An n-Card Threshold Protocol Is Impossible

Shizuru Iino*, Yang Li*, Kazuo Sakiyama*, and Daiki Miyahara*†
*Department of Informatics, The University of Electro-Communications, Tokyo, Japan
Email: {s.iino, miyahara}@uec.ac.jp
† National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

***Abstract*—Card-based cryptography uses a physical deck of playing cards to achieve secure computations. The first card-based protocol, called the five-card trick, allows Alice and Bob to decide whether to have a date without revealing their actual minds, i.e., it performs a secure computation of the logical two-input AND function. It uses a two-color deck of red and black cards, and the Boolean values are encoded based on the arrangement order of two distinct cards. The five-card trick was later optimized into a four-card AND protocol, achieving the minimal number of cards, because each input bit is encoded using two cards. Afterward, an alternative encoding, where each bit is represented by a single card, enabled a three-card AND protocol with a minor modification to the five-card trick. However, the construction of a two-card AND protocol remains an open problem. In this study, we provide a negative result for the problem: a two-card AND protocol is impossible. We extend this result to a more general class of functions, and consequently, derive the main impossibility theorem: an n-card n-input threshold protocol is impossible. Our result implies that the three-card AND protocol uses the minimal number of cards. A unique aspect of our proof is its simplicity; we show that it suffices to consider the initial and its subsequent states, unlike previous impossibility proofs that enumerate all possible states.**

***Index Terms*—Card-based cryptography, logical AND function, secure computation, threshold function.**

I. Introduction

Card-based cryptography performs a secure computation using a deck of physical cards. The most famous card-based protocol is the five-card trick [1] proposed in 1989 by Den Boer. Here, we use a two-color deck of cards, i.e., blacks $\boxed{\clubsuit}$ and reds $\boxed{\heartsuit}$, where the backs of all cards are identical $\boxed{?}$. The arrangement order of one black and one red determines the Boolean values, as follows:

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

We say two face-down cards are a *commitment to x* if the sequence of the two cards indicates a bit $x \in \{0, 1\}$ based on Eq. (1) and write it as follows:

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

A. Five-Card Trick

This paper begins by introducing the five-card trick [1], which performs a secure computation of the logical AND value $a \wedge b$ given commitments to $a, b \in \{0, 1\}$ as well as a red card:

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

That is, this AND protocol allows us to learn the value of $a \wedge b$ without leaking any information about the values of a, b. The five-card trick proceeds as follows.

1) Replace the two cards forming the commitment to a to have a commitment to $\overline{a}$:

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

   Here, note that the three reds are consecutive if and only if $a \wedge b = 1$.
2) Turn over the middle red and apply a *random cut*, which is a cyclic shuffling and is denoted by $\langle \cdots \rangle$:

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

3) Reveal all the five cards; we have $a \wedge b = 1$ if the revealed sequence has three consecutive reds (up to cyclic shift) and $a \wedge b = 0$ otherwise:

$$\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?} \to \begin{cases} \boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit} & \text{if } a \wedge b = 1 \text{ (up to cyclic shift)}, \\ \boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit} & \text{if } a \wedge b = 0 \text{ (up to cyclic shift)}. \end{cases}$$

As described above, the five cards are arranged in the first step so that the middle three cards are $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$ only when $a = b = 1$. A random cut applied in the second step randomizes the order of the sequence, and the resulting sequence is cyclically indistinguishable when $a \wedge b = 0$, i.e., no information about the inputs is leaked.

The five-card trick is referred to as the *non-committed format* because all the cards are turned over to observe the output; the resulting cards does not serve as a commitment. In general, non-committed-format protocols share this property, and the final sequence itself cannot be used as an input of another protocol. Conversely, a card-based protocol is defined as being in *committed format* if its output serves as a commitment to the result value.

B. Motivation

Card-based cryptographic protocols are not intended for direct practical deployment, but they provide a clear and insightful model for exposing information-theoretic and combinatorial limitations, and thus remain valuable for deepening the theoretical foundations of cryptography. Card-based cryptography has long served as a fertile setting for exploring

 DOI: https://doi.org/10.1109/CCWC67433.2026.11393758

concrete models of secure computation: from the classic five-card trick to modern formal frameworks. Studying minimal-card protocols clarifies fundamental limits of protocol design.

The five-card trick described above is an elegant AND protocol, but raises a natural question: can we reduce the number of required cards more? This question was affirmatively answered by Mizuki et al. [2] in 2012, and they proposed a four-card non-committed-format AND protocol:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\ \underbrace{\boxed{?}\boxed{?}}_{b} \ \rightarrow\ \cdots\ \rightarrow\ a \wedge b.$$

Clearly, this uses the minimum number of cards for computing the two-input AND function, as long as we follow the encoding rule defined in Eq. (1).

To further reduce the number of cards required, it would be natural to modify the above encoding rule. Indeed, Mizuki and Shizuya [3] in 2014 proposed another encoding rule, which uses a card with a rotationally asymmetric pattern, i.e., *up-down* cards:[1]

$$\boxed{\downarrow} = 0, \quad \boxed{\uparrow} = 1. \tag{2}$$

That is, in Eq. (1), one bit is represented by the order of two different cards, whereas in Eq. (2), one bit is represented by the top and bottom orientation of a single up-down card. This new encoding is a natural extension, and a three-card committed-format AND protocol was also proposed by the same authors [3]:

$$\underbrace{\boxed{*}}_{a}\boxed{\downarrow}\underbrace{\boxed{*}}_{b} \ \rightarrow\ \cdots\ \rightarrow \underbrace{\boxed{*}}_{a\wedge b},$$

where the backs of all cards are identical and have rotational symmetry of order 2, shown as $\boxed{*}$[2], and a commitment is defined in the same way. In addition, Marcedone et al. [6] reported a number of protocols devised in a university class, including the five-card trick based on Eq. (2). Shinagawa et al. [4] showed an implementation of the three-card committed-format AND protocol [3] introduced above.

