to be turned face up is determined based on the suits of the cards that have already been revealed. Formally, an adaptive-open protocol additionally specifies a function that takes a sequence of pairs $(p_i, s_i) \in [\ell] \times \{\heartsuit,\clubsuit\}$ of a position and a suit as input, and outputs the next position $p \in [\ell]$.

3.2 Private Simultaneous Messages Protocols

A Private Simultaneous Messages (PSM) protocol [4, 8] is a cryptographic primitive that enables n input parties each holding an input x_i to reveal $f(x_1, \ldots, x_n)$ to an external party without revealing any extra information. Below, we give the formal definition.

Definition 3.2. *A* Private Simultaneous Messages protocol Σ *(or a* PSM *protocol for short) for a function* $f : \{0,1\}^n \to \{0,1\}$ *is specified by three algorithms* Gen*,* Enc*, and* Dec*, where:*

[3] In all protocols presented in the paper, L_i^0 contains the same number of $\heartsuit$ (and hence the same number of $\clubsuit$) as L_i^1. In this case, the number of cards of Π equals $\ell = \sum_{i\in[n]} \ell_i$.

- $\mathsf{Gen}() \to \rho$: Gen *is a randomized algorithm that takes no input and outputs randomness* $\rho \leftarrow \mathcal{D}$*, where* $\mathcal{D}$ *is a probability distribution over* $\{0,1\}^r$*.*
- $\mathsf{Enc}(i, x_i, \rho) \to \mu_i$: Enc *is a deterministic algorithm that takes* $i \in [n]$*,* $x_i \in \{0,1\}$*, and* $\rho \in \{0,1\}^r$ *as input and outputs a message* $\mu_i \in \{0,1\}^{c_i}$*.*
- $\mathsf{Dec}(\mu_1, \ldots, \mu_n) \to y$: Dec *is a deterministic algorithm that takes* n *messages* $(\mu_i)_{i\in[n]}$ *as input and outputs* $y \in \{0,1\}$*.*

satisfying the following properties:

Correctness. *For any* $\boldsymbol{x} = (x_i)_{i\in[n]}$*, it holds that*

$$\Pr\left[\mathsf{Dec}((\mathsf{Enc}(i, x_i, \rho))_{i\in[n]}) = f(x_1, \ldots, x_n)\right] = 1,$$

where the probability is taken over the choice of $\rho \leftarrow \mathsf{Gen}()$*.*

Privacy. *For any* $\boldsymbol{x} = (x_i)_{i\in[n]}, \boldsymbol{x}' = (x'_i)_{i\in[n]}$ *with* $f(\boldsymbol{x}) = f(\boldsymbol{x}')$*, the distribution of* $(\mathsf{Enc}(i, x_i, \rho))_{i\in[n]}$ *induced by* $\rho \leftarrow \mathsf{Gen}()$ *is identical to that of* $(\mathsf{Enc}(i, x'_i, \rho'))_{i\in[n]}$ *induced by* $\rho' \leftarrow \mathsf{Gen}()$*.*

We define the randomness complexity *of* Σ *as* r *and the* communication complexity *of* Σ *as* $\sum_{i\in[n]} c_i$*.*

By default, we suppose that $\mathcal{D}$ is a uniform distribution over a subset of $\{0,1\}^r$.
The state-of-the-art PSM protocol for any function is given by [1].

Proposition 3.3 (Beimel et al. [1])**.** *For any function* $f : \{0,1\}^n \to \{0,1\}$*, there exists a PSM protocol for* f *with communication and randomness complexity* $O(n^3 2^{n/2})$*.*

For specific functions, PSM protocols with better communication and randomness complexity are known [7, 23]. These protocols have a common template. We refer to such special protocols as *additive* PSM protocols. Let m be a prime. We say that a PSM protocol $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is additive over $\mathbb{Z}_m$ if there are a subgroup $U \subseteq \mathbb{Z}_m^*$ and n functions $(g_i : \{0,1\} \to \mathbb{Z}_m)_{i\in[n]}$ such that

- Randomness generated by Gen is $\rho = (u, (r_i)_{i\in[n]})$, where u is sampled according to the uniform distribution over U and $(r_i)_{i\in[n]}$ are random additive shares of zero (i.e., uniformly random elements conditioned on $\sum_{i\in[n]} r_i = 0$).
- Each message has the form of $\mu_i = u \cdot g_i(x_i) - r_i \bmod m$.
- Dec computes an output based on $\sum_{i\in[n]} \mu_i \bmod m$.

The communication and randomness complexity is $O(n \log m)$.

Proposition 3.4 (Ishai [7])**.** *There exists an additive PSM protocol over* $\mathbb{Z}_m$ *for* AND_n *such that* m *is any prime larger than* n *and* $U = \mathbb{Z}_m^*$*. In particular, the communication and randomness complexity is* $O(n \log n)$*.*

Proposition 3.5 (Shinagawa et al. [23])**.** *For any symmetric function* $f : \{0,1\}^n \to \{0,1\}$*, there exists an additive PSM protocol over* $\mathbb{Z}_m$ *for* f *such that* $m = (1 + o(1))n^2 2^{2n-2}$ *for any sufficiently large* n *and* $U \subseteq \mathbb{Z}_m^*$ *is the subgroup of all quadratic residues a modulo* m*, i.e.,* $a \in \mathbb{Z}_m^*$ *for which there exists* $x \in \mathbb{Z}_m^*$ *such that* $x^2 = a \bmod m$*. In particular, the communication and randomness complexity is* $O(n^2)$*.*

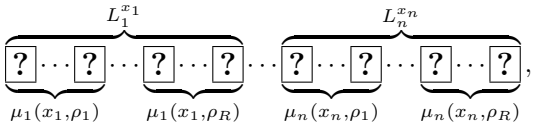
Notations.

- Let $f : \{0,1\}^n \to \{0,1\}$ be a function.
- Let $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PSM protocol for f with randomness complexity r and communication complexity c.
- Let $\mathrm{supp}(\mathcal{D}) = \{\rho_1, \ldots, \rho_R\}$ be an enumeration of all possible values of randomness for Σ.

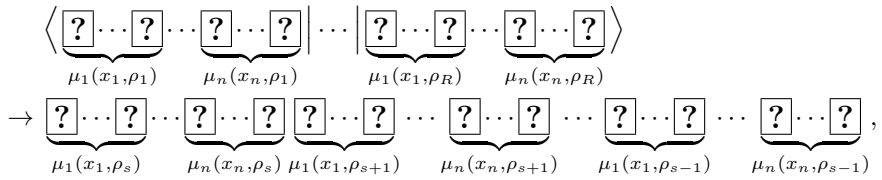
Construction of L_i^b. For each $i \in [n]$ and each $b \in \{0,1\}$,

1. Denote $\mu_i(b, \rho) = \mathsf{Enc}(i, b, \rho) \in \{0,1\}^{c_i}$ for each $\rho \in \mathrm{supp}(\mathcal{D})$.
2. Let $\widetilde{L}_i^b = \mu_i(b, \rho_1) || \cdots || \mu_i(b, \rho_R) \in \{0,1\}^{c_i R}$, where $||$ denotes the concatenation of strings.
3. Let $L_i^b \in \{\heartsuit, \clubsuit\}^{\ell_i}$ be the sequence of suits determined by $\widetilde{L}_i^b$ based on the encoding rule $\clubsuit\heartsuit = 0$ and $\heartsuit\clubsuit = 1$, where $\ell_i = 2c_i R$.

