Card-based Protocol against Actively Revealing Card Attack*

Ken Takashima[1], Daiki Miyahara[1,2], Takaaki Mizuki[3], and Hideaki Sone[3]

[1] Graduate School of Information Sciences, Tohoku University,
6–3–09 Aramaki-Aza-Aoba, Aoba-ku, Sendai 980–8579, Japan
[2] National Institute of Advanced Industrial Science and Technology,
2–4–7, Aomi, Koto-ku, Tokyo, 135–0064, Japan
[3] Cyberscience Center, Tohoku University,
6–3 Aramaki-Aza-Aoba, Aoba-ku, Sendai 980–8578, Japan

Abstract. In 1989, den Boer presented the first card-based protocol, called the "five-card trick" that securely computes the AND function using a deck of physical cards via a series of actions such as shuffling and turning over cards. This protocol enables a couple to confirm their mutual love without revealing their individual feelings. During such a secure computation protocol, it is important to keep any information about the inputs secret. Almost all existing card-based protocols are secure under the assumption that all players participating in a protocol are semi-honest or covert, i.e., they do not deviate from the protocol if there is a chance that they will be caught when cheating. In this paper, we consider a more malicious attack in which a player as an active adversary can reveal cards illegally without any hesitation. Against such an actively revealing card attack, we define the t-secureness, meaning that no information about the inputs leaks even if at most t cards are revealed illegally. Subsequently, we design a 1-secure AND protocol. Thus, our contribution is the construction of the first formal framework to handle actively revealing card attacks and their countermeasures.

Keywords: Cryptography, Card-based protocols, Active security, Secure multiparty computations

1 Introduction

In 1989, den Boer presented the first card-based protocol, called the *five-card trick* that securely computes the AND function using a deck of physical cards [1]. Assuming that Alice has a private bit $a \in \{0,1\}$ and Bob has a private bit $b \in \{0,1\}$, the five-card trick, which uses five cards $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$, proceeds as follows.

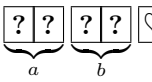
1. According to the encoding rule:

$$\boxed{\clubsuit}\boxed{\heartsuit} = 0 \text{ and } \boxed{\heartsuit}\boxed{\clubsuit} = 1, \tag{1}$$

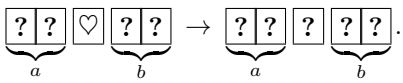
* This paper appears in Proceedings of TPNC 2019. The final authenticated version is available online at https://doi.org/10.1007/978-3-030-34500-6_6.

Alice commits her private bit a to two face-down cards of different colors ($\clubsuit, \heartsuit$) without anyone seeing the order of the two cards:

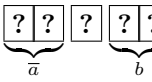
Such a pair of face-down cards is called a *commitment* to a. Similarly, Bob places a commitment to b on the table. Therefore, together with the remaining red card $\boxed{\heartsuit}$, the initial sequence of the five cards is



2. Move the rightmost red card to the center and turn it over:

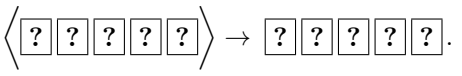


3. Swap the first and second cards (from the left), namely the two cards constituting the commitment to a; owing to the encoding (1), this action performs the NOT operation such that a commitment to the negation $\overline{a}$ of a can be obtained:



It is noteworthy that only when $a = b = 1$, the three cards in the middle will be $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$.

4. Apply a *random cut*, denoted by $\langle\cdot\rangle$; it is a shuffle action to cyclically shift the sequence of cards at random:



The shift offset is uniformly distributed on $\{0, 1, 2, 3, 4\}$, and nobody knows the offset[i].

5. Open all the five cards.
   - If three consecutive red cards $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$ (apart from cyclic rotation) appear, we have $a \wedge b = 1$.
   - If $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$ do not appear, we have $a \wedge b = 0$.

This is the five-card trick, which securely computes the AND function, i.e., it reveals only the value of $a \wedge b$. As an application, for instance, this card-based protocol enables Alice and Bob to confirm their mutual love without revealing their individual feelings.