However, the outstanding issue remains the possibility of reducing the number of cards required for a three-card AND protocol. That is, the necessity of one additional card remains an open problem. A natural extension of this problem is whether or not an $(n+1)$-card protocol is the minimum number of cards for the n-input AND function for $n \geq 2$. Note that an $(n+1)$-card AND protocol is possible if we repeatedly use the three-card committed-format AND protocol [3] $n-1$ times:

$$\underbrace{\boxed{*}}_{x_1}\underbrace{\boxed{*}}_{x_2}\cdots\underbrace{\boxed{*}}_{x_n}\boxed{\downarrow} \ \rightarrow\ \underbrace{\boxed{*}}_{x_1\wedge x_2}\cdots\underbrace{\boxed{*}}_{x_n}\boxed{\downarrow} \ \rightarrow\ \cdots\ \rightarrow \underbrace{\boxed{*}}_{x_1\wedge x_2\wedge\cdots\wedge x_n}.$$

It is also noteworthy that a *very limited* case of the above problem was solved: An impossibility result on a two-color deck presented by Koch, Walzer, and Härtel [7] states that there is no four-card committed-format AND protocol with finite runtime as long as we follow Eq. (1). Since any protocol based on Eq. (2) can be trivially converted into a protocol based on Eq. (1), it can be concluded that there is no two-card committed-format AND protocol with finite-runtime.

C. Contributions

In this work, we provide a negative result of the aforementioned problem: there is no n-card AND protocol even in non-committed format. From this result, we broaden our focus to a more general class of functions, called *threshold* functions. Given n inputs, threshold functions, denoted by TH^t_n, determines whether the sum of the inputs is greater than t or not.

$$\mathsf{TH}^t_n(x_1, x_2, \ldots, x_n) = \begin{cases} 0 & \text{if } \Sigma_{i=1}^n x_i \leq t, \\ 1 & \text{otherwise,} \end{cases} \tag{3}$$

where t is the threshold boundary with $0 \leq t < n$. We note that it is equivalent to the n-input AND function if $t = n-1$ and to the majority function if $t = \lfloor \frac{n}{2} \rfloor$.

Our main impossibility result is shown in the following theorem. For clarity, we explicitly state the scope of our lower bound: the impossibility result applies to any TH^t_n (threshold) function when inputs are encoded as up-down card commitments and the protocol may use only the operations permitted in our computational model (shuffles, rotations, reveals, and any other allowed local or global transformations). Extensions of the model (e.g., additional markings on cards, more than two colors, or external physical randomness sources) are outside the scope of this claim and require separate analysis.

Theorem 1. *An n-card TH^t_n protocol is impossible.*

This theorem yields the following corollaries.

Corollary 2. *An n-card majority protocol is impossible.*

Corollary 3. *An n-card AND protocol is impossible.*

Regarding the upper bound on the number of cards, no non-trivial TH^t_n protocol has yet been established, other than the one combining AND, NOT, and copying bit protocols [3]. We note that the two-color setting has been extensively studied [8], [9], including specific results for majority [10]–[12], symmetric functions [13]–[17], and all Boolean functions [18]–[21]. Complementing the negative result, we also provide an upper bound for TH^t_n protocol with up-down cards, as in the following theorem.

Theorem 4. *An $(n+1)$-card TH^t_n protocol is possible.*

Theorems 1 and 4 thus yields the following corollary.

Corollary 5. *The number $n+1$ is the necessary and sufficient number of cards required to compute TH^t_n.*

Our impossibility proof is formal and developed within a computational model tailored to up-down cards. Existing models [22] were designed for two-color decks and later extended to standard playing cards [23], where many useful lemmas for lower-bound arguments were established. A recent upgraded model that formalizes rotation operations was proposed by

[1] They [3] used not an up-down card but a black card for the encoding, i.e., $\boxed{\clubsuit}$ and $\boxed{\spadesuit}$.

[2] In the literature $\boxed{\ }$ is often used to represent the back of an up-down card (cf. [4], [5]), but in this study we select $\boxed{*}$.

Shinagawa and Nuida [24], but they did not restate the lemmas needed for proofs. We therefore introduce the computational model for the use of up-down cards, and our reformulated lemmas also constitute a major technical contributions of this work.

The proof relies on a fairly simple observation. To illustrate this, consider the two-card AND case. Given two input commitments to $a, b \in \{0, 1\}$, the total number of possible card sequences corresponding to each input is four, which equals the number of sequences that can be formed by the up-down cards (i.e., $2^2 = 4$). , as shown below.

$$\underbrace{\boxed{*}}_{a}\underbrace{\boxed{*}}_{b} \leftarrow \begin{cases} \boxed{\downarrow}\boxed{\downarrow} & \text{if } (a,b) = (0,0), \\ \boxed{\uparrow}\boxed{\downarrow} & \text{if } (a,b) = (1,0), \\ \boxed{\downarrow}\boxed{\uparrow} & \text{if } (a,b) = (0,1), \\ \boxed{\uparrow}\boxed{\uparrow} & \text{if } (a,b) = (1,1). \end{cases} \tag{4}$$

Therefore, unlike the five-card trick introduced in Sect. I-A, the two-card case inherently lacks the necessary "room" for proper randomization. Any shuffling action thus fails to randomize the sequence when $a \wedge b = 0$. This observation allows us to significantly simplify the proof by considering only the transformation from the initial state shown in Eq. (4) to a subsequent state via a shuffling action. This approach applies directly to the n-card TH^t_n case. Note that the previous impossibility proofs [7], [23], [25]–[27] enumerate all possible states and actions, such as revealing a card. This streamlined approach constitutes the distinctive feature of our study.

Remark. Most recently, Sakurai and Kaji [5] showed that any symmetric function can be computed with $n + 1$ cards. They constructed it by converting an existing $(2n + 2)$-card protocol [18], originally designed for a two-color deck, to the use of up-down cards. While this implies that Theorem 4 has been established in their work, we emphasize that our $(n + 1)$-card protocol is constructed directly for the computation of TH^t_n, offering a more specialized construction. Additionally, Iwasaki [28] established that four cards are the necessary and sufficient number to compute TH^1_2 in a committed-format, i.e., the two-input AND function, if we restrict the set of available shuffles to exclude those that "combine" permutations and rotations. Specifically, he assumes that rotations and permutations are not applied simultaneously within a single shuffle operation. In contrast, our results are derived within a framework that allows for such combined shuffles.