Shuffling. Given a sequence of $\ell = \sum_{i\in[n]} \ell_i$ face-down cards

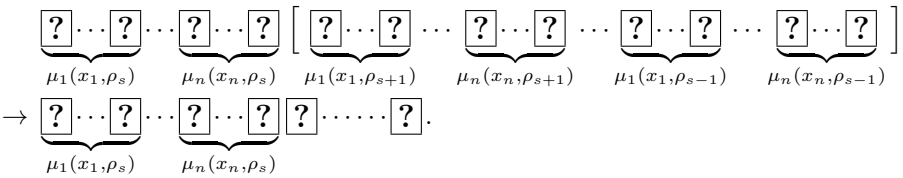


1. Apply a pile-shifting shuffle as follows:



   where $s \in [R]$.

2. Apply a complete shuffle to the sequence of cards, except for those corresponding to $(\mu_1(x_1, \rho_s), \ldots, \mu_n(x_n, \rho_s))$. That is,



Output.

1. Reveal all face-down cards.
2. Run Dec on input $(\mu_1(x_1, \rho_s), \ldots, \mu_n(x_n, \rho_s))$ to obtain $y \in \{0,1\}$.
3. Output y.

Figure 1: A single-shuffle full-open protocol based on a PSM protocol

4 Single-Shuffle Protocols for Any Function

First, we show a transformation from a PSM protocol for a function f into a single-shuffle full-open protocol for f using a composition of a pile-shifting shuffle and a complete shuffle.

Theorem 4.1. *Let $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PSM protocol for a function $f : \{0,1\}^n \to \{0,1\}$ with randomness complexity r and communication complexity c. Then, there exists a single-shuffle full-open protocol Π for f such that*

- *The number of cards is at most $c2^{r+1}$.*
- *The shuffle is a composition of a pile-shifting shuffle and a complete shuffle.*

Proof. Let Π be a card-based protocol described in Figure 1. The correctness of Π follows straightforwardly from that of the PSM protocol Σ since

$$\mathsf{Dec}((\mu_i(x_i, \rho_s))_{i\in[n]}) = \mathsf{Dec}((\mathsf{Enc}(i, x_i, \rho_s))_{i\in[n]}) = f((x_i)_{i\in[n]}).$$

To see privacy, let $\boldsymbol{x} = (x_i)_{i\in[n]}$ and $\boldsymbol{x}' = (x'_i)_{i\in[n]}$ be inputs with $f(\boldsymbol{x}) = f(\boldsymbol{x}')$. Since ρ_s at Step 1 of the shuffling phase is uniformly sampled from $\{\rho_1, \ldots, \rho_R\}$, the privacy of Σ ensures that the distribution of $(\mu_i(x_i, \rho_s))_{i\in[n]}$ is identical to that of $(\mu_i(x'_i, \rho_s))_{i\in[n]}$. Furthermore, a complete shuffle at Step 2 ensures that the suits except for those corresponding to $(\mu_i(x_i, \rho_s))_{i\in[n]}$ or $(\mu_i(x'_i, \rho_s))_{i\in[n]}$ are equally distributed, independent of $\boldsymbol{x}$ or $\boldsymbol{x}'$. Therefore, we conclude that the distribution of suits revealed in the output phase on input $\boldsymbol{x}$ is identical to that on input $\boldsymbol{x}'$.

Since we use the encoding rule $\heartsuit\clubsuit = 1$ and $\clubsuit\heartsuit = 0$, the number of cards to encode $(\mu_i(x_i, \rho))_{i\in[n]}$ is $2c$. Thus, the total number of cards is $2cR \leq c2^{r+1}$. □

It is possible to extend the above construction to the setting where $\mathcal{D}$ is an arbitrary distribution over $\{0,1\}^r$. At Step 1 in the shuffling phase, we sample a non-uniform random shift $s \in [R]$ according to $\mathcal{D}$, that is, s is sampled with probability $\Pr_{\mathcal{D}}[\rho_s]$.

Next, we show variants of the construction in Theorem 4.1, which use simpler shuffles at the price of giving up full-opening. The first variant modifies the construction in such a way that only the pile of cards corresponding to $p_s := (\mu_i(x_i, \rho_s))_{i\in[n]}$ are revealed at Step 1 in the output phase. Then, it is no longer necessary to apply a complete shuffle to the remaining cards at Step 2 of the shuffling phase. Since the positions of cards to be revealed are fixed, this leads to a single-shuffle static-open protocol that only requires a pile-shifting shuffle.

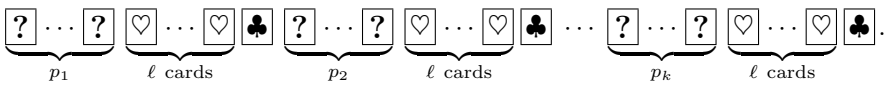
Theorem 4.2. *Let Σ be a PSM protocol for a function $f : \{0,1\}^n \to \{0,1\}$ with randomness complexity r and communication complexity c. Then, there exists a single-shuffle static-open protocol Π for f such that*

- *The number of cards is $c2^{r+1}$.*
- *The shuffle is a pile-shifting shuffle.*

As for the second variant, we observe that at Step 1 of the shuffling phase in Figure 1, a pile-shifting shuffle is applied to the sequence to select a pile p_s uniformly at random from the sequence of piles $(p_1, \ldots, p_R)$. We refer to this functionality as *uniform selection*. With some additional cards, uniform selection can be implemented using a random cut only. This result is formalized in the following lemma.

Lemma 4.3. *Let $(p_1, p_2, \ldots, p_k)$ be a sequence of k piles of cards, each consisting of the same number of cards. Suppose that p_i does not contain ℓ consecutive $\heartsuit$s as a subsequence for any $i \in [k]$. Then, there exists an adaptive-open protocol that implements a uniform selection from $(p_1, p_2, \ldots, p_k)$ by using a random cut and $(\ell + 1)k$ additional cards.*

Proof. The protocol additionally uses ℓk $\boxed{\heartsuit}$s and k $\boxed{\clubsuit}$s. Using these $(\ell+1)k$ additional binary cards, we first arrange the sequence as follows:



We refer to each sequence of ℓ $\boxed{\heartsuit}$s followed by one $\boxed{\clubsuit}$ as a *delimiter*. Next, we turn all delimiters face-down and apply a random cut to the entire sequence. Then, we reveal the cards from left to right until we encounter a delimiter of the form $\boxed{\heartsuit}\cdots\boxed{\heartsuit}\boxed{\clubsuit}$. Once a delimiter is found, we output the pile immediately to its left. (Note that each pile has a fixed number of cards.) Since each p_i does not contain ℓ consecutive $\heartsuit$s, we can always find such a delimiter. Thus, the above procedure implements a uniform selection. □

Since the encoding rule $\clubsuit\heartsuit = 0, \heartsuit\clubsuit = 1$ ensures that each pile p_s does not contain $\ell = 3$ consecutive $\heartsuit$s, uniform selection from $(p_1, \ldots, p_R)$ can be implemented with $4R \leq 4 \cdot 2^r = 2^{r+2}$ additional cards. We thus obtain a single-shuffle adaptive-open protocol using a random cut and at most 2^{r+2} additional cards.

