Practical Card-based Cryptography[*]

Takaaki Mizuki Hiroki Shizuya

Tohoku University
tm-paper+cardmaliw[atmark]g-mail.tohoku-university.jp

Abstract

It is known that secure multi-party computations can be achieved using a number of black and red physical cards (with identical backs). In previous studies on such card-based cryptographic protocols, typically an ideal situation where all players are semi-honest and all cards of the same suit are indistinguishable from one another was assumed. In this paper, we consider more realistic situations where, for example, some players possibly act maliciously, or some cards possibly have scuff marks, so that they are distinguishable, and propose methods to maintain the secrecy of players' private inputs even under such severe conditions.

1 Introduction

It is known that secure multi-party computations can be conducted using a number of black ($\boxed{\clubsuit}$) and red ($\boxed{\heartsuit}$) physical cards with identical backs ($\boxed{?}$). Indeed, as listed in Table 1, several *card-based cryptographic protocols* have been invented thus far for secure computations, such as secure AND and XOR. In previous studies on such card-based protocols, typically an ideal situation where all players are semi-honest and all cards of the same color are indistinguishable from one another was assumed. In contrast, this paper considers more realistic situations where, for example, some players act maliciously, or some cards have scuff marks (scratches) so that they are distinguishable.

This paper begins with a review of the "five-card trick [3]," the first card-based protocol.

1.1 Five-card Trick

The five-card trick, invented in 1989 by den Boer, securely computes the AND function using five cards [3]. Before introducing the details of the protocol, we present some notations.

To deal with Boolean values, we fix an encoding rule using a pair of cards as

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

[*] This paper appears in Proceedings of FUN 2014. The final publication is available at link.springer.com.

Table 1: Existing card-based protocols

	No. of colors	No. of cards	Avg. no. of trials
○ Non-committed-format AND			
den Boer [3] (§1.1)	2	5	1
Mizuki-Kumamoto-Sone [8]	2	4	1
○ Committed-format AND			
Crépeau-Kilian [2]	4	10	6
Niemi-Renvall [10]	2	12	2.5
Stiglic [13]	2	8	2
Mizuki-Sone [7] (§2.1)	2	6	1
○ Committed-format XOR			
Crépeau-Kilian [2]	4	14	6
Mizuki-Uchiike-Sone [9]	2	10	2
Mizuki-Sone [7] (§2.2)	2	4	1
○ Committed-format half adder			
Mizuki-Asiedu-Sone [5]	2	8	1
○ Committed-format full adder			
Mizuki-Asiedu-Sone [5]	2	10	1
○ Committed-format 3-variable symmetric-function evaluation			
Nishida-Mizuki-Sone [11]	2	8	1

For a bit $x \in \{0,1\}$, when two face-down cards $\boxed{?}\boxed{?}$ have a value equaling x according to the encoding (1) above, the pair of these face-down cards is called a *commitment* to x, and is written as

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

Now, assume that Alice, holding a bit $a \in \{0,1\}$, and Bob, holding a bit $b \in \{0,1\}$, together want to securely compute the conjunction $a \wedge b$, i.e., they wish to learn only the value of $a \wedge b$. The five-card trick [3] achieves this as follows.

1. Alice privately arranges a commitment to negation $\bar{a}$ of bit a, and Bob privately arranges a commitment to b. These two commitments together with a red card are put forth:

   $$\underbrace{\boxed{?}\boxed{?}}_{\bar{a}}\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{b} \rightarrow \underbrace{\boxed{?}\boxed{?}}_{\bar{a}}\boxed{?}\underbrace{\boxed{?}\boxed{?}}_{b}.$$

   It should be noted that the three middle cards would be $\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$ only if $a = b = 1$.

2. Alice and Bob apply a *random cut*, which is denoted by $\langle\cdot\rangle$, to the sequence of five cards:

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

A random cut means a cyclic shuffling operation; Alice and Bob can implement it by cutting the deck in turn until they are satisfied that the cards have been adequately shuffled.