D. Related Work

Threshold Functions. Existing research on card-based cryptography has primarily concentrated on minimizing the number of cards necessary. Many protocols are constructed using two-color decks, and there has been extensive work for computing threshold functions. Haga et al. [9] were the first to propose a sorting protocol, and they showed that this sorting primitive can be straightforwardly build a committed-format TH^t_n protocol. In this paper, we extend the sorting protocol to the up-down card variant in Sect. III. Research on majority functions is more extensive, and for the three-input case, the minimum number of required cards has been determined. Nishida et al. [29] first proposed a committed-format majority protocol using two additional cards, and subsequently, Toyoda et al. [10] developed a protocol that achieves the minimum number of cards without using any additional cards.

$$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{b}\underbrace{\boxed{?}\boxed{?}}_{c} \rightarrow \cdots \rightarrow \underbrace{\boxed{?}\boxed{?}}_{\mathsf{TH}^1_3(a,b,c)}.$$

Private Model. Research on the *private* model employing private permutations has also been progressing. In 2022, Nakai et al. [8] demonstrated that TH^t_n is computable with $n+1$ cards. Regarding majority functions, Abe et al. [11] proposed an n-card protocol in 2022, which was subsequently improved to $\lceil \frac{n}{2} \rceil + 1$ cards in 2023 [12]. Furthermore, a general protocol for computing arbitrary logic circuits was proposed by Ono and Manabe [21]. We note that this study focuses on the *shuffling* model, where all operations are performed in public (assuming that we are given input commitments).

II. Preliminaries

We begin by describing the machine model of card-based protocols [22]–[24], [30] specialized for up-down cards. Using this model, we formally present our proof in the subsequent sections. Due to space constraints, we omit an introduction and a formal description of the five-card trick [1] with up-down cards.

A. Machine Model

Overview. We first formalize up-down cards and a set of allowable operations, such as shuffling and rotating. Next, we define a protocol as a probabilistic state machine, structurally analogous to a Turing machine. Finally, we provide rigorous definitions for correctness and security within this model.

Notations. For an integer $i \in \mathbb{Z}$, we denote by $[i]$ the integer interval from 1 to i. Let $\Sigma := \{\downarrow, \uparrow\}$ be a finite set of *symbols*. For $c \in \Sigma$, we write $\frac{c}{?}$ for a *face-up* card and $\frac{?}{c}$ for a *face-down* card, where "?" denotes the back symbol. We define $\mathsf{top}(\frac{c}{?}) := c$, $\mathsf{top}(\frac{?}{c}) :=?$, $\mathsf{swap}(\frac{c}{?}) := \frac{?}{c}$, and $\mathsf{swap}(\frac{?}{c}) := \frac{c}{?}$. Let $C := \{\frac{c}{?} \mid c \in \Sigma\} \cup \{\frac{?}{c} \mid c \in \Sigma\}$ be the set of all possible cards over Σ. A d-tuple $\Gamma = (\alpha_1, \alpha_2, \ldots, \alpha_d) \in C^d$ is called a *sequence* of d cards, and $\mathsf{vis}(\Gamma)$ denotes a *visible* sequence defined as follows:

$$\mathsf{vis}(\Gamma) := (\mathsf{top}(\alpha_1), \mathsf{top}(\alpha_2), \ldots, \mathsf{top}(\alpha_d)).$$

The set of all visible sequences of d cards is denoted by Vis^d:

$$\mathsf{Vis}^d := \{\mathsf{vis}(\Gamma) \mid \Gamma \in C^d\}.$$

The following two functions are defined on the symbol set Σ:

- $\mathsf{id}(\downarrow) = \downarrow$ and $\mathsf{id}(\uparrow) = \uparrow$ (i.e., the identity function),
- $\neg(\downarrow) = \uparrow$ and $\neg(\uparrow) = \downarrow$ (i.e., the negation function that flips the symbol).

Each of these functions can be naturally extended from Σ to C. We refer to a set of these functions $\mathsf{CMap} := \{\mathsf{id}, \neg\}$ as the set of *rotation* functions.

Protocol. A *protocol* using up-down cards is a quadruple (d, U, Q, A), where:

- $d \in \mathbb{N}$ is the number of up-down cards used in the protocol;
- $U \subseteq C^d$ is an input set;
- Q is a state set with two distinguished states q_0 and q_f, being the initial and final state, respectively;
- A is an action function:

$$A\colon (Q \setminus \{q_f\}) \times \mathsf{Vis}^d \to Q \times \mathsf{Action}^d,$$

where Action^d is a set of actions performed on a sequence of d cards introduced in the next paragraph.

Action. Given a sequence $\Gamma = (\alpha_1, \alpha_2, \ldots, \alpha_d) \in C^d$, a set of actions Action^d includes the following actions on Γ.