Theorem 4.4. *Let Σ be a PSM protocol for a function $f : \{0,1\}^n \to \{0,1\}$ with randomness complexity r and communication complexity c. Then, there exists a single-shuffle adaptive-open protocol Π for f such that*

- *The number of cards is $c2^{r+1} + 2^{r+2}$.*
- *The shuffle is a random cut.*

We apply the above theorems to the PSM protocol in Proposition 3.3 and obtain the following corollary.

Corollary 4.5. *For any function $f : \{0,1\}^n \to \{0,1\}$, there exist three single-shuffle protocols Π_1, Π_2, and Π_3 for f such that*

- *Π_1 is full-open and uses a composition of a pile-shifting shuffle and a complete shuffle;*
- *Π_2 is static-open and uses a pile-shifting shuffle;*
- *Π_3 is adaptive-open and uses a random cut;*

and all of them require $2^{O(n^3 2^{n/2})}$ cards.

5 Single-Shuffle Full-Open Protocols with Fewer Cards

We show a different single-shuffle full-open protocol reducing the number of cards from $2^{O(n^3 2^{n/2})}$ to $O(n^2 2^n)$ at the cost of using a composition of $O(n2^n)$ shuffles.

5.1 A Construction Based on Additive PSM

First, we present a transformation from additive PSM protocols to single-shuffle full-open protocols. Recall that in an additive PSM protocol over $\mathbb{Z}_m$, each message has the form of $uy_i - r_i \bmod m$, where u is sampled uniformly at random from a subgroup $U \subseteq \mathbb{Z}_m^*$, $y_i = g_i(x_i)$ is determined by an input x_i, and $(r_i)_{i\in[n]}$ are random additive shares of zero.

The following proposition shows a card-based protocol for multiplying each y_i by a common u.

Proposition 5.1. *Continuing the notations in Figure 2, for any $(y_i)_{i\in[n]} \in \mathbb{Z}_m^n$, a sequence of face-down cards with suits $(E_m^{\heartsuit}(y_1), \ldots, E_m^{\heartsuit}(y_n))$ is transformed into a sequence with suits $(E_m^{\heartsuit}(uy_1), \ldots, E_m^{\heartsuit}(uy_n))$, where $u \leftarrow_{\$} U$. The procedures involve one pile-shifting shuffle.*

Notations.

- Let m be a prime.
- Let $U = \{\alpha_0, \ldots, \alpha_{k-1}\}$ be a subgroup of $\mathbb{Z}_m^*$. Since U is necessarily a cyclic group, we may assume that α_0 is a generator and $\alpha_\ell = \alpha_0^{\ell+1}$ for any $\ell \in \{0, 1, \ldots, k-1\}$.
- Let $\beta_1, \ldots, \beta_t \in \mathbb{Z}_m^*$ be coset representatives with respect to U, i.e., $\mathbb{Z}_m^* = \bigcup_{s\in[t]} \beta_s U$, where $t = (m-1)/k$.

Input. mn face-down cards with suits $(E_m^{\heartsuit}(y_1), \ldots, E_m^{\heartsuit}(y_n))$, where $y_i \in \mathbb{Z}_m$ and $E_m^{\heartsuit}(y_i) = (c_{i,0}, \ldots, c_{i,m-1}) \in \{\heartsuit, \clubsuit\}^m$.

Output. mn face-down cards with suits $(E_m^{\heartsuit}(uy_1), \ldots, E_m^{\heartsuit}(uy_n))$, where $u \leftarrow_{\$} U$.

Procedures.

1. Rearrange the input sequence and obtain m piles $(p(0), \ldots, p(m-1))$, where
$$p(\gamma) = \underset{c_{1,\gamma}}{\boxed{?}} \cdots \underset{c_{n,\gamma}}{\boxed{?}}.$$
2. Rearrange the sequence of $(p(\gamma))_{\gamma\in\mathbb{Z}_m^*}$ and obtain k piles $(q(0), \ldots, q(k-1))$, where
$$q(\ell) = \underbrace{\boxed{?}\cdots\boxed{?}}_{p(\beta_1\alpha_\ell)} \cdots \underbrace{\boxed{?}\cdots\boxed{?}}_{p(\beta_t\alpha_\ell)}.$$
3. Apply a pile-shifting shuffle as follows:
$$\left\langle \underbrace{\boxed{?}\cdots\boxed{?}}_{q(0)} \middle| \cdots \middle| \underbrace{\boxed{?}\cdots\boxed{?}}_{q(k-1)} \right\rangle \rightarrow \underbrace{\boxed{?}\cdots\boxed{?}}_{q(\Delta)} \cdots \underbrace{\boxed{?}\cdots\boxed{?}}_{q(k-1+\Delta)},$$
where $\Delta \leftarrow_{\$} \mathbb{Z}_k$ and indices are taken modulo k.
4. Set $q'(\ell) := q(\ell + \Delta)$ for each $\ell \in \mathbb{Z}_k$ and parse it as
$$q'(\ell) = \underbrace{\boxed{?}\cdots\boxed{?}}_{p(\gamma_{1,\ell})} \cdots \underbrace{\boxed{?}\cdots\boxed{?}}_{p(\gamma_{t,\ell})}.$$
5. Rearrange the sequence of $p(0)$ and $(q'(\ell))_{\ell\in\mathbb{Z}_k}$ and obtain m piles $(p'(\gamma))_{\gamma\in\mathbb{Z}_m}$ as follows: Let $p'(0) = p(0)$. For each $\gamma \in \mathbb{Z}_m^*$, express $\gamma = \beta_s\alpha_\ell$ for some $s \in [t]$ and $\ell \in \mathbb{Z}_k$. Set $p'(\gamma) = p(\gamma_{s,\ell})$, which is the s-th pile of $q'(\ell)$.
6. Let $(c'_{i,\gamma})_{i\in[n],\gamma\in\mathbb{Z}_m}$ be cards specified as follows:
$$\overbrace{\underset{c'_{1,0}}{\boxed{?}}\cdots\underset{c'_{n,0}}{\boxed{?}}}^{p'(0)} \cdots \overbrace{\underset{c'_{1,m-1}}{\boxed{?}} \cdots \underset{c'_{n,m-1}}{\boxed{?}}}^{p'(m-1)}.$$
7. Output $((c'_{1,\gamma})_{\gamma\in\mathbb{Z}_m}, \ldots, (c'_{n,\gamma})_{\gamma\in\mathbb{Z}_m})$.

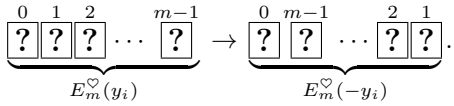
Figure 2: A card-based protocol for multiplying inputs by $u \leftarrow_{\$} U$

Input. mn face-down cards with suits $(E_m^{\heartsuit}(y_1), \ldots, E_m^{\heartsuit}(y_n))$, where $y_i \in \mathbb{Z}_m$ and $E_m^{\heartsuit}(y_i) = (c_{i,0}, \ldots, c_{i,m-1}) \in \{\heartsuit, \clubsuit\}^m$.