During such a secure computation protocol, it is important to keep any information about the inputs secret. As seen above, the five-card trick preserves

[i] It is well known that humans can implement a random cut securely [13].

the secrecy of the inputs a, b by virtue of the face-down cards, and the shuffle action eliminates the individual values of the inputs aside from the exact value of $a \wedge b$. In other words, the five-card trick is secure provided that all players obey the protocol. Similar to the five-card trick, almost all existing card-based protocols (e.g., [2, 4, 6, 10, 11]) are secure under the assumption that all players are *semi-honest* or *covert*, i.e., they do not deviate from the protocol if there is a chance that they will be caught when cheating. In most cases, a card-based protocol is executed completely publicly with all eyes fixed on how the cards are manipulated, and hence, any illegal actions by the players (or others) will be noticed [8]; thus, any semi-honest or covert player always follows the protocol.

By contrast, this paper considers a more malicious attack: We assume that one player (e.g., Alice) is an active adversary who may possibly reveal face-down cards illegally without any hesitation. For example, if Alice suddenly reveals the commitment to b at Step 3 during the execution of the five-card trick, Bob's private input will be leaked immediately. We call such a malicious attack the *actively revealing card attack*.

To prevent face-down cards from being revealed illegally, we may place each card into an envelope, as indicated by Koch and Walzer [3]. However, using envelopes is not convenient; hence, we solicit another solution that does not rely on any additional tools such as envelopes. Thus, we have to devise a method to keep individual players' inputs secret even if some of the face-down cards are revealed maliciously. To this end, we borrow an idea from secret sharing schemes [12] such that each input commitment will be split into several "share" commitments. Specifically, as the "revealing-card tolerance," we introduce the concept of "t-secureness" in which any information regarding the inputs will be preserved even if at most t cards are revealed maliciously. Subsequently, we design a 1-secure AND protocol. Thus, our main contribution is to construct the first formal framework to handle actively revealing card attacks and their countermeasures.

This paper focuses on *non-committed format* protocols that specify the output value by revealing some face-down cards, as shown in the five-card trick (or in others, e.g., [6]). By contrast, there are *committed format* protocols that produce commitments (consisting of face-down cards) as the output (e.g., [2,10,11]): Because the output is hidden owing to the face-down cards, during such a committed format protocol, information regarding the input as well as output will not be leaked. Meanwhile, committed format protocols have been formalized well; no formal treatment of non-committed format protocols has been reported (note that because a committed format protocol does not leak any information, it suffices to consider perfect secrecy; meanwhile, a non-committed format protocol needs to leak some information regarding the input to reveal the output value, and hence, a more careful treatment is required). Herein, we first formalize a non-committed format protocol. This formalization is one of our major results.

It is noteworthy that Mizuki and Shizuya [8] previously adopted a similar idea to deal with the situation where some of the cards may be flawed, i.e., the cards may have scuff marks on their backs (undoubtedly, the problem of flawed

cards is different from that of the actively revealing card attack, but they may share some common features). Because this previous work [8] considered only committed format protocols, it is interesting future work to apply the technique proposed herein to design "scuff-proof" non-committed format protocols.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce a formal approach for describing a card-based protocol. In Section 3, we formally define a non-committed format protocol. In Section 4, we define the t-secureness against the actively revealing card attack. In Section 5, we construct a 1-secure AND protocol and confirm its security. Finally, the paper is concluded in Section 6.

2 Preliminaries

In this section, the way to formally describe a card-based protocol is presented.

The computational model of card-based protocols has been formalized via abstract machine [3, 4, 7, 9]. Roughly speaking, a protocol consists of a series of three actions: turn, perm, and shuf actions, along with a sequence of cards.

Consider a sequence of d cards. A turn action is specified by a set $T \subseteq \{1, 2, \ldots, d\}$ of positions of cards; the action (turn, T) turns over every card whose position is in T. A perm action is specified by a permutation $\pi \in S_d$, where S_d denotes the symmetric group of degree d; the action (perm, π) rearranges the positions of d cards according to π. A shuf action is specified by a set $\Pi \subseteq S_d$ of permutations; the action (shuf, Π) probabilistically rearranges the positions of d cards according to a permutation π uniformly drawn from Π. We call a protocol using exactly d cards a *d-card* protocol.