- turn: For $T \subseteq [d]$, a *turn* action (turn, T) turns over cards with positions specified by T, i.e.,

$$\mathsf{turn}_T(\Gamma) = (\beta_1, \beta_2, \ldots, \beta_d),$$
$$\text{such that } \beta_i = \begin{cases} \mathsf{swap}(\alpha_i), & \text{if } i \in T, \\ \alpha_i, & \text{otherwise.} \end{cases}$$

- perm: Let S_d denote the symmetric group of degree d. For a permutation $\pi \in S_d$ and a d-tuple of rotation functions $\vec{\phi} = (\phi_1, \ldots, \phi_d) \in \mathsf{CMap}^d$, an (extended) *perm* action $(\mathsf{perm}, (\vec{\phi}, \pi))$ permutes a sequence according to π and applies the functions in $\vec{\phi}$, i.e.,

$$\mathsf{perm}_\omega(\Gamma) = (\phi_1(\alpha_{\pi^{-1}(1)}), \phi_2(\alpha_{\pi^{-1}(2)}), \ldots, \phi_d(\alpha_{\pi^{-1}(d)})),$$

where $\omega = (\vec{\phi}, \pi)$, which is an element of the wreath product $\mathsf{CMap}^d \wr S_d$.[3]
For example, given three commitments to a, b, c, the action $(\mathsf{perm}, ((\neg, \neg, \neg), (1\,3)))$ transforms them as follows:

$$\underbrace{\boxed{*}}_{a}\underbrace{\boxed{*}}_{b}\underbrace{\boxed{*}}_{c} \xrightarrow{\mathsf{perm}} \underbrace{\boxed{*}}_{\bar{c}}\underbrace{\boxed{*}}_{\bar{b}}\underbrace{\boxed{*}}_{\bar{a}}.$$

That is, the first and third cards are swapped, and each of the three cards is rotated to represent its negated value.
- shuf: For a permutation set $\Omega \subseteq \mathsf{CMap}^d \wr S_d$ and a probability distribution $\mathcal{F}$ on Ω, a *shuffling* action $(\mathsf{shuf}, \Omega, \mathcal{F})$ applies an (extended) permutation ω drawn from Ω according to $\mathcal{F}$, i.e.,

$$\mathsf{shuf}_{\Omega,\mathcal{F}} = \mathsf{perm}_\omega(\Gamma),$$

where specific choice of ω remains hidden from all players. We write (shuf, Ω) and omit $\mathcal{F}$ if it is uniform. Note that if $|\Omega| = 1$, then it is equivalent to (perm, ω) for $\omega \in \Omega$.

[3] CMap forms a group isomorphic to $\mathbb{Z}_2$, and $\omega \in \mathsf{CMap}^d \wr S_d$ corresponds to rearranging a bitstring of length d and applying bitwise negation at chosen positions.

For example, given three commitments to a, b, c, the action $(\mathsf{shuf}, \{((\mathsf{id}, \mathsf{id}, \mathsf{id}), \mathsf{id}), ((\neg, \neg, \neg), (1\,3))\})$ transforms them as follows:

$$\underbrace{\boxed{*}}_{a}\underbrace{\boxed{*}}_{b}\underbrace{\boxed{*}}_{c} \xrightarrow{\mathsf{shuf}} \begin{cases} \underbrace{\boxed{*}}_{a}\underbrace{\boxed{*}}_{b}\underbrace{\boxed{*}}_{c} & \text{with prob. } 1/2, \\ \underbrace{\boxed{*}}_{\bar{c}}\underbrace{\boxed{*}}_{\bar{b}}\underbrace{\boxed{*}}_{\bar{a}} & \text{with prob. } 1/2. \end{cases}$$

- result: This special action specifies ℓ output face-down cards and only occurs with the final state, denoted by $(\mathsf{result}, p_1, p_2, \ldots, p_\ell)$ for $p_i \in [d]$.

Correctness. The correctness of $\mathcal{P}$ computing a Boolean function f is defined as follows.

Definition 6 (Correctness)**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. We say that $\mathcal{P} = (d, U, Q, A)$ correctly computes f if the following holds.*

- *$\mathcal{P}$ terminates with a finite number of actions in expectation.*
- *For each input value $b \in \{0,1\}^n$, the input set U contains a unique input sequence $\Gamma^b = (\alpha_1, \ldots, \alpha_d)$ such that*

$$\alpha_i = \begin{cases} \frac{?}{\downarrow}, & \text{if } b[i] = 0, \\ \frac{?}{\uparrow}, & \text{if } b[i] = 1, \end{cases} \quad \text{for } 1 \le i \le n,$$

where $b[i]$ denotes the i-th bit of b.
- *For each input value $b \in \{0,1\}^n$, the protocol $\mathcal{P}$ terminates with a final sequence $\Gamma = (\beta_1, \ldots, \beta_d)$ along with $(\mathsf{result}, p_1, \ldots, p_\ell)$, such that*

$$(\beta_{p_1}, \beta_{p_2}, \ldots, \beta_{p_\ell}) \in \begin{cases} O_0, & \text{if } f(b) = 0, \\ O_1, & \text{if } f(b) = 1, \end{cases} \tag{5}$$

where $O_0, O_1 \subset \left\{\frac{?}{\uparrow}, \frac{?}{\downarrow}\right\}^\ell$ are sets of ℓ-card face-down sequences satisfying $O_0 \cap O_1 = \emptyset$.

Here, O_0 and O_1 "encode" the output value in the final sequence, and their disjointness ensures that the output is uniquely determined. Note that $\mathcal{P}$ is in a committed format iff $\ell = 1$, because $O_0 = \{\frac{?}{\downarrow}\}$ and $O_1 = \{\frac{?}{\uparrow}\}$ (or vice versa).

Security. Security is also an indispensable aspect of protocol design. Intuitively, the security of $\mathcal{P}$ means that no information about the input is leaked through any observable part of the execution of $\mathcal{P}$, including the final output. To capture this, let us introduce some terms; an enumeration of all intermediate sequences $(\Gamma_0, \Gamma_1, \ldots, \Gamma_f)$ from the initial to the final state during the execution of $\mathcal{P}$ is a *sequence-trace* of $\mathcal{P}$. The corresponding *visible* sequence-trace is defined as $(\mathsf{top}(\Gamma_0), \mathsf{top}(\Gamma_1), \ldots, \mathsf{top}(\Gamma_f))$.

Definition 7 (Security)**.** *Let $\mathcal{P} = (d, U, Q, A)$ be a protocol. We say that $\mathcal{P}$ computing $f\colon \{0,1\}^n \to \{0,1\}$ is secure if it satisfies following two requirements:*

- *Let $\mathcal{M}$ be a probability distribution over the input set U, M be the random variable on U induced by $\mathcal{M}$, and $V^{\mathcal{M},\mathcal{P}}$ be the random variable representing the visible sequence-trace resulting from the execution of $\mathcal{P}$ on a*

random input drawn according to $\mathcal{M}$. Then, M and $V^{\mathcal{M},\mathcal{P}}$ are stochastically independent.