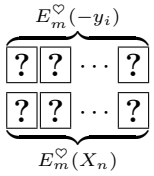
Output. mn face-down cards with suits $(E_m^{\heartsuit}(y_1 - r_1), \ldots, E_m^{\heartsuit}(y_n - r_n))$, where $(r_i)_{i \in [n]}$ are random additive shares of zero.

Procedures. Let X_n be a variable on $\mathbb{Z}_m$. Initially, set $X_n \leftarrow y_n$. For each $i = 1, 2, \ldots, n-1$, do the following procedures.

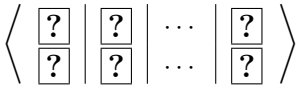
1. Reverse the order of the integer encoding $E_m^{\heartsuit}(y_i)$ except the 0-th card as follows:



2. Place $E_m^{\heartsuit}(X_n)$ on the bottom as follows:



3. Apply a pile-shifting shuffle as follows:



Here, the upper sequence is $E_m^{\heartsuit}(-y_i + r_i)$ and the bottom sequence is $E_m^{\heartsuit}(X_n + r_i)$ for $r_i \leftarrow_{\$} \mathbb{Z}_m$.

4. Reverse the order of the upper sequence and obtain $E_m^{\heartsuit}(y_i - r_i)$. Then we have $E_m^{\heartsuit}(y_i - r_i)$ and $E_m^{\heartsuit}(X_n + r_i)$. Update $X_n \leftarrow X_n + r_i$ and proceed to the next iteration.

Figure 3: A card-based protocol for adding random additive shares of zeros to inputs

Proof. Let $(y_i)_{i \in [n]} \in \mathbb{Z}_m^n$. Let $d_i = c_{i,y_i}$ denote the unique suit $\heartsuit$ contained in $E_m^{\heartsuit}(y_i)$. After Step 1, for each $i \in [n]$, the suit d_i is moved to the i-th position of the pile $p(y_i)$.

For $i \in [n]$ with $y_i = 0$, d_i remains in the pile $p(0)$ after Step 3. Hence, d_i is placed at the i-th position of $p'(0)$, i.e., $c'_{i,0} = d_i = \heartsuit$ and $c'_{i,\gamma} = \clubsuit$ for all $\gamma \neq 0$.

For $i \in [n]$ with $y_i \neq 0$, let $s_i \in [t]$ and $\ell_i \in \mathbb{Z}_k$ be such that $y_i = \beta_{s_i} \alpha_{\ell_i}$. Note that at Step 4, $p(\gamma_{s,\ell})$ is the s-th pile of $q'(\ell) = q(\ell + \Delta)$ and $\gamma_{s,\ell} = \beta_s \alpha_{\ell+\Delta} = \beta_s \alpha_0^{\ell+\Delta+1} = \beta_s \alpha_\ell \alpha_0^{\Delta}$. Thus, $\gamma_{s,\ell} = y_i$ if and only if $s = s_i$ and $\alpha_\ell = \alpha_{\ell_i} \alpha_0^{-\Delta}$, i.e., $\ell = \ell_i - \Delta \bmod k$. This means that the suit d_i is moved to the s_i-th pile $p(\gamma_{s_i,\ell_i-\Delta})$ of $q'(\ell_i - \Delta)$. At Step 5, d_i is contained in $p'(\gamma)$ with $\gamma = \gamma_{s_i,\ell_i-\Delta} = uy_i$, where $u = \alpha_0^{-\Delta}$. Hence, $c'_{i,uy_i} = d_i = \heartsuit$ and $c'_{i,\gamma} = \clubsuit$ for any $\gamma \neq uy_i$.

In summary, $(c'_{i,\gamma})_{\gamma \in \mathbb{Z}_m}$ is identical to $E_m^{\heartsuit}(uy_i)$ for all $i \in [n]$. Since Δ is uniformly distributed over $\mathbb{Z}_k$, $u = \alpha_0^{-\Delta}$ is uniformly distributed over U. □

The following proposition shows a card-based protocol for adding random additive shares $(r_i)_{i \in [n]}$ of zero to $(y_i)_{i \in [n]}$.

Proposition 5.2. *Continuing the notations in Figure 3, for any $(y_i)_{i\in[n]} \in \mathbb{Z}_m^n$, a sequence of face-down cards with suits $(E_m^\heartsuit(y_1), \ldots, E_m^\heartsuit(y_n))$ is transformed into a sequence with suits $(E_m^\heartsuit(y_1 - r_1), \ldots, E_m^\heartsuit(y_n - r_n))$, where $(r_i)_{i\in[n]}$ are random additive shares of zero. The procedures involve $n-1$ pile-shifting shuffles.*

Proof. Let $i = 1$. After the first iteration, we obtain two piles with suits $E_m^\heartsuit(y_1 - r_1)$ and $E_m^\heartsuit(X_n + r_1) = E_m^\heartsuit(y_n + r_1)$, where $r_1 \leftarrow_\$ \mathbb{Z}_m$ is a random element resulting from a pile-shifting shuffle at Step 3.

Let $i \geq 1$. Assume that after the i-th iteration, we obtain $i+1$ piles with suits $E_m^\heartsuit(y_1 - r_1), \ldots, E_m^\heartsuit(y_i - r_i), E_m^\heartsuit(y_n + R_i)$, where $r_1, \ldots, r_i \leftarrow_\$ \mathbb{Z}_m$ are independent random elements and $R_i = \sum_{j=1}^{i} r_j$. Then, in the $(i+1)$-th iteration, we have $X_n = y_n + R_i$. After Steps 3 and 4, we obtain two piles with suits $E_m^\heartsuit(y_{i+1} - r_{i+1})$ and $E_m^\heartsuit(x_n + r_{i+1}) = E_m^\heartsuit(y_n + R_i + r_{i+1})$. We thus obtain $i+2$ piles with suits $E_m^\heartsuit(y_1 - r_1), \ldots, E_m^\heartsuit(y_i - r_i), E_m^\heartsuit(y_{i+1} - r_{i+1}), E_m^\heartsuit(y_n + R_{i+1})$, where $R_{i+1} = \sum_{j=1}^{i+1} r_j$.

By induction on i, we finally obtain n piles with suits $E_m^\heartsuit(y_1 - r_1), \ldots, E_m^\heartsuit(y_{n-1} - r_{n-1}), E_m^\heartsuit(y_n - r_n)$, where $r_1, \ldots, r_{n-1} \leftarrow_\$ \mathbb{Z}_m$ are independent random elements and $r_n = -R_n = -\sum_{j=1}^{n-1} r_j$. □

Finally, by combining the above protocols, we construct a single-shuffle full-open protocol for f from an additive PSM protocol for f.

Theorem 5.3. *Let $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an additive PSM protocol over $\mathbb{Z}_m$ for a function $f : \{0,1\}^n \to \{0,1\}$. Then, there exists a single-shuffle full-open protocol Π for f such that*

- *The number of cards is mn.*
- *The shuffle is a composition of n pile-shifting shuffles.*

Proof. The protocol Π is described in Figure 4. At Step 2 in the output phase, we have that $\mu_i = u \cdot g_i(x_i) - r_i$ for each $i \in [n]$. Since $u \in U$ and $\sum_{i\in[n]} r_i = 0$, Dec correctly outputs $y = f(x_1, \ldots, x_n)$.