To illustrate, recall the execution of the five-card trick [1] presented in the previous section. It uses two types of cards, $\boxed{\clubsuit}$ and $\boxed{\heartsuit}$, whose backs are $\boxed{?}$. All cards of the same type are indistinguishable. The five-card trick, which is a 5-card protocol, starts with a sequence of five cards:

$$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{b}\boxed{\heartsuit}. \tag{2}$$

Step 2 of the five-card trick is formally captured by $(\mathsf{perm}, (3\ 4\ 5))$ along with $(\mathsf{turn}, \{3\})$. Step 3 is $(\mathsf{perm}, (1\ 2))$. In Step 4, we apply a random cut that can be written as $(\mathsf{shuf}, \mathsf{RC}_5)$, where $\mathsf{RC}_5 = \{(1\ 2\ 3\ 4\ 5)^i \mid 1 \leq i \leq 5\}$. Step 5 is $(\mathsf{turn}, \{1, 2, 3, 4, 5\})$.

To discuss the correctness and security of protocols, we use the concept of *statuses* of a protocol. For example, the initial status of the five-card trick (as in (2)) is described as follows:

$$\begin{aligned}
&\clubsuit\heartsuit\clubsuit\heartsuit\heartsuit && (p_{00}, 0, 0, 0)\\
&\clubsuit\heartsuit\heartsuit\clubsuit\heartsuit && (0, p_{01}, 0, 0)\\
&\heartsuit\clubsuit\clubsuit\heartsuit\heartsuit && (0, 0, p_{10}, 0)\\
&\heartsuit\clubsuit\heartsuit\clubsuit\heartsuit && (0, 0, 0, p_{11}),
\end{aligned}$$

where p_{ij} denotes the probability that input (a, b) is equal to (i, j) for every $(i, j) \in \{0, 1\}^2$; in other words, $(p_{00}, p_{01}, p_{10}, p_{11})$ denotes a probability distribution on the input set $\{0, 1\}^2$. The status above consists of four *entries*, each of which is a pair of a symbol sequence (such as ♣♡♣♡♡) and a *probability trace* (such as $(p_{00}, 0, 0, 0)$); the first entry means that the symbol sequence ♣♡♣♡♡ and the event $(a, b) = (0, 0)$ occur with a probability of p_{00} (and ♣♡♣♡♡ with $(a, b) \neq (0, 0)$ never occurs), the second entry means that ♣♡♡♣♡ and $(a, b) = (0, 1)$ occur with a probability of p_{01}, and so on. The initial status (and succeeding statuses) are transformed into another status by an action as shown in Figure 1. In particular, the turn action results in ten "leaf" statuses.

The expression of protocols in Figure 1 was established by Koch and Walzer [3] where a tree structure specifies a protocol. We modify it slightly using the probability traces introduced by Mizuki and Komano [5]. We call such a tree the *(modified) KWH-tree* of a protocol. Borrowing a terminology in graph theory, we call the bottom statuses in a KWH-tree the *leaf* statuses.

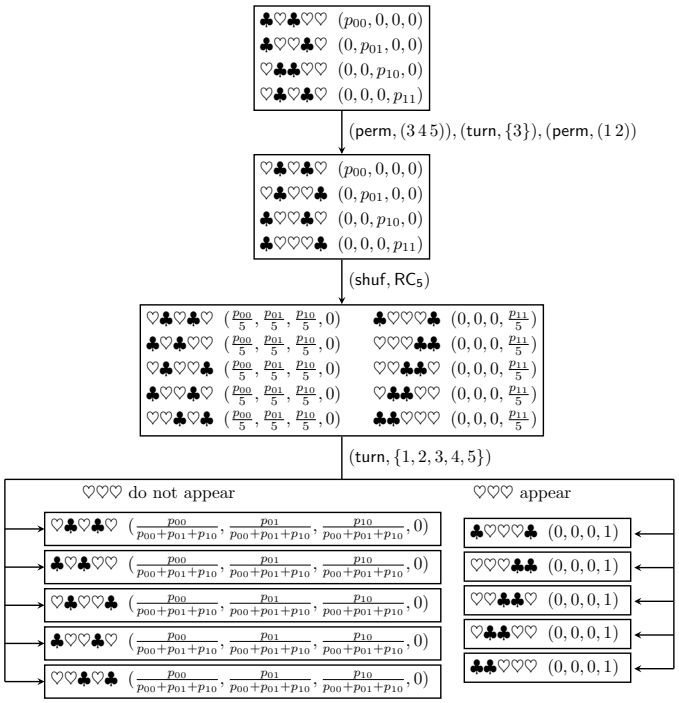


Fig. 1. The (modified) KWH-tree of the five-card trick

Note that in each of the first three statuses (namely, "non-leaf" statuses) depicted in Figure 1, the (coordinate-wise) sum of all probability traces is equal