- *For each $z \in \{0,1\}$, let $\mathcal{M}_z$ be the conditional distribution of $\mathcal{M}$ given $f(M) = z$. Let $R^{\mathcal{M},\mathcal{P}}$ be the random variable representing the output sequence extracted by the* result *action. Then, for each $z \in \{0,1\}$, M and $R^{\mathcal{M},\mathcal{P}}$ are conditionally independent given $f(M) = z$.*

III. An $n+1$-Card Threshold Protocol

This section presents the construction of an $(n+1)$-card TH^t_n protocol employing up-down cards. We first present a brief idea to construct the protocol and then describe the protocol. Due to space constraints, we omit a pseudocode for our proposed protocol.

A. Idea

The proposed protocol is based on the idea of sequentially adding a commitment to a sorted sequence. Suppose that we have a sequence of three commitments to $x_1, x_2, x_3 \in \{0,1\}$ that has already been sorted in descending order, denoted by Γ, and want to correctly add a commitment to x_4 to Γ:

$$\Gamma = \boxed{*}\boxed{*}\boxed{*} \quad \leftarrow \quad \underbrace{\boxed{*}}_{x_4}.$$

If we reveal the value of x_4, we can add it to the right or left to obtain a sorted sequence of four cards Γ' (in descending order):

$$\Gamma' = \begin{cases} \boxed{*}\boxed{*}\boxed{*}\boxed{\downarrow} & \text{if } x_4 = 0, \\ \boxed{\uparrow}\boxed{*}\boxed{*}\boxed{*} & \text{if } x_4 = 1. \end{cases}$$

However, we cannot reveal the value of x_4, and hence, we add a randomization as follows:[4]

- Do nothing (i.e., real world) with probability 1/2.
- All three cards of Γ are negated and rearranged in reverse order, and the commitment to x_4 is negated (i.e., reverse world) with probability 1/2.

Then, we reveal the commitment to obtain the value of either x_4 or $\overline{x_4}$, and add it to Γ in the same manner in both the real and reverse worlds, since Γ remains sorted in descending order even in the reverse world. Note that revealing either x_4 or $\overline{x_4}$ does not leak information of x_4 (if we do not know which value is revealed).

By repeating this insertion procedure for every input, the protocol incrementally builds the sorted sequence. Finally, we need to "reverse" the sequence if it is in the reverse world. For this, one additional card called *marker card* is used to mark whether the sorted sequence is in the real or reverse. Therefore, this protocol uses $n+1$ cards in total.

[4]This randomization is the up-down version of the random inversion protocol [16].

B. Description

Given a commitment to $x_i \in \{0,1\}$ for $1 \le i \le n$, an $(n+1)$-card TH^t_n protocol proceeds as follows.

1) Place the input commitments along with a marker card $\boxed{\downarrow}$ and turn over the marker card as follows:
$$\underbrace{\boxed{*}}_{x_1}\underbrace{\boxed{*}}_{x_2}\cdots\underbrace{\boxed{*}}_{x_n}\boxed{\downarrow} \;\rightarrow\; \underbrace{\boxed{*}}_{x_1}\underbrace{\boxed{*}}_{x_2}\cdots\underbrace{\boxed{*}}_{x_n}\underbrace{\boxed{*}}_{0}.$$
2) Repeat the following steps from $k = 2$ to $k = n$:
   a) Apply a "specific" shuffling operation: it results in either the sequence of $n+1$ cards are remained the same, or all values are negated plus the first $k-1$ cards are reversed in order. An example for $k = 4$ and $n = 4$ is shown below. Here, for simplicity, the card faces are omitted and only each value $y_i \in \{0,1\}$ are displayed.
   $$y_1y_2y_3y_4y_5 \;\rightarrow\; \begin{cases} y_1y_2y_3y_4y_5, & \text{with prob. } 1/2, \\ \overline{y}_3\overline{y}_2\overline{y}_1\overline{y}_4\overline{y}_5, & \text{with prob. } 1/2. \end{cases}$$
   b) Reveal the k-th card: if it is $\boxed{\downarrow}$, then leave it in place; otherwise, move it to the leftmost position.
   $$y_1y_2\cdots y_{k-1}y_k \rightarrow \begin{cases} y_1y_2\cdots y_{k-1}0 \;\rightarrow\; y_1y_2\cdots y_{k-1}0 \\ y_1y_2\cdots y_{k-1}1 \;\rightarrow\; 1y_ky_1y_2\cdots y_{k-1} \end{cases}$$
   c) Turn over the previously revealed card.
3) Perform the shuffling operation described in Step 2(a) with $k = n+1$.
4) Reveal the marker card: if it is $\boxed{\uparrow}$, then negate all cards and reverse the order of the first n cards.
5) The sequence of the first n cards is now a sorted sequence of the input commitments. Output the $(t+1)$-st card, which is a commitment to $\mathsf{TH}^t_n(x_1, x_2, \ldots, x_n)$.

Correctness and security of this protocol is clear from the above description, and this proves Theorem 4.

IV. Impossibility of n-Card Threshold Protocol

We present a formal proof for our main impossibility result, based on the machine model defined in Sect. II.