3. Reveal all of the five cards; then, we have either three (cyclically) consecutive $\boxed{\heartsuit}$'s or not:

$$\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit} \text{ or } \boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}.$$

The former case implies $a \wedge b = 1$ and the latter implies $a \wedge b = 0$.

1.2 Other Existing Protocols

As seen in the previous subsection, the five-card trick [3] developed in 1989 performs a secure AND computation with five cards. In 2012, it was proved that the same cryptographic approach can be conducted with four cards [8] (see Table 1 again).

All the remaining protocols in Table 1 are, however, "committed format." Committed-format protocols are those that produce their output as commitments; for example, AND protocols [2, 7, 10, 13] and XOR protocols [2, 7, 9] generate the commitments

$$\underbrace{\boxed{?}\boxed{?}}_{a \wedge b} \text{ and } \underbrace{\boxed{?}\boxed{?}}_{a \oplus b},$$

respectively, without revealing the values of inputs a and b. It should be noted that any protocol in Table 1 whose average number of trials is more than one is a Las Vegas algorithm. We introduce the existing efficient AND and XOR protocols [7] in Section 2.

A secure NOT computation is trivial, i.e., only swapping the two cards of a commitment yields the negation

$$\underbrace{\boxed{?}\boxed{?}}_{x} \rightarrow \overbrace{\boxed{?}\boxed{?}}^{\rightleftharpoons} \rightarrow \underbrace{\boxed{?}\boxed{?}}_{\bar{x}}.$$

In addition, there are protocols for copying a commitment [2, 7, 10]. Therefore, obviously, by combining these AND/XOR/NOT and copy protocols, one can construct a card-based protocol for any given (multi-valued multiple-variable) function provided that many cards are available.

Further, there are some efficient protocols designed only for specific functions, such as the adder and the majority function [5, 11]. A formal mathematical model for card-based protocols appears in [6].

1.3 Semi-honest Model

As seen in the execution of the five-card trick, introduced above in Section 1.1, when executing a card-based protocol, all players gather at the same place and publicly apply operations, such as flipping cards over and making random cuts, to the deck of cards in cooperation. Therefore, basically, it is very difficult for any player to deviate from the protocol, and hence, all the players are typically assumed to be semi-honest.

For example, in the case of the five-card trick, if the commitments to the input values are put correctly in step 1 and a random cut is applied correctly in step 2, then the outcome in step 3 must be information-theoretically secure, that is, only the value of $a \wedge b$ becomes public and no other information leaks.

As mentioned above, the assumption that a protocol is always executed correctly with all eyes fixed on how the cards are manipulated after all players place commitments on the table as their input is natural[1]. However, in the case of a commitment that is supposed to be placed according to every player's private bit, a player may be able to act maliciously. For instance, ignoring the encoding rule (1), Alice might place two cards of the same color ($\boxed{\clubsuit}\boxed{\clubsuit}$ or $\boxed{\heartsuit}\boxed{\heartsuit}$) with their faces down on the table. This paper addresses such an active attack, and its countermeasure is discussed in Section 3.

1.4 Our Main Results

The main results of this paper are as follows. In Section 3, taking the five-card trick as an example, we demonstrate that an attack that exploits the input format as mentioned above is possible and then propose a general way to prevent such an attack. In Section 4, we discuss the advantages and disadvantages when the cards were manufactured such that the pattern on their back sides is rotationally symmetric. In Section 5, we deal with an issue where some cards possibly have scuff marks on their backs so that they are distinguishable, and propose methods to maintain secrecy under such a severe condition.

Section 2 is devoted to a review of the existing committed-format protocols, and Section 6 concludes the paper.

2 Existing Committed-format AND/XOR Protocols

In this section, we introduce Mizuki-Sone's AND and XOR protocols [7], which are the best among the currently known committed-format protocols (recall Table 1). As seen below, the results of this paper are partially based on the idea behind these protocols.