to $(p_{00}, p_{01}, p_{10}, p_{11})$; this guarantees that no information regarding the input (a, b) will be leaked. Regarding the ten leaf statuses, each of them has only one probability trace, which is either $(\frac{p_{00}}{p_{00}+p_{01}+p_{10}}, \frac{p_{01}}{p_{00}+p_{01}+p_{10}}, \frac{p_{10}}{p_{00}+p_{01}+p_{10}}, 0)$ or $(0, 0, 0, 1)$; this implies that any information other than the value of $a \wedge b$ will not be leaked.

To the best of our knowledge, Figure 1 is the first attempt to depict the KWH-tree of the five-card trick. Because a formal treatment for non-committed format protocols does not exist, we will create such a formal framework, as will be explained in the next section.

3 Formalizing Non-committed Format Protocols

In this section, we formally define a non-committed format protocol for a Boolean function.

First, we define an n-input protocol.

Definition 1 *Let $d \geq 2n$ for an integer $n \geq 2$, and let $\mathcal{P}$ be a d-card protocol. We say that $\mathcal{P}$ is an* n-input protocol *if its initial status consists of the following 2^n entries:*

$$
\begin{array}{ll}
\underbrace{\clubsuit\heartsuit\clubsuit\heartsuit\clubsuit\heartsuit\cdots\clubsuit\heartsuit\clubsuit\heartsuit}_{000\ldots00_2}\ \alpha & (p_0, 0, 0, \ldots, 0, 0) \\
\underbrace{\clubsuit\heartsuit\clubsuit\heartsuit\clubsuit\heartsuit\cdots\clubsuit\heartsuit\heartsuit\clubsuit}_{000\ldots01_2}\ \alpha & (0, p_1, 0, \ldots, 0, 0) \\
\vdots & \\
\underbrace{\heartsuit\clubsuit\heartsuit\clubsuit\heartsuit\clubsuit\cdots\heartsuit\clubsuit\heartsuit\clubsuit}_{111\ldots11_2}\ \alpha & (0, 0, 0, \ldots, 0, p_{2^n-1})
\end{array}
$$

where α is any symbol sequence of length $d-2n$. Here, p_i, $0 \leq i \leq 2^n-1$, is the probability that the n-bit input is equal to the binary expression of i. Furthermore, we call the tuple $(p_0, \ldots, p_{2^n-1})$ an input distribution.

As shown in Definition 1, we implicitly assume a one-to-one mapping between $\{0,1\}^n$ and $\{0, 1, \ldots, 2^n - 1\}$. Thus, throughout this paper, if we write q_b for $b \in \{0,1\}^n$ and a tuple $(q_0, \ldots, q_{2^n-1})$, we regard the subscription b as the corresponding decimal number.

Next, we define some properties regarding the statuses.

Definition 2 *Let $\mathcal{P}$ be an n-input protocol with an input distribution $(p_0, \ldots, p_{2^n-1})$, and consider a Boolean function $f : \{0,1\}^n \to \{0,1\}$.*

- *A status S of $\mathcal{P}$ is called an* opaque status *if the (coordinate-wise) sum of its probability traces is equal to $(p_0, \ldots, p_{2^{n-1}})$.*

– *We say that a status S is an* output-0 status *if the sum of its probability traces $(q_0, \ldots, q_{2^n-1})$ satisfies*

$$\begin{cases} q_b = \frac{p_b}{\sum_{i \in f^{-1}(0)} p_i} & \text{if } f(b) = 0 \\ q_b = 0 & \text{if } f(b) = 1 \end{cases}$$

for every $b \in \{0,1\}^n$, where $f^{-1}(0)$ is the preimage of 0 under f.
– *We say that a status S is an* output-1 status *if the sum of its probability traces $(q_0, \ldots, q_{2^n-1})$ satisfies*

$$\begin{cases} q_b = 0 & \text{if } f(b) = 0 \\ q_b = \frac{p_b}{\sum_{i \in f^{-1}(1)} p_i} & \text{if } f(b) = 1 \end{cases}$$

for every $b \in \{0,1\}^n$.