A. Lemmas for Impossibility Proof

Before presenting the proof, from the correctness definition in Definition 6, we derive the following lemma.[5]

Lemma 8. *Let $\mathcal{P} = (d, U, Q, A)$ be a protocol computing $f\colon \{0,1\}^n \to \{0,1\}$, and suppose that a shuffling action $(\mathsf{shuf}, \Omega, \mathcal{F})$ for any $\Omega \subseteq \mathsf{CMap}^d \wr S_d$ is applied first, i.e., $A(q_0, \{?\}^d) = (q_1, (\mathsf{shuf}, \Omega, \mathcal{F}))$. Then, for any two inputs $b_0, b_1 \in \{0,1\}^n$ with $f(b_0) = 0$ and $f(b_1) = 1$, and for any distinct $\omega, \omega' \in \Omega$, it must hold that*

$$\omega(\Gamma^{b_0}) \neq \omega'(\Gamma^{b_1}), \quad \textit{where } \Gamma^{b_0}, \Gamma^{b_1} \in U.$$

Proof. Assume for contradiction that $\omega(\Gamma^{b_0}) = \omega'(\Gamma^{b_1})$. Then, the protocol applies all subsequent actions to this sequence

[5]This is equivalent to the claim regarding $\perp$-sequences in [23].

and must produce the same final sequence for both b_0 and b_1. Let Γ_{out} denote the output sequence extracted by the result action of $\mathcal{P}$. By Eq. (5), we have both $\Gamma_{\text{out}} \in O_0$ and $\Gamma_{\text{out}} \in O_1$ because Γ_{out} would be the output for both b_0 and b_1. This leads to a contradiction since $O_0 \cap O_1 = \emptyset$, which completes the proof. □

From the security definition in Definition 7, we derive the following lemma.[6]

Lemma 9. *Let $\mathcal{P} = (d, U, Q, A)$ be a secure protocol computing $f\colon \{0,1\}^n \to \{0,1\}$, and suppose that the first action is a shuffle $(\mathsf{shuf}, \Omega, \mathcal{F})$ with $\Omega \subseteq \mathsf{CMap}^d \wr S_d$, and the next is a* turn *action. Then, for any input $b \in \{0,1\}^n$, there must exist distinct $\omega, \omega' \in \Omega$ such that*

$$\omega(\Gamma^b) \neq \omega'(\Gamma^b), \quad \textit{where } \Gamma^b \in U.$$

Proof. Assume for contradiction that for some input $b \in \{0,1\}^n$, we have $\omega(\Gamma^b) = \omega'(\Gamma^b)$ for all $\omega, \omega' \in \Omega$. That is, the shuffle yields the same sequence for b regardless of which permutation is chosen from Ω; in other words, no randomness is introduced by the shuffle for this input. Since the next action is the turn action, the visible sequence for b after the action must be fixed, i.e., the turn action uniquely determines the input b. However, this contradicts Definition 7, which states that the visible sequence must reveal no information about the input. This completes the proof. □

B. Impossibility Proof

We are ready to present our impossibility result.

Theorem 10 (Formal version of Theorem 1)**.** *A TH^t_n protocol $\mathcal{P} = (n, U, Q, A)$ is impossible.*

Proof. First, due to the security requirement, inputs cannot be disclosed (even partially). By Lemma 9, a shuf action must be applied before any turn or result action.

Given n commitments, consider $(\mathsf{shuf}, \Omega, \mathcal{F})$ for an arbitrary $\Omega \subseteq \mathsf{CMap}^n \wr S_n$ with $|\Omega| \geq 2$ and a probability distribution $\mathcal{F}$ is applied in the beginning. We may assume without loss of generality that Ω includes the identity permutation and another permutation $\omega \in \mathsf{CMap}^n \wr S_n$, as the argument applies equally to the case where Ω does not include the identity permutation.

The proof focuses on inputs whose Hamming weight lies near the boundary of the threshold t. We denote by $\mathrm{wt}(x)$ the Hamming weight of a bit string x, and naturally extend it to a sequence, where $\mathrm{wt}(\Gamma)$ denotes the number of $\frac{?}{\uparrow}$ in $\Gamma \in C^n$. Let $b_0, b_1 \in \{0,1\}^n$ denote input values with $\mathrm{wt}(b_0) = t$ and $\mathrm{wt}(b_1) = t+1$, respectively. Here, from Eq. (3), we have $\mathsf{TH}^t_n(b_0) = 0$ and $\mathsf{TH}^t_n(b_1) = 1$.

Let k denote the number of negation functions $\neg(\cdot)$ in ω, where $0 \leq k \leq n$. We perform a case analysis on k as follows.

- The case for $k = 0$ is trivial. Since ω just permutes a sequence, it follows that for $b' \in \{0\}^n \cup \{1\}^n$ we have $\Gamma^{b'} = \omega(\Gamma^{b'})$, which contradicts Lemma 9. Recall that for each input b, a unique sequence Γ^b is contained in U.
- When $1 \leq k \leq n$, there should exist an input $b'_1 \in \{0,1\}^n$ with $\mathrm{wt}(b'_1) > t$ such that $\omega(\Gamma^{b_0}) = \Gamma^{b'_1}$ when $t \leq \lfloor n/2 \rfloor$ and symmetrically $\omega(\Gamma^{b_1}) = \Gamma^{b'_0}$ when $t > \lfloor n/2 \rfloor$. This contradicts Lemma 8.

The rest of this proof specifies the concrete configuration of b_0 to determine b'_1 assuming $t \leq \lfloor n/2 \rfloor$, and the other case is analogous. Let $\omega = (\vec{\phi}, \pi)$ where $\vec{\phi} = (\phi_1, \phi_2, \ldots, \phi_n)$. We denote the set of indices of the negation functions in $\vec{\phi}$ by $I \subset [n]$, i.e., $\phi_i = \neg$ iff $i \in I$, where $|I| = k$. With this notation, $\mathrm{wt}(\omega(\Gamma^{b_0}))$ is maximized when the positions of $\frac{?}{\downarrow}$ in Γ^{b_0} overlap with I to the maximum extent possible, i.e., at $\min(n-t, k)$ positions. Here, $\min(n-t, k)$ represents the number of $\frac{?}{\downarrow}$ that are negated by ω while $k - \min(n-t, k)$ represents the number of $\frac{?}{\uparrow}$ that are negated. We therefore have

$$\begin{aligned}
\mathrm{wt}(\omega(\Gamma^{b_0})) &= t + \min(n-t, k) - (k - \min(n-t, k)), \\
&= t - k + 2\min(n-t, k), \\
&> t \quad (\because 1 \leq k \leq n \text{ and } t \leq \lfloor n/2 \rfloor).
\end{aligned}$$

This implies that there exists such b'_1 with $\omega(\Gamma^{b_0}) = \Gamma^{b'_1}$, which completes the proof. □

Referring to Corollaries 2 and 3, we immediately obtain the special cases for the AND and majority functions.