First, we present some notations. For a pair of bits (x, y), define operations get and shift as

$$\mathsf{get}^0(x,y) = x, \quad \mathsf{get}^1(x,y) = y;$$
$$\mathsf{shift}^0(x,y) = (x,y), \quad \mathsf{shift}^1(x,y) = (y,x).$$

That is, $\mathsf{get}^0(x,y)$ returns the first bit of the pair, $\mathsf{get}^1(x,y)$ returns the second bit, $\mathsf{shift}^0(x,y)$ returns the pair without changing it, and $\mathsf{shift}^1(x,y)$ swaps the pair. Using these notations, we can write

$$a \wedge b = \mathsf{get}^{a\oplus r}(\mathsf{shift}^r(0,b)) \tag{2}$$

where $r \in \{0,1\}$ is an arbitrary bit. In addition, for two bits x and y, the expression

$$\underbrace{\boxed{?}\boxed{?}\boxed{?}\boxed{?}}_{(x,y)}$$

means

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

[1] We assume that no player has the skills of a professional magician.

2.1 AND protocol

Given commitments to a and b together with two additional cards, Mizuki-Sone's AND protocol [7] produces a commitment to $a \wedge b$, as follows.

1. In addition to the two commitments, arrange a commitment to 0:

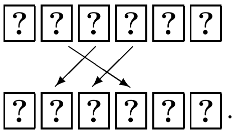
$$\underbrace{\boxed{?}\boxed{?}}_{a}\boxed{\clubsuit}\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{b} \rightarrow \underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{0}\underbrace{\boxed{?}\boxed{?}}_{b},$$

which can be written as

$$\underbrace{\boxed{?}}_{a}\;\underbrace{\boxed{?}}_{\bar{a}}\;\underbrace{\boxed{?}\boxed{?}}_{0}\underbrace{\boxed{?}\boxed{?}}_{b}$$

where a single-card encoding, $\boxed{\clubsuit} = 0$, $\boxed{\heartsuit} = 1$, is used for the sake of convenience.

2. Rearrange the order of the sequence as



3. Bisect the sequence of six cards, and switch them randomly (we call it a *random bisection cut* [7] denoted by $[\,\cdot\,|\,\cdot\,]$):

$$\underbrace{\boxed{?}}_{a}\;\underbrace{\boxed{?}\boxed{?}}_{0}\;\underbrace{\boxed{?}}_{\bar{a}}\;\underbrace{\boxed{?}\boxed{?}}_{b}$$

$$\downarrow$$

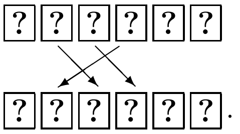
$$\left[\boxed{?}\boxed{?}\boxed{?}\,\middle|\,\boxed{?}\boxed{?}\boxed{?}\right]$$

$$\downarrow$$

$$\underbrace{\boxed{?}}_{a}\;\underbrace{\boxed{?}\boxed{?}}_{0}\;\underbrace{\boxed{?}}_{\bar{a}}\;\underbrace{\boxed{?}\boxed{?}}_{b} \quad \text{or} \quad \underbrace{\boxed{?}}_{\bar{a}}\;\underbrace{\boxed{?}\boxed{?}}_{b}\;\underbrace{\boxed{?}}_{a}\;\underbrace{\boxed{?}\boxed{?}}_{0},$$

where each case occurs with the probability of 1/2.

4. Rearrange the order of the sequence as follows:



Then, we have

$$\underbrace{\boxed{?}\boxed{?}}_{a\oplus r}\underbrace{\boxed{?}\boxed{?}\boxed{?}\boxed{?}}_{\mathsf{shift}^r(0,b)}$$

where r is a (uniformly distributed) random bit because of the random bisection cut.

5. Reveal the first two cards from the left; then, the value of $a \oplus r$ together with Eq. (2) tells us the position of the desired commitment to $a \wedge b$:

$$\boxed{\clubsuit}\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{a\wedge b}\boxed{?}\boxed{?} \quad \text{or} \quad \boxed{\heartsuit}\boxed{\clubsuit}\boxed{?}\boxed{?}\underbrace{\boxed{?}\boxed{?}}_{a\wedge b}.$$

Since r is random, no information about bit a leaks. In addition, the two face-up cards are available for another computation. It should be noted, furthermore, that the other pair of two face-down cards is a commitment to $\bar{a} \wedge b$.