Furthermore, u is uniformly distributed over U and $(r_i)_{i\in[n]}$ are random additive shares of zero. Thus, the distribution of $(\mu_i)_{i\in[n]}$ is identical to that of messages of the additive PSM protocol Σ. The privacy of Σ ensures that $(\mu_i)_{i\in[n]}$ and hence the suits of revealed cards depend only on $f(x_1, \ldots, x_n)$. □

It is known that there exists an additive PSM protocol over $\mathbb{Z}_m$ for any function $f : \{0,1\}^n \to \{0,1\}$ if $m = 2^{O(2^n)}$ [7]. Although it is possible to apply Theorem 5.3 to this additive PSM protocol, the resulting single-shuffle full-open protocol requires $mn = 2^{O(2^n)}$ cards, which does not improve the number of cards $2^{O(n^3 2^{n/2})}$ of the construction in Corollary 4.5.

Nevertheless, Theorem 5.3 improves the number of cards for special functions if we use additive PSM protocols in Propositions 3.4 and 3.5. The following corollaries show single-shuffle full-open protocols for the n-input AND function and any symmetric function, using fewer cards than Corollary 4.5.

Corollary 5.4. *There exists a single-shuffle full-open protocol Π for AND_n such that the number of cards is $O(n^2)$ and the shuffle is a composition of n pile-shifting shuffles.*

Corollary 5.5. *For any symmetric function $f : \{0,1\}^n \to \{0,1\}$, there exists a single-shuffle full-open protocol Π for f such that the number of cards is $(1 + o(1))n^3 2^{2n-2}$ and the shuffle is a composition of n pile-shifting shuffles.*

5.2 A Single-Shuffle Full-Open Protocol for Any Function

We extend the single-shuffle full-open protocol for AND_n to a protocol that applies to any function $f : \{0,1\}^n \to \{0,1\}$. The protocol is obtained by concatenating many protocols for AND_n each corresponding to an input $\boldsymbol{a}$ with $f(\boldsymbol{a}) = 1$. This method requires a composition of $|f^{-1}(1)| = O(2^n)$ shuffles but achieves a smaller number of cards compared to the general solution in Corollary 4.5.

Theorem 5.6. *Let Π_0 be a single-shuffle full-open protocol for the n-input AND function. Then, there exists a single-shuffle full-open protocol Π for a function $f : \{0,1\}^n \to \{0,1\}$ such that*

Notations.

- Let $f : \{0,1\}^n \to \{0,1\}$ be a function.
- Let $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an additive PSM protocol over $\mathbb{Z}_m$ for f.
- Let U be a subgroup of $\mathbb{Z}_m^*$ specified by Σ.
- Let $(g_i : \{0,1\} \to \mathbb{Z}_m)_{i\in[n]}$ be n functions specified by Σ.

Construction of L_i^b. For each $i \in [n]$ and each $b \in \{0,1\}$, let $L_i^b = E_m^{\heartsuit}(g_i(b))$.

Shuffling. Given a sequence of mn face-down cards $\underbrace{\boxed{?}\cdots\boxed{?}}_{E_m^{\heartsuit}(g_1(x_1))}\cdots\underbrace{\boxed{?}\cdots\boxed{?}}_{E_m^{\heartsuit}(g_n(x_n))}$,

1. Apply the procedures in Figure 2 to obtain

$$\underbrace{\boxed{?}\cdots\boxed{?}}_{E_m^{\heartsuit}(u\cdot g_1(x_1))}\cdots\underbrace{\boxed{?}\cdots\boxed{?}}_{E_m^{\heartsuit}(u\cdot g_n(x_n))},$$

   where $u \leftarrow_{\$} U$.
2. Apply the procedures in Figure 3 to obtain

$$\underbrace{\boxed{?}\cdots\boxed{?}}_{E_m^{\heartsuit}(u\cdot g_1(x_1)-r_1)}\cdots\underbrace{\boxed{?}\cdots\boxed{?}}_{E_m^{\heartsuit}(u\cdot g_n(x_n)-r_n)},$$

   where $(r_i)_{i\in[n]}$ are random additive shares of zero.

Output.

1. Reveal all face-down cards.
2. For each $i \in [n]$, let μ_i be the position of $\heartsuit$ in the sequence $E_m^{\heartsuit}(u \cdot g_i(x_i) - r_i)$.
3. Run Dec on input $(\mu_1, \ldots, \mu_n)$ to obtain $y \in \{0,1\}$.
4. Output y.

Figure 4: A single-shuffle full-open protocol based on an additive PSM protocol

Notations.

- Let $f : \{0,1\}^n \to \{0,1\}$ be a function.
- Let $f^{-1}(1) = \{\boldsymbol{a} \in \{0,1\}^n : f(\boldsymbol{a}) = 1\} = \{\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(N)}\}$ and denote $\boldsymbol{a}^{(j)} = (a_1^{(j)}, \ldots, a_n^{(j)})$.
- Let Π_0 be a single-shuffle full-open protocol for AND_n.
- Let $(\lambda_i^b)_{i\in[n],b\in\{0,1\}}$ be $2n$ sequences of suits, $\mathcal{D}_0$ be a probability distribution, Output_0 be an output function specified by Π_0.

Construction of L_i^b. For each $i \in [n]$ and each $b \in \{0,1\}$, let $L_i^b = \lambda_i^{\phi^{(1)}(b)} || \cdots || \lambda_i^{\phi^{(N)}(b)}$, where $||$ denotes the concatenation of suits and we set $\phi^{(j)}(b) = 1$ if $a_i^{(j)} = b$ and $\phi^{(j)}(b) = 0$ otherwise.

Shuffling. Given a sequence of face-down cards

$$\underbrace{?\cdots?}_{\lambda_1^{\phi^{(1)}(x_1)}}\cdots\underbrace{?\cdots?}_{\lambda_1^{\phi^{(N)}(x_1)}}\cdots\underbrace{?\cdots?}_{\lambda_n^{\phi^{(1)}(x_n)}}\cdots\underbrace{?\cdots?}_{\lambda_n^{\phi^{(N)}(x_n)}},$$

1. For each $j \in [N]$, sample a permutation $\pi^{(j)} \leftarrow \mathcal{D}_0$ independently and apply $\pi^{(j)}$ to part of the sequence

$$\underbrace{?\cdots?}_{\lambda_1^{\phi^{(j)}(x_1)}}\cdots\underbrace{?\cdots?}_{\lambda_n^{\phi^{(j)}(x_n)}} \to \underbrace{?\cdots?}_{\mu^{(j)}}.$$

2. Apply a pile-scrambling shuffle as follows:

$$\Big[\underbrace{?\cdots?}_{\mu^{(1)}}\Big|\cdots\Big|\underbrace{?\cdots?}_{\mu^{(N)}}\Big] \to \underbrace{?\cdots?}_{\mu^{(j_1)}}\cdots\underbrace{?\cdots?}_{\mu^{(j_N)}}.$$

Output.

1. Reveal all face-down cards.
2. Compute $(y^{(1)}, \ldots, y^{(N)}) = (\mathsf{Output}_0(\mu^{(j_1)}), \ldots, \mathsf{Output}_0(\mu^{(j_N)}))$.
3. If there exists a unique $h \in [N]$ such that $y^{(h)} = 1$, then output 1. Otherwise, output 0.