We are now ready to formally define a non-committed format protocol.

Definition 3 *Let $\mathcal{P}$ be an n-input protocol, and let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. We say that $\mathcal{P}$* works for f in a non-committed format *if the following holds:*

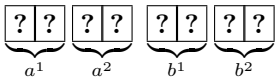
– *every leaf status is either an output-*0 *status or an output-*1 *status, and all other statuses are opaque;*
– *the expected height of its KWH-tree is finite.*

One can easily verify that the five-card trick satisfies Definition 3.

4 Defining Revealing-Card Tolerance

As mentioned before, this paper considers an active attack where an adversary can reveal some cards without obeying a protocol. Because the execution of a protocol is conducted publicly, it is difficult for an adversary to illegally reveal many cards simultaneously. Thus, we assume that such a malicious adversary can reveal at most t cards at most once when the protocol is executed.

Note that any n-input protocol (defined in Definition 1) cannot be "secure" against the actively revealing card attack because the adversary can reveal some cards that constitute the input commitments to obtain the secret values immediately after the protocol starts. Therefore, the input commitments must be masked. To achieve this, we borrow an idea from secret sharing schemes. Hence, instead of directly placing commitments to their private bits, Alice places two commitments to $a^1, a^2 \in \{0,1\}$ such that $a = a^1 \oplus a^2$, where Alice's private bit a is split into a^1 and a^2 randomly, and Bob places two commitments similarly:



.

For such an input sequence, even if at most one card is revealed illegally, the values of a and b will not be leaked. By further extending this, we have an $(n, t+1)$-input protocol, as in the following Definition 4. Hereinafter, for $b \in \{0,1\}^n$, $b[i]$ denotes the i-th bit (of the n-bit sequence b).

Definition 4 *Let $d \geq 2n(t+1)$ for integers $n \geq 2$ and $t \geq 1$, and let $\mathcal{P}$ be a d-card protocol. We say that $\mathcal{P}$ is an* $(n, t+1)$-input protocol *if its initial status consists of all entries in $\bigcup_{b\in\{0,1\}^n} \mathcal{E}_b$ such that*

$$\mathcal{E}_b = \Big\{ (x_1^1 \dots x_1^{t+1} x_2^1 \dots x_2^{t+1} \dots x_n^1 \dots x_n^{t+1} \alpha,\ (0, \dots, 0, \frac{p_b}{n2^t}, 0, \dots, 0)) \\ \Big| \bigoplus_{j=1}^{t+1} x_i^j = b[i], 1 \leq i \leq n \Big\}$$

for every $b \in \{0,1\}^n$, where $x_i^j \in \{0,1\}$ is interpreted as a pair of symbols based on the encoding: $0 = \clubsuit\heartsuit$ and $1 = \heartsuit\clubsuit$, and α is any symbol sequence of length $d - 2n(t+1)$.

We are now ready to define the "t-secureness" as in the following Definition 6 along with Definition 5.

Definition 5 *Let $\mathcal{P}$ be an $(n, t+1)$-input protocol, and let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. We define opaque, output-0, and output-1 statuses similarly as in Definition 2. Additionally, we define "working for f" similarly to Definition 3.*

Definition 6 *Let $\mathcal{P}$ be an $(n, t+1)$-input protocol working for a Boolean function f in a non-committed format. We say that $\mathcal{P}$ is* t-secure *if any resulting status from applying any action (turn, T) with $|T| \leq t$ to every status of $\mathcal{P}$ is either an opaque status, an output-0 status, or an output-1 status.*

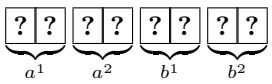
5 Our 1-Secure AND Protocol

We describe the construction of a 1-secure AND protocol in this section. In Section 5.1, we present its outline; our protocol consists of the setup, first, second, and third phases. In Sections 5.2, 5.3, and 5.4, we provide the details of the first, second, and third phases, respectively.