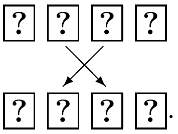
2.2 XOR protocol

Mizuki-Sone's XOR protocol [7] produces a commitment to $a \oplus b$ without any additional card, as follows.

1. Arrange two commitments as

$$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{b}.$$

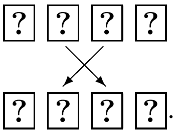
2. Rearrange the order of the sequence as



3. Apply a random bisection cut

$$\left[\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\right] \rightarrow \boxed{?}\boxed{?}\boxed{?}\boxed{?}.$$

4. Rearrange the order of the sequence again as



Then, we have

$$\underbrace{\boxed{?}\boxed{?}}_{a\oplus r}\underbrace{\boxed{?}\boxed{?}}_{b\oplus r}$$

where r is a random bit.

5. Reveal the leftmost two cards; then, we know whether $r = a$ or $r = \bar{a}$, and we have

$$\boxed{\clubsuit}\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{a\oplus b} \quad \text{or} \quad \boxed{\heartsuit}\boxed{\clubsuit}\underbrace{\boxed{?}\boxed{?}}_{\overline{a\oplus b}},$$

and hence, we obtain a commitment to $a \oplus b$. (Note that the secure NOT computation can transform a commitment to $\overline{a \oplus b}$ into one to $a \oplus b$.)

3 Attack Exploiting Input Format

This section addresses an “injection attack” type problem, namely, the issue where an input that does not follow the encoding rule (1) is given to a protocol. In Section 3.1, we illustrate how the attack succeeds by considering the five-card trick as an example. In Section 3.2, we present a general method for preventing such an attack.

3.1 Example of the Attack

Consider the five-card trick explained in Section 1.1, and suppose that Bob is honest but Alice is malicious. Then, assume that Alice placed two cards $\boxed{\clubsuit}\boxed{\clubsuit}$ of the same color with their face down on the table, which is not in a correct format for a commitment (encoding (1)). That is, the sequence of five cards in step 1 of the protocol satisfies

$$\underset{\clubsuit}{\boxed{?}}\underset{\clubsuit}{\boxed{?}}\underset{\heartsuit}{\boxed{?}}\underbrace{\boxed{?}\boxed{?}}_{b},$$

where a mark denoting its color is attached below a card for the sake of convenience.

Hence, if $b = 1$, two red cards $\boxed{\heartsuit}\boxed{\heartsuit}$ would be consecutive; if $b = 0$, they would not be. Therefore, after all five cards are revealed in step 3, their order will tell us the value of Bob’s private bit b (further, the protocol does not terminate successfully).

One possible way to prevent such an attack might be to hand only one pair of a black card and a red one to Alice; however, it is possible that Alice could conceal her action when she makes her commitment, and hence, the situation where she is able to input an injection $\boxed{\clubsuit}\boxed{\clubsuit}$ covertly, having obtained another black card $\boxed{\clubsuit}$ from somewhere, may reasonably occur.

3.2 Countermeasure

Here, we give a general method to avoid the attack described in the previous subsection. The basic idea is simple: we check that the two cards placed on the table by each player satisfy the encoding rule (1). The method proposed below is based on the idea behind the XOR protocol [7], introduced in Section 2.2.

Assume that we want to check that the two cards

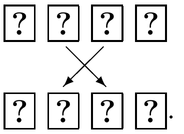
$$\underset{\alpha_1}{\boxed{?}}\underset{\alpha_2}{\boxed{?}}$$

placed by Alice comprise a black card and a red one (where α_1 and α_2 denote the marks of colors). Adding two cards $\boxed{\clubsuit}\boxed{\heartsuit}$, we execute the following procedure.