Figure 5: A single-shuffle full-open protocol based on a single-shuffle full-open protocol for AND_n

- *The number of cards is $N\ell$, where $N = |f^{-1}(1)| = |\{\boldsymbol{a} \in \{0,1\}^n : f(\boldsymbol{a}) = 1\}|$ and ℓ is the number of cards of Π_0.*
- *The shuffle is a composition of N independent shuffles for Π_0 and a pile-scramble shuffle.*

Proof. Let Π be a card-based protocol described in Figure 5.

Let $\boldsymbol{x} = (x_i)_{i\in[n]}$ be inputs. The sequence of suits $(\lambda_i^{\phi^{(j)}(x_i)})_{i\in[n]}$ corresponds to an input sequence when executing Π_0 on input $(\phi^{(j)}(x_i))_{i\in[n]}$. Thus, $\mathsf{Output}_0(\mu^{(j)}) = 1$ if and only if $\phi^{(j)}(x_1) = \cdots = \phi^{(j)}(x_n) = 1$, i.e., $\boldsymbol{x} = \boldsymbol{a}^{(j)}$. On the other hand, if $f(\boldsymbol{x}) = 1$, then there exists a unique $j \in [N]$ such that $\boldsymbol{x} = \boldsymbol{a}^{(j)}$, and if $f(\boldsymbol{x}) = 0$, then $\boldsymbol{x} \neq \boldsymbol{a}^{(j)}$ for all $j \in [N]$. Therefore, if $f(\boldsymbol{x}) = 1$, then there exists a unique $y^{(h)}$ such that $y^{(h)} = 1$ at Step 3 in the output phase, and the protocol outputs 1. (This indicates that $\boldsymbol{x} = \boldsymbol{a}^{(j_h)}$.) If $f(\boldsymbol{x}) = 0$, then $y^{(1)} = \cdots = y^{(N)} = 0$ and the protocol outputs 0. This shows the correctness of Π.

To see privacy, let $\boldsymbol{x} = (x_i)_{i\in[n]}$ and $\tilde{\boldsymbol{x}} = (\tilde{x}_i)_{i\in[n]}$ be inputs with $f(\boldsymbol{x}) = f(\tilde{\boldsymbol{x}})$. Let $(\mu^{(j_1)}, \ldots, \mu^{(j_N)})$ and $(\tilde{\mu}^{(j_1)}, \ldots, \tilde{\mu}^{(j_N)})$ denote the suits of cards revealed in the output phase when executing Π on input $\boldsymbol{x}$ and on input $\tilde{\boldsymbol{x}}$, respectively. First, assume that $f(\boldsymbol{x}) = f(\tilde{\boldsymbol{x}}) = 0$, i.e., $\boldsymbol{x}, \tilde{\boldsymbol{x}} \notin \{\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(N)}\}$. Then, for any $j \in [N]$, both the outputs of Π_0 on input $(\phi^{(j)}(x_i))_{i\in[n]}$ and on input $(\phi^{(j)}(\tilde{x}_i))_{i\in[n]}$ are 0. Thus, the privacy of Π_0 ensures that the distributions of $\mu^{(j)}$ and $\tilde{\mu}^{(j)}$ are identical, which then implies that the distributions of $(\mu^{(j_h)})_{h\in[N]}$ and $(\tilde{\mu}^{(j_h)})_{h\in[N]}$ are identical. Next, assume that $f(\boldsymbol{x}) = f(\tilde{\boldsymbol{x}}) = 1$. Then, during the execution of Π on input $\boldsymbol{x}$, there exists a unique $j \in [N]$ such that Π_0 outputs 1 on input $(\phi^{(j)}(x_i))_{i\in[n]}$. Also, during the execution of Π on input $\tilde{\boldsymbol{x}}$, there exists a unique $j' \in [N]$ such that Π_0 outputs 1 on input $(\phi^{(j')}(\tilde{x}_i))_{i\in[n]}$. Thus, the privacy of Π_0 ensures that the distribution of $\mu^{(j)}$ is identical to that of $\tilde{\mu}^{(j')}$; and that all other sequences $\mu^{(k)}$ for $k \in [N] \setminus \{j\}$ and $\tilde{\mu}^{(k')}$ for $k' \in [N] \setminus \{j'\}$ follow the same distribution (namely, the distribution of cards revealed during the execution of Π_0 with output 0). Since the N piles $(\mu^{(1)}, \ldots, \mu^{(N)})$ and $(\tilde{\mu}^{(1)}, \ldots, \tilde{\mu}^{(N)})$ are randomly permuted at Step 2 in the shuffling phase, we conclude that their resulting distributions are identical.

For each $j \in [N]$, the number of cards to encode $(\lambda_i^{\phi^{(j)}(x_i)})_{i\in[n]}$ is ℓ. Thus, the total number of cards is $N\ell$. □

By applying Theorem 5.6 to the protocol for AND_n in Corollary 5.4, we obtain a single-shuffle full-open protocol for any function using fewer cards than Corollary 4.5.

Corollary 5.7. *For any $f : \{0,1\}^n \to \{0,1\}$, there exists a single-shuffle full-open protocol for f such that the number of cards is $O(n^2 2^n)$ and the shuffle is a composition of at most $(n+1)2^n$ pile-shifting shuffles and a pile-scramble shuffle.*

6 Relation with PSM

Shinagawa and Nuida [28] showed that a single-shuffle full-open protocol for $f : \{0,1\}^n \to \{0,1\}$ using ℓ cards implies a PSM protocol for f with communication complexity $c = \ell n$ and randomness complexity $r = \ell \log \ell + \ell n$. Conversely, Theorem 4.1 shows that a PSM protocol for f with communication complexity c and randomness complexity r implies a single-shuffle full-open protocol for f using $\ell = c2^{r+1}$ cards.

Combining both results, we can relate the minimum number of cards required by single-shuffle full-open protocols for f to the minimum communication/randomness complexity of PSM protocols for f. We prepare notations. For $f : \{0,1\}^n \to \{0,1\}$, we define

- $\mathsf{card}(f)$ as the minimum number of cards required by any single-shuffle full-open protocol for f.
- $\mathsf{rand}(f)$ as the minimum randomness complexity of any PSM protocol for f.
- $\mathsf{comm}(f)$ as the minimum communication complexity of any PSM protocol for f.

First, we show two lemmas regarding the communication and randomness complexity of PSM protocols[4].

[4]These results are folklore to the best of our knowledge. We include formal proofs for the sake of completeness.