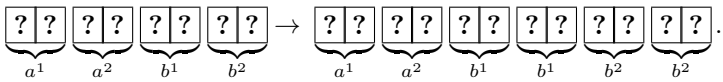
5.1 Outline of Our Protocol

Because we wish to design a 1-secure AND computation of two variables (namely, $n = 2$ and $t = 1$), we should use a $(2, 2)$-input protocol. Therefore, Alice and Bob create $a^1, a^2, b^1, b^2 \in \{0,1\}$ such that $a = a^1 \oplus a^2$ and $b = b^1 \oplus b^2$ as input. Thus, it suffices to compute $(a^1 \wedge b^1) \oplus (a^1 \wedge b^2) \oplus (a^2 \wedge b^1) \oplus (a^2 \wedge b^2) = a \wedge b$. To this end, our protocol proceeds as follows.

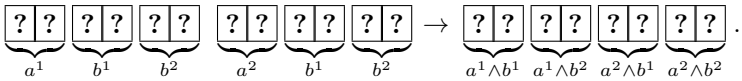
Setup phase. Satisfying Definition 4, Alice places two commitments to a^1, a^2, and Bob places two commitments to b^1, b^2:



First phase. Make two copied commitments to each of b^1 and b^2 using the existing COPY protocol [10]:



Second phase. From the commitments to a^1, b^1, b^2, make commitments to $a^1 \wedge b^1, a^1 \wedge b^2$ using the existing AND protocol [8]; similarly, from the commitments to a^2, b^1, b^2, make commitments to $a^2 \wedge b^1, a^2 \wedge b^2$:

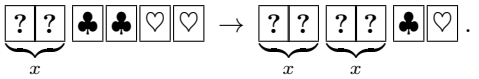


Third phase. Compute $(a^1 \wedge b^1) \oplus (a^1 \wedge b^2) \oplus (a^2 \wedge b^1) \oplus (a^2 \wedge b^2)$.

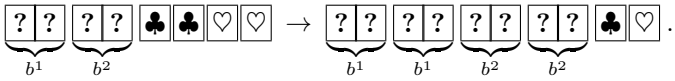
Here, we analyze the security of the setup phase. There are 16 possibilities for $(a^1, a^2, b^1, b^2) \in \{0,1\}^4$, and hence, the initial status can be written as the first four columns and the last column in Table 1. (Note that any action $(\mathsf{turn}, \{i\})$ reveals at most one bit among four bits.) One can easily confirm that any action $(\mathsf{turn}, \{i\})$ results in an opaque status.

5.2 First Phase

In this phase, we duplicate the commitments to b^1 and b^2. To this end, we use the existing COPY protocol [10], which performs the following (refer to [10] for the details):



By executing the COPY protocol twice, we have



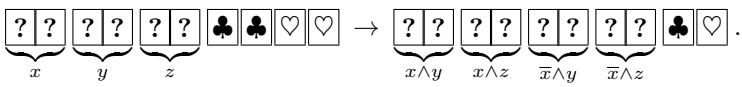
During this first phase, any action $(\mathsf{turn}, \{i\})$ reveals at most one bit among b^1 and b^2; hence, similar to the setup phase, any resulting status from an illegal reveal will be opaque. (We omit the details owing to page limitations.)

Table 1. Essential truth table for deriving statuses of our protocol.

a^1	a^2	b^1	b^2	$a^1 \wedge b^1$	$a^1 \wedge b^2$	$a^2 \wedge b^1$	$a^2 \wedge b^2$	$\overline{a^1} \wedge b^1$	$\overline{a^1} \wedge b^2$	$\overline{a^2} \wedge b^1$	$\overline{a^2} \wedge b^2$	Prob. trace
0	0	0	0	0	0	0	0	0	0	0	0	$(p_{00}/4, 0, 0, 0)$
0	0	1	1	0	0	0	0	1	1	1	1	$(p_{00}/4, 0, 0, 0)$
1	1	0	0	0	0	0	0	0	0	0	0	$(p_{00}/4, 0, 0, 0)$
1	1	1	1	1	1	1	1	0	0	0	0	$(p_{00}/4, 0, 0, 0)$
0	0	0	1	0	0	0	0	0	1	0	1	$(0, p_{01}/4, 0, 0)$
0	0	1	0	0	0	0	0	1	0	1	0	$(0, p_{01}/4, 0, 0)$
1	1	0	1	0	1	0	1	0	0	0	0	$(0, p_{01}/4, 0, 0)$
1	1	1	0	1	0	1	0	0	0	0	0	$(0, p_{01}/4, 0, 0)$
0	1	0	0	0	0	0	0	0	0	0	0	$(0, 0, p_{10}/4, 0)$
0	1	1	1	0	0	1	1	1	1	0	0	$(0, 0, p_{10}/4, 0)$
1	0	0	0	0	0	0	0	0	0	0	0	$(0, 0, p_{10}/4, 0)$
1	0	1	1	1	1	0	0	0	0	1	1	$(0, 0, p_{10}/4, 0)$
0	1	0	1	0	0	0	1	0	1	0	0	$(0, 0, 0, p_{11}/4)$
0	1	1	0	0	0	1	0	1	0	0	0	$(0, 0, 0, p_{11}/4)$
1	0	0	1	0	1	0	0	0	0	0	1	$(0, 0, 0, p_{11}/4)$
1	0	1	0	1	0	0	0	0	0	1	0	$(0, 0, 0, p_{11}/4)$