V. Concluding Remarks

In this paper, we provided an upper bound and a lower bound for the number of cards required to compute threshold functions. We constructed an $(n+1)$-card threshold protocol with up-down cards, and additionally, we proved that an n-card threshold protocol is impossible. The proof is provided in a formal way, based on a machine model formulated in the literature. Compared to the existing research, our proof was based on a simple observation, because it considers only the initial and its subsequent states to prove the nonexistence. We hope this study will serve as a foundation for future lower-bound results on up-down cards, such as symmetric functions.

Card-based cryptographic protocols are not primarily intended for direct practical application to cybersecurity, but they provide a "physical" model for exposing information-theoretic and combinatorial limitations, making them valuable for deepening the theoretical foundations of cryptography. Educational applications are also conceivable: for instance, the five-card trick [1] serves as a good example to explain the concept of secure computations, as shown in several textbooks [31]. Although the primary motivation of this work is theoretical, we believe that our nontrivial finding that the five-card trick cannot be realized with two cards, will capture broad interest.

Acknowledgment

We thank the anonymous referees, whose comments have helped us to improve the presentation of the paper. This work was supported in part by JSPS KAKENHI Grant Number JP23H00479 and Institute of Mathematics for Industry, Joint Usage/Research Center in Kyushu University. (2023a020 and 2024a035).

[6] This is equivalent to an i/j-state for $i, j \geq 2$ [23].

References

[1] B. D. Boer, "More efficient match-making and satisfiability *The Five Card Trick*," in *Advances in Cryptology – EUROCRYPT' 89*, ser. LNCS, J.-J. Quisquater and J. Vandewalle, Eds., vol. 434. Heidelberg: Springer, 1990, pp. 208–217. [Online]. Available: https://doi.org/10.1007/3-540-46885-4_23

[2] T. Mizuki, M. Kumamoto, and H. Sone, "The five-card trick can be done with four cards," in *Advances in Cryptology—ASIACRYPT 2012*, ser. LNCS, X. Wang and K. Sako, Eds., vol. 7658. Berlin, Heidelberg: Springer, 2012, pp. 598–606. [Online]. Available: https://doi.org/10.1007/978-3-642-34961-4_36

[3] T. Mizuki and H. Shizuya, "Practical card-based cryptography," in *Fun with Algorithms*, ser. LNCS, A. Ferro, F. Luccio, and P. Widmayer, Eds., vol. 8496. Cham: Springer, 2014, pp. 313–324. [Online]. Available: https://doi.org/10.1007/978-3-319-07890-8_27

[4] K. Shinagawa, K. Nuida, T. Nishide, G. Hanaoka, and E. Okamoto, "Committed AND protocol using three cards with more handy shuffle," in *2016 International Symposium on Information Theory and Its Applications*. IEEE, 2016, pp. 700–702. [Online]. Available: https://ieeexplore.ieee.org/document/7840515/

[5] T. Sakurai and Y. Kaji, "General conversion scheme of card-based protocols for two-colored cards to updown cards," in *Emerging Security Information, Systems and Technologies (SECURWARE 2025)*. Wilmington: IARIA, 2025, pp. 33–39.

[6] A. Marcedone, Z. Wen, and E. Shi, "Secure dating with four or fewer cards," Cryptology ePrint Archive, Report 2015/1031, 2015. [Online]. Available: https://ia.cr/2015/1031

[7] A. Koch, S. Walzer, and K. Härtel, "Card-based cryptographic protocols using a minimal number of cards," in *Advances in Cryptology—ASIACRYPT 2015*, ser. LNCS, T. Iwata and J. H. Cheon, Eds., vol. 9452. Berlin, Heidelberg: Springer, 2015, pp. 783–807. [Online]. Available: https://doi.org/10.1007/978-3-662-48797-6_32

[8] T. Nakai, Y. Misawa, Y. Tokushige, M. Iwamoto, and K. Ohta, "Secure computation for threshold functions with physical cards: Power of private permutations," *New Gener. Comput.*, vol. 40, pp. 95–113, 2022. [Online]. Available: https://doi.org/10.1007/s00354-022-00153-7

[9] R. Haga, K. Toyoda, Y. Shinoda, D. Miyahara, K. Shinagawa, Y. Hayashi, and T. Mizuki, "Card-based secure sorting protocol," in *Advances in Information and Computer Security*, ser. LNCS, C.-M. Cheng and M. Akiyama, Eds., vol. 13504. Cham: Springer, 2022, pp. 224–240. [Online]. Available: https://doi.org/10.1007/978-3-031-15255-9_12

[10] 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*, ser. LNCS, A. Adhikari, R. Küsters, and B. Preneel, Eds., vol. 13143. Cham: Springer, 2021, pp. 536–555. [Online]. Available: https://doi.org/10.1007/978-3-030-92518-5_24

[11] Y. Abe, T. Nakai, Y. Kuroki, S. Suzuki, Y. Koga, Y. Watanabe, M. Iwamoto, and K. Ohta, "Efficient card-based majority voting protocols," *New Gener. Comput.*, vol. 40, pp. 173–198, 2022. [Online]. Available: https://doi.org/10.1007/s00354-022-00161-7

[12] Y. Abe, T. Nakai, Y. Watanabe, M. Iwamoto, and K. Ohta, "A computationally efficient card-based majority voting protocol with fewer cards in the private model," *IEICE Trans. Fundam.*, vol. 106, no. 3, pp. 315–324, 2023. [Online]. Available: https://doi.org/10.1587/transfun.2022CIP0021

[13] S. Ruangwises and T. Itoh, "Securely computing the n-variable equality function with $2n$ cards," *Theor. Comput. Sci.*, vol. 887, pp. 99–110, 2021. [Online]. Available: https://doi.org/10.1016/j.tcs.2021.07.007