1. Arrange Alice’s input and a commitment to 0 as

$$\underset{\alpha_1}{\boxed{?}}\underset{\alpha_2}{\boxed{?}}\boxed{\clubsuit}\boxed{\heartsuit} \;\rightarrow\; \underset{\alpha_1}{\boxed{?}}\underset{\alpha_2}{\boxed{?}}\underbrace{\boxed{?}\boxed{?}}_{0}.$$

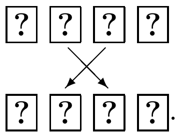
2. Rearrange the order as



3. Apply a random bisection cut

$$\left[\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\right] \rightarrow \boxed{?}\boxed{?}\boxed{?}\boxed{?}.$$

4. Rearrange the order again as



Then, we have

$$\underset{\mathsf{SHIFT}^r(\alpha_1,\alpha_2)}{\boxed{?}\boxed{?}}\quad\underbrace{\boxed{?}\boxed{?}}_{r},$$

where r is a random bit, and furthermore, the order of the leftmost two cards is α_1, α_2 if $r = 0$; and α_2, α_1 if $r = 1$. It should be noted that, if Alice placed a commitment (in a correct format) as her input, then it would be

$$\underbrace{\boxed{?}\boxed{?}}_{a\oplus r}\underbrace{\boxed{?}\boxed{?}}_{r}.$$

5. Reveal the leftmost two cards. If the two face-up cards are $\boxed{\clubsuit}\boxed{\clubsuit}$ or $\boxed{\heartsuit}\boxed{\heartsuit}$, then Alice must have acted maliciously. Otherwise, Alice placed the commitment in a correct format, and hence,

$$\boxed{\clubsuit}\boxed{\heartsuit}\underbrace{\boxed{?}\boxed{?}}_{a}\quad\text{or}\quad\boxed{\heartsuit}\boxed{\clubsuit}\underbrace{\boxed{?}\boxed{?}}_{\bar{a}};$$

consequently, we keep a commitment to a without leaking any information about a (it was only a secure XOR computation of a and 0).

Given two face-down cards placed by a player, this procedure allows us to determine whether they follow the format correctly or not, and in the former case, no information about the commitment leaks.

4 Backs with a Rotationally Symmetric Pattern

As seen thus far, any cards used in the previous work have non-rotationally symmetric patterns, such as $\boxed{\clubsuit}$ or $\boxed{\heartsuit}$ (for face sides) and $\boxed{?}$ (for back sides). Therefore, during the execution of a protocol, players can easily arrange all cards in the same (up/down) direction; usually, people arrange them so that the bottom edge of every card is down. (Actually, as seen below, a bottom-edge-up card, such as $\boxed{\rotatebox{180}{$\clubsuit$}}$, possibly leaks some information.)

In this section, we discuss the advantages and disadvantages of the cards being manufactured such that the pattern on their back sides is rotationally symmetric, such as $\boxed{}$ (plain-colored backs). In particular, in Section 4.1, we demonstrate that indeed such a card possibly leaks information about a player's private input. However, since such a (single) card can hold information with up/down directions, it enables us to construct a protocol with fewer colors and fewer cards, as shown in Section 4.2.

4.1 Disadvantage

Consider the case where Alice and Bob execute the five-card trick with a deck of cards whose backs are rotationally symmetric, such as □.

When Alice makes a commitment to $\bar{a}$, suppose that she places two face-down cards □□ on the table so that the bottom edge of the first (namely, leftmost) card is up (like $\boxed{\rotatebox{180}{$\clubsuit$}}$ or $\boxed{\rotatebox{180}{$\spadesuit$}}$):

$$\underbrace{\boxed{}\boxed{}}_{\bar{a}}\boxed{}\underbrace{\boxed{}\boxed{}}_{b}.$$

If the bottom edges of the remaining four cards are all down (like $\boxed{\clubsuit}$ or $\boxed{\heartsuit}$), then after applying a random cut and revealing the five cards, the position of the card whose bottom edge is up tells Alice about the complete status of the five cards before the random cut, and hence, she can learn the value of b. It should be noted that if Bob notices the bottom-edge-up card, then he can learn a, as well; thus, malicious Alice potentially takes a risk.

Against such an attack, we can apply the method given in Section 3.2 directly; it suffices to check whether the directions of the input commitments are the same before starting an intended protocol. Recall that the method results in either the very same sequence of four cards or the sequence where the first two and the second two cards are both swapped, and hence, any rotated card can be found.