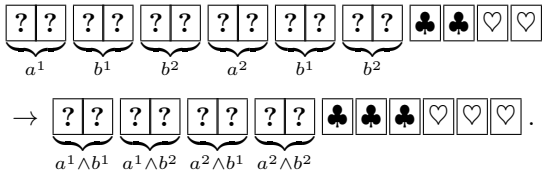
5.3 Second Phase

In this phase, we use the existing AND protocol [8]:



We destroy the commitments to $\overline{x} \wedge y$ and $\overline{x} \wedge z$ by shuffling each of them.

By executing the AND protocol twice, we have

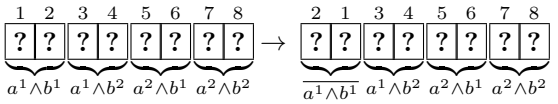


During this second phase, any action $(\mathsf{turn}, \{i\})$ reveals at most one bit among a^1, a^2, b^1, b^2, $a^1 \wedge b^1, a^1 \wedge b^2, a^2 \wedge b^1, a^2 \wedge b^2$, $\overline{a^1} \wedge b^1, \overline{a^1} \wedge b^2, \overline{a^2} \wedge b^1, \overline{a^2} \wedge b^2$; Table 1 implies that any illegal resulting status will be opaque.

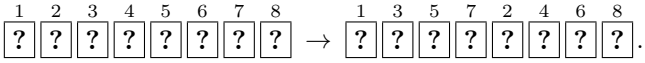
5.4 Third Phase

In this phase, we compute $(a^1 \wedge b^1) \oplus (a^1 \wedge b^2) \oplus (a^2 \wedge b^1) \oplus (a^2 \wedge b^2)$ from the commitments to $a^1 \wedge b^1, a^1 \wedge b^2, a^2 \wedge b^1, a^2 \wedge b^2$. Our "4-bit XOR subprotocol" proceeds as follows.

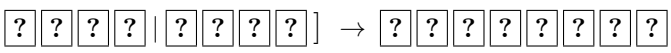
1. Negate the commitment to $a^1 \wedge b^1$ by $(\mathsf{perm}, (1\ 2))$:



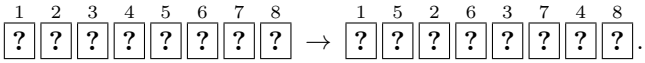
2. Rearrange the sequence of the eight cards by $(\mathsf{perm}, (2\ 5\ 3)(4\ 6\ 7))$:



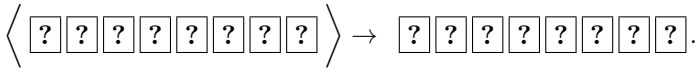
3. Apply a random bisection cut [10], denoted by $[\cdot|\cdot]$, which is the shuffle action $(\mathsf{shuf}, \{\mathsf{id}, (1\,5)(2\,6)(3\,7)(4\,8)\})$:



4. Apply $(\mathsf{perm}, (2\,3\,5)(4\,7\,6))$, which is the inverse permutation of Step 2:



5. Apply a random cut, namely $(\mathsf{shuf}, \mathsf{RC}_8)$, where RC_8 is defined similarly to RC_5:



6. Reveal all the cards by $(\mathsf{turn}, \{1, 2, 3, 4, 5, 6, 7, 8\})$. Count the commitments[ii]:
   - If the number of commitments to 1 is odd, $a \wedge b = 0$.
   - If the number of commitments to 1 is even, $a \wedge b = 1$.