[14] 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*, ser. LNCS, H. Seidl, Z. Liu, and C. S. Pasareanu, Eds., vol. 13572. Cham: Springer, 2022, pp. 388–406. [Online]. Available: https://doi.org/10.1007/978-3-031-17715-6_25

[15] H. Shikata, D. Miyahara, and T. Mizuki, "Few-helping-card protocols for some wider class of symmetric Boolean functions with arbitrary ranges," in *ACM ASIA Public-Key Cryptography Workshop*. New York: ACM, 2023, pp. 33–41. [Online]. Available: https://doi.org/10.1145/3591866.3593073

[16] 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*. New York: ACM, 2024, pp. 1–10. [Online]. Available: https://doi.org/10.1145/3659467.3659902

[17] S. Ikeda, Y. Takahashi, K. Shinagawa, and K. Nuida, "Efficient card-based protocols for symmetric and partially doubly symmetric functions," in *Advances in Information and Computer Security*, ser. LNCS, C. Cid and N. Yanai, Eds., vol. 16208. Singapore: Springer, 2025, pp. 189–209. [Online]. Available: https://doi.org/10.1007/978-981-95-4674-9_10

[18] T. Nishida, Y. Hayashi, T. Mizuki, and H. Sone, "Card-based protocols for any Boolean function," in *Theory and Applications of Models of Computation*, ser. LNCS, R. Jain, S. Jain, and F. Stephan, Eds., vol. 9076. Cham: Springer, 2015, pp. 110–121. [Online]. Available: https://doi.org/10.1007/978-3-319-17142-5_11

[19] K. Shinagawa and K. Nuida, "A single shuffle is enough for secure card-based computation of any Boolean circuit," *Discrete Applied Mathematics*, vol. 289, pp. 248–261, 2021. [Online]. Available: https://doi.org/10.1016/j.dam.2020.10.013

[20] K. Shinagawa, T. Mizuki, J. Schuldt, K. Nuida, N. Kanayama, T. Nishide, G. Hanaoka, and E. Okamoto, "Card-based protocols using regular polygon cards," *IEICE Trans. Fundam.*, vol. E100.A, no. 9, pp. 1900–1909, 2017. [Online]. Available: https://doi.org/10.1587/transfun.E100.A.1900

[21] H. Ono and Y. Manabe, "Card-based cryptographic logical computations using private operations," *New Gener. Comput.*, vol. 39, no. 1, pp. 19–40, 2021. [Online]. Available: https://doi.org/10.1007/s00354-020-00113-z

[22] T. Mizuki and H. Shizuya, "A formalization of card-based cryptographic protocols via abstract machine," *Int. J. Inf. Secur.*, vol. 13, no. 1, pp. 15–23, 2014. [Online]. Available: https://doi.org/10.1007/s10207-013-0219-4

[23] A. Koch, M. Schrempp, and M. Kirsten, "Card-based cryptography meets formal verification," in *Advances in Cryptology–ASIACRYPT 2019*, ser. LNCS, S. D. Galbraith and S. Moriai, Eds., vol. 11921. Cham: Springer, 2019, pp. 488–517. [Online]. Available: https://doi.org/10.1007/978-3-030-34578-5_18

[24] K. Shinagawa and K. Nuida, "Card-based protocols imply PSM protocols," in *Theoretical Aspects of Computer Science*, ser. LIPIcs, O. Beyersdorff, M. Pilipczuk, E. Pimentel, and N. K. Thàng, Eds., vol. 327. Dagstuhl: Schloss Dagstuhl, 2025, pp. 72:1–72:18. [Online]. Available: https://doi.org/10.4230/LIPIcs.STACS.2025.72

[25] 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*, ser. LNCS, T. Takagi and T. Peyrin, Eds., vol. 10626. Cham: Springer, 2017, pp. 126–155. [Online]. Available: https://doi.org/10.1007/978-3-319-70700-6_5

[26] K. Fujita, S. Ikeda, K. Shinagawa, and K. Yoneyama, "Formal verification and proof of impossibility for four-card XOR protocols using only random cuts," in *ACM ASIA Public-Key Cryptography Workshop*, K. Emura and H. Morita, Eds. New York: ACM, 2025, pp. 9–16. [Online]. Available: https://doi.org/10.1145/3709015.3728665

[27] S. Iino, S. Ikeda, K. Shinagawa, Y. Li, K. Sakiyama, and D. Miyahara, "Impossibility of four-card AND protocols with a single closed shuffle," in *Cryptology and Network Security*, ser. LNCS, Y. Kim, A. Miyaji, and M. Tibouchi, Eds., vol. 16351. Singapore: Springer, 2025, pp. 213–229. [Online]. Available: https://doi.org/10.1007/978-981-95-4434-9_10

[28] A. Iwasaki, "Minimum number of up-down cards for finite-time committed-AND protocol without interlocking operations," in *Advances in Information and Computer Security*, ser. LNCS, C. Cid and N. Yanai, Eds., vol. 16208. Singapore: Springer, 2025, pp. 210–226. [Online]. Available: https://doi.org/10.1007/978-981-95-4674-9_11

[29] T. Nishida, T. Mizuki, and H. Sone, "Securely computing the three-input majority function with eight cards," in *Theory and Practice of Natural Computing*, ser. LNCS, A.-H. Dediu, C. Martín-Vide, B. Truthe, and M. A. Vega-Rodríguez, Eds., vol. 8273. Berlin, Heidelberg: Springer, 2013, pp. 193–204. [Online]. Available: https://doi.org/10.1007/978-3-642-45008-2_16

[30] T. Mizuki and H. Shizuya, "Computational model of card-based cryptographic protocols and its applications," *IEICE Trans. Fundam.*, vol. E100.A, no. 1, pp. 3–11, 2017. [Online]. Available: https://doi.org/10.1587/transfun.E100.A.3

[31] R. Pass and A. Shelat, *A Course in Cryptography*, 2010.