Lemma 6.1. *Let $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PSM protocol for a function $f : \{0,1\}^n \to \{0,1\}$ with communication complexity c and randomness complexity r. Then, there exists a PSM protocol $\Sigma' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ for f with communication complexity $c' = \min\{c, (r+1)n\}$ and randomness complexity $r' = r$.*

Proof. Let $i \in [n]$ and suppose that the bit-length c_i of messages for party i is larger than $r+1$. Define a map $M_i : \{0,1\} \times \{0,1\}^r \to \{0,1\}^{c_i}$ as $M_i(b,\rho) = \mathsf{Enc}(i,b,\rho)$ for any $b \in \{0,1\}$ and $\rho \in \{0,1\}^r$. Define $\mathcal{M}_i = \{M_i(b,\rho) \in \{0,1\}^{c_i} : b \in \{0,1\}, \rho \in \{0,1\}^r\}$. Let $C_i = |\mathcal{M}_i|$ and enumerate $\mathcal{M}_i = \{m_{i,1}, \ldots, m_{i,C_i}\}$. Define $\phi_i : \{0,1\} \times \{0,1\}^r \to [C_i]$ as a map that maps (b,ρ) into k such that $M_i(b,\rho) = m_{i,k}$. We define a new PSM protocol $\Sigma^{(i)} = (\mathsf{Gen}^{(i)}, \mathsf{Enc}^{(i)}, \mathsf{Dec}^{(i)})$ for f as $\mathsf{Gen}^{(i)} = \mathsf{Gen}$,

$$\mathsf{Enc}^{(i)}(j,b,\rho) = \begin{cases} \phi_i(b,\rho) \in [C_i], & \text{if } j = i, \\ \mathsf{Enc}(j,b,\rho) \in \{0,1\}^{c_j}, & \text{otherwise,} \end{cases}$$
$$\mathsf{Dec}^{(i)}(\mu_1, \ldots, \mu_n) = \mathsf{Dec}(\mu_1, \ldots, \mu_{i-1}, m_{i,k}, \mu_{i+1}, \ldots, \mu_n),$$

where $k = \mu_i \in [C_i]$. Since $\lceil \log C_i \rceil \le r+1$, the communication complexity of $\Sigma^{(i)}$ is $\sum_{j \neq i} c_j + \min\{c_i, r+1\}$ and the randomness complexity of $\Sigma^{(i)}$ is r. By applying the above procedures to every $i \in [n]$ with $c_i > r+1$, we obtain a PSM protocol $\Sigma' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ with communication complexity $c' = \sum_{i \in [n]} \min\{c_i, r+1\} \le \min\{c, (r+1)n\}$ and randomness complexity $r' = r$. □

Lemma 6.2. *Let $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PSM protocol for a function $f : \{0,1\}^n \to \{0,1\}$ with communication complexity c and randomness complexity r. Then, there exists a PSM protocol $\Sigma' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ for f with communication complexity $c' = c$ and randomness complexity $r' = \min\{r, c2^n\}$.*

Proof. Let $\mathcal{D}$ be a probability distribution over $\{0,1\}^r$ from which Σ samples randomness. Suppose that $|\mathrm{supp}(\mathcal{D})| > 2^{c2^n}$. For each $\rho \in \mathrm{supp}(\mathcal{D})$, define $\mathsf{Msg}_\rho : \{0,1\}^n \to \{0,1\}^c$ as $\mathsf{Msg}_\rho(\boldsymbol{x}) = (\mathsf{Enc}(i, x_i, \rho))_{i \in [n]}$ for any $\boldsymbol{x} = (x_i)_{i \in [n]} \in \{0,1\}^n$. Then, there exist $\rho_1 \neq \rho_2 \in \mathrm{supp}(\mathcal{D})$ such that Msg_{ρ_1} is identical to Msg_{ρ_2}, i.e., $\mathsf{Msg}_{\rho_1}(\boldsymbol{x}) = \mathsf{Msg}_{\rho_2}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \{0,1\}^n$. Define $\mathcal{D}_1$ as a probability distribution whose support is $\mathrm{supp}(\mathcal{D}_1) = \mathrm{supp}(\mathcal{D}) \setminus \{\rho_2\}$ and whose probability mass function is

$$\Pr_{\mathcal{D}_1}[\rho] = \begin{cases} \Pr_{\mathcal{D}}[\rho], & \text{if } \rho \neq \rho_1, \\ \Pr_{\mathcal{D}}[\rho_1] + \Pr_{\mathcal{D}}[\rho_2], & \text{if } \rho = \rho_1. \end{cases} \tag{1}$$

Let Σ_1 be a PSM protocol which is the same as Σ except that randomness is chosen according to $\mathcal{D}_1$. Since both ρ_1 and ρ_2 produce identical messages for all inputs, the distribution of messages does not change if $\mathcal{D}$ is replaced with $\mathcal{D}_1$, which means that Σ_1 is also a PSM protocol for f. Note that $|\mathrm{supp}(\mathcal{D}_1)| = |\mathrm{supp}(\mathcal{D})| - 1$. We can apply the above procedures as long as the size of the support of the probability distribution of randomness is larger than 2^{c2^n}. Thus, we obtain a PSM protocol $\Sigma'' = (\mathsf{Gen}'', \mathsf{Enc}'', \mathsf{Dec}'')$ for f such that the size of the support of the distribution $\mathcal{D}''$ of randomness is $R := |\mathrm{supp}(\mathcal{D}'')| \le 2^{c2^n}$. Enumerate $\mathrm{supp}(\mathcal{D}'') = \{\rho_1, \ldots, \rho_R\}$. Let $r' = \lceil \log R \rceil \le c2^n$. Define $\mathcal{D}'$ as a probability distribution over $\{0,1\}^{r'}$ whose probability mass function is $\Pr_{\mathcal{D}'}[\sigma] = \rho_\sigma$ for any $\sigma \in \{0,1\}^{r'}$ (we here identify $\sigma \in [R]$ with its binary representation). We define $\Sigma' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ as follows: Gen' samples randomness $\sigma \in \{0,1\}^{r'}$ according to $\mathcal{D}'$, $\mathsf{Enc}'(i,b,\sigma) = \mathsf{Enc}''(i,b,\rho_\sigma)$, and $\mathsf{Dec}' = \mathsf{Dec}''$. Then, Σ' is a PSM protocol for f with randomness complexity $r' = \min\{r, c2^n\}$. □

Finally, we show relations among $\mathsf{card}(f)$, $\mathsf{rand}(f)$ and $\mathsf{comm}(f)$.

Theorem 6.3. *For any $f : \{0,1\}^n \to \{0,1\}$, it holds that*

$$\frac{\mathsf{rand}(f)}{\log(\mathsf{rand}(f)) + n} \le \mathsf{card}(f) \le 2^{\mathsf{rand}(f)+1}(\mathsf{rand}(f)+1)n, \tag{2}$$
$$\frac{\mathsf{comm}(f)}{n} \le \mathsf{card}(f) \le 2^{\mathsf{comm}(f)2^n+1}\mathsf{comm}(f). \tag{3}$$

Proof. Let $\ell = \mathsf{card}(f)$, $r = \mathsf{rand}(f)$, and $c = \mathsf{comm}(f)$. It follows from [28] that $r \leq \ell \log \ell + \ell n = \ell \log(\ell 2^n)$, i.e., $r2^n \leq (\ell 2^n) \log(\ell 2^n)$, and $c \leq \ell n$. Since $a \leq b \log b$ implies $a/\log a \leq b$,[5] we have that $r/(\log r + n) \leq \ell$.

Let Π_{r} be a PSM protocol for f achieving the minimum randomness complexity r. Due to Lemma 6.1, we may assume that the communication complexity of Π_{r} is at most $(r+1)n$. Then, it follows from Theorem 4.1 that $\ell \leq 2^{r+1}(r+1)n$.