Owing to page limitations, we omit the KWH-tree of our XOR subprotocol, which implies the correctness and secrecy.

6 Conclusion

In this paper, we first described the KWH-tree of the five-card trick and formally defined non-committed protocols. Against the actively revealing card attack, we defined the t-secureness and presented a 1-secure AND protocol.

Acknowledgment.

This work was supported by JSPS KAKENHI Grant Numbers JP17K00001 and JP19J21153. We would like to thank the anonymous reviewers for their fruitful comments.

[ii] We can specify the boundary between commitments by the color of consecutive cards. For example, if we obtain a sequence ♣♣♡♣♡♣♡♡, we can place delimiters in the middle of each of ♣♣ and ♡♡ as ♣|♣♡♣♡♣♡|♡; hence, we have ♣♡|♣♡|♣♡|♡♣.

References

1. den Boer, B.: More efficient match-making and satisfiability the five card trick. In: Quisquater, J.J., Vandewalle, J. (eds.) Advances in Cryptology — EUROCRYPT '89. Lecture Notes in Computer Science, vol. 434, pp. 208–217. Springer, Berlin, Heidelberg (1990)
2. Ishikawa, R., Chida, E., Mizuki, T.: Efficient card-based protocols for generating a hidden random permutation without fixed points. In: Calude, C.S., Dinneen, M.J. (eds.) Unconventional Computation and Natural Computation. Lecture Notes in Computer Science, vol. 9252, pp. 215–226. Springer, Cham (2015)
3. Koch, A., Walzer, S.: Foundations for actively secure card-based cryptography. Cryptology ePrint Archive, Report 2017/423 (2017), https://eprint.iacr.org/2017/423
4. Koch, A., Walzer, S., Härtel, K.: Card-based cryptographic protocols using a minimal number of cards. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology – ASIACRYPT 2015. Lecture Notes in Computer Science, vol. 9452, pp. 783–807. Springer, Berlin, Heidelberg (2015)
5. Mizuki, T., Komano, Y.: Analysis of information leakage due to operative errors in card-based protocols. In: Iliopoulos, C., Sung, W., Leong, H.W. (eds.) Combinatorial Algorithms. Lecture Notes in Computer Science, vol. 10979, pp. 250–262. Springer, Cham (2018)
6. Mizuki, T., Kumamoto, M., Sone, H.: The five-card trick can be done with four cards. In: Wang, X., Sako, K. (eds.) Advances in Cryptology – ASIACRYPT 2012. Lecture Notes in Computer Science, vol. 7658, pp. 598–606. Springer, Berlin, Heidelberg (2012)
7. Mizuki, T., Shizuya, H.: A formalization of card-based cryptographic protocols via abstract machine. International Journal of Information Security **13**(1), 15–23 (2014). https://doi.org/10.1007/s10207-013-0219-4
8. Mizuki, T., Shizuya, H.: Practical card-based cryptography. In: Ferro, A., Luccio, F., Widmayer, P. (eds.) Fun with Algorithms. Lecture Notes in Computer Science, vol. 8496, pp. 313–324. Springer, Cham (2014)
9. Mizuki, T., Shizuya, H.: Computational model of card-based cryptographic protocols and its applications. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E100.A**(1), 3–11 (2017). https://doi.org/10.1587/transfun.E100.A.3
10. Mizuki, T., Sone, H.: Six-card secure AND and four-card secure XOR. In: Deng, X., Hopcroft, J.E., Xue, J. (eds.) Frontiers in Algorithmics. Lecture Notes in Computer Science, vol. 5598, pp. 358–369. Springer, Berlin, Heidelberg (2009)
11. Niemi, V., Renvall, A.: Secure multiparty computations without computers. Theoretical Computer Science **191**(1–2), 173–183 (1998). https://doi.org/10.1016/S0304-3975(97)00107-2
12. Shamir, A.: How to share a secret. In: Ashenhurst, R.L. (ed.) Communications of the ACM. vol. 22, pp. 612–613. ACM New York, NY, USA (1979). https://doi.org/10.1145/359168.359176
13. Ueda, I., Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: How to implement a random bisection cut. In: Martín-Vide, C., Mizuki, T., Vega-Rodríguez, M.A. (eds.) Theory and Practice of Natural Computing. Lecture Notes in Computer Science, vol. 10071, pp. 58–69. Springer, Cham (2016)