Let Π_{c} be a PSM protocol for f achieving the minimum communication complexity c. Due to Lemma 6.2, we may assume that the randomness complexity of Π_{c} is at most $c2^n$. Then, it follows from Theorem 4.1 that $\ell \leq 2^{c2^n+1}c$. □

We note that the inequalities in (2) and the first one in (3) still hold if card-based protocols and PSM protocols are restricted to the *uniform* ones, by which we mean that a shuffle and randomness are sampled from uniform distributions over some sets. Indeed, if the initial PSM protocol uses uniform randomness, then the transformation in Theorem 4.1 yields a card-based protocol that uses uniform shuffles only (i.e., a pile-shifting shuffle and a complete shuffle). The transformation in [28] produces a PSM protocol that uses uniform randomness if the initial card-based protocol uses uniform shuffles only. The transformation in Lemma 6.1 also preserves the uniformity of randomness: if the original protocol Σ samples randomness uniformly, then so does the transformed protocol Σ'. Therefore, the inequalities in (2) and the first inequality in (3) still hold in the uniform setting. On the other hand, the transformation in Lemma 6.2 does not preserve uniformity since it modifies the probability mass function in Eq. (1). Currently, it is not ensured that the second inequality in (3) still holds in the uniform setting.

References

[1] A. Beimel, E. Kushilevitz, and P. Nissim. The complexity of multiparty PSM protocols and related models. In *Advances in Cryptology – EUROCRYPT 2018, Part II*, pages 287–318, 2018.

[2] B. D. Boer. More efficient match-making and satisfiability the five card trick. In *Advances in Cryptology – EUROCRYPT' 89*, pages 208–217, 1990.

[3] C. Crépeau and J. Kilian. Discreet solitary games. In *Advances in Cryptology — CRYPTO' 93*, pages 319–330, 1994.

[4] U. Feige, J. Kilian, and M. Naor. A minimal model for secure computation (extended abstract). In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 554–563, 1994.

[5] R. Gradwohl, M. Naor, B. Pinkas, and G. N. Rothblum. Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. *Theory of Computing Systems*, 44(2):245–268, 2009.

[6] J. Heather, S. Schneider, and V. Teague. Cryptographic protocols with everyday objects. *Formal Aspects of Computing*, 26(1):37–62, 2014.

[7] Y. Ishai. Randomization techniques for secure computation. *Secure Multi-Party Computation*, 10:222–248, 2013.

[8] Y. Ishai and E. Kushilevitz. Private simultaneous messages protocols with applications. In *Fifth Israel Symposium on Theory of Computing and Systems, ISTCS 1997*, pages 174–184. IEEE Computer Society, 1997.

[9] R. Ishikawa, E. Chida, and T. Mizuki. Efficient card-based protocols for generating a hidden random permutation without fixed points. In *Unconventional Computation and Natural Computation*, pages 215–226, 2015.

[5] Otherwise, $b \log b < (a/\log a)\log(a/\log a) = a - a \log\log a/\log a < a$.

[10] J. Kastner, A. Koch, S. Walzer, D. Miyahara, Y. Hayashi, T. Mizuki, and H. Sone. The minimum number of cards in practical card-based protocols. In *Advances in Cryptology—ASIACRYPT 2017*, pages 126–155, 2017.

[11] T. Kuzuma, K. Toyoda, D. Miyahara, and T. Mizuki. Card-based single-shuffle protocols for secure multiple-input AND and XOR computations. In *ACM ASIA Public-Key Cryptography Workshop*, pages 51–58, 2022.

[12] D. Miyahara, H. Haneda, and T. Mizuki. Card-based zero-knowledge proof protocols for graph problems and their computational model. In *Provable and Practical Security*, pages 136–152, 2021.

[13] D. Miyahara, Y. Hayashi, T. Mizuki, and H. Sone. Practical card-based implementations of Yao's millionaire protocol. *Theoretical Computer Science*, 803:207–221, 2020.

[14] T. Mizuki, M. Kumamoto, and H. Sone. The five-card trick can be done with four cards. In *Advances in Cryptology—ASIACRYPT 2012*, pages 598–606, 2012.

[15] T. Mizuki and H. Shizuya. A formalization of card-based cryptographic protocols via abstract machine. *International Journal of Information Security*, 13(1):15–23, 2014.

[16] V. Niemi and A. Renvall. Solitaire zero-knowledge. *Fundamenta Informaticae*, 38(1,2):181–188, 1999.

[17] T. Nishida, Y. Hayashi, T. Mizuki, and H. Sone. Card-based protocols for any Boolean function. In *Theory and Applications of Models of Computation*, pages 110–121, 2015.

[18] S. Odaka and Y. Komano. Card-based arithmetic operations and application to statistical data aggregation. In *Innovative Security Solutions for Information Technology and Communications*, 2024.

[19] T. Ono, K. Shinagawa, T. Nakai, Y. Watanabe, and M. Iwamoto. Single-shuffle card-based protocols with six cards per gate. In *Information Security and Cryptology*, pages 157–169, 2024.

[20] S. Ruangwises. Using five cards to encode each integer in Z/6Z. In *Innovative Security Solutions for Information Technology and Communications*, pages 165–177, 2022.

[21] S. Ruangwises and T. Itoh. Securely computing the n-variable equality function with $2n$ cards. In *Theory and Applications of Models of Computation*, pages 25–36, 2020.

[22] H. Shikata, K. Toyoda, D. Miyahara, and T. Mizuki. Card-minimal protocols for symmetric Boolean functions of more than seven inputs. In *Theoretical Aspects of Computing – ICTAC 2022*, pages 388–406, 2022.

[23] K. Shinagawa, R. Eriguchi, S. Satake, and K. Nuida. Private simultaneous messages based on quadratic residues. *Designs, Codes and Cryptography*, 91(12):3915–3932, 2023.

[24] K. Shinagawa, D. Miyahara, and T. Mizuki. How to play Old Maid with virtual players. *Theory of Computing Systems*, 69(1), 2025.

[25] K. Shinagawa and T. Mizuki. The six-card trick: Secure computation of three-input equality. In *Information Security and Cryptology*, pages 123–131, 2018.

[26] K. Shinagawa and K. Nuida. A single shuffle is enough for secure card-based computation of any Boolean circuit. *Discrete Applied Mathematics*, 289:248–261, 2021.

[27] K. Shinagawa and K. Nuida. Single-shuffle full-open card-based protocols imply private simultaneous messages protocols. Cryptology ePrint Archive, Paper 2022/1306, 2022.

[28] K. Shinagawa and K. Nuida. Card-Based Protocols Imply PSM Protocols. In *42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025)*, volume 327 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 72:1–72:18, 2025.

[29] K. Shinagawa and K. Nuida. A note on single-cut full-open protocols, 2025.

[30] Y. Takahashi, K. Shinagawa, H. Shikata, and T. Mizuki. Efficient card-based protocols for symmetric functions using four-colored decks. In *ACM ASIA Public-Key Cryptography Workshop*, pages 1–10, 2024.

[31] K. Toyoda, D. Miyahara, and T. Mizuki. Another use of the five-card trick: Card-minimal secure three-input majority function evaluation. In *Progress in Cryptology—INDOCRYPT 2021*, pages 536–555, 2021.

[32] K. Tozawa, H. Morita, and T. Mizuki. Single-shuffle card-based protocol with eight cards per gate. In *Unconventional Computation and Natural Computation*, pages 171–185, 2023.

[33] A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 160–164, 1982.