Thus, when using a deck of cards whose backs have rotationally symmetric patterns, one should note their up/down directions during an execution of a card-based protocol. In a sense, this can be performed more easily when the non-rotationally symmetric back pattern is adopted; or it is a reasonable idea that both sides are designed to be rotationally symmetric.

4.2 Advantage

In Section 4.1 above, we mentioned that one needs to note the up/down directions of the cards during a protocol for cards with rotationally symmetric backs, such as □. However, we mention here that there is an advantage to using such rotationally symmetric backs. That is, we design a new protocol that suits a deck of cards with such a property.

For a (single) black card $\boxed{\clubsuit}$ whose back is □, consider an encoding

$$\boxed{\clubsuit} = 0, \quad \boxed{\rotatebox{180}{$\clubsuit$}} = 1,$$

and write

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

for bit x, the value of which the face-down card holds in accordance with the encoding. Then, inverting a face-down card, that is, rotating the card by 180 degrees, yields a NOT computation. Below, we construct AND and XOR protocols under the encoding.

First, consider an XOR computation. Given (up/down-direction) commitments to a and b

$$\underbrace{\boxed{}}_{a}\underbrace{\boxed{}}_{b},$$

a shuffle in which they are inverted together or remain the same can be easily implemented; for example, it suffices for Alice and Bob to rotate the two cards together in turn until they are satisfied that the cards have been adequately shuffled. After applying such a shuffle, we have

$$\underbrace{\square}_{a\oplus r}\underbrace{\square}_{b\oplus r}$$

where r is a random bit. According to the idea on which the XOR protocol [7] explained in Section 2.2 is based, turning over the left card produces a (up/down-direction) commitment to $a \oplus b$.

Next, consider an AND computation. We simulate the idea behind the AND protocol [7] explained in Section 2.1. Starting from

$$\underbrace{\square}_{a}\boxed{\clubsuit}\underbrace{\square}_{b} \quad \rightarrow \quad \underbrace{\square}_{a}\underbrace{\square}_{0}\underbrace{\square}_{b},$$

apply a shuffle where the actions of inverting the leftmost card and swapping the rightmost two cards are synchronized, then we have

$$\underbrace{\square}_{a}\underbrace{\square}_{0}\underbrace{\square}_{b} \quad \text{or} \quad \underbrace{\square}_{\bar{a}}\underbrace{\square}_{b}\underbrace{\square}_{0},$$

and hence, revealing the leftmost card gives us a commitment to $a \wedge b$. It should, however, be noted that it is not clear whether a person could easily physically implement such a shuffle.

Thus, when adopting cards having a rotationally symmetric pattern on their backs, AND and XOR computations can be achieved with a single color and half of the number of cards required for the previous protocols; however, there remains an implementation issue for the AND computation.

5 Backs with Scuff Marks

The previous work, implicitly or explicitly, assumes that all cards of the same color are indistinguishable from one another. However, in reality, such an assumption does not always hold; for example, some cards possibly have scuff marks on their backs making them distinguishable from other cards.

Now, suppose that a black card $\boxed{\clubsuit}$ has a scuff mark on its back, $\boxed{?_{1}}$, where the tiny number 1 represents the scuff mark. If an input commitment made by Alice contains that flawed card, then we have

$$\underbrace{\boxed{?_{1}}\boxed{?}}_{a} \quad \text{or} \quad \underbrace{\boxed{?}\boxed{?_{1}}}_{a},$$

and hence, a person who has noticed the scuff mark can learn $a = 0$ (in the former case) or $a = 1$ (in the latter case). Therefore, when an input commitment has a scuff mark, critical information leakage occurs.

To avoid this, adopting an idea similar to the one on which the Secret Sharing Scheme or Garbled Circuit (e.g. refer to [1, 4, 12]) is based, we make a commitment shared, as follows. For a bit x and a natural number $s \geq 2$, a sequence of s commitments

$$\underbrace{\boxed{?}\boxed{?}}_{x_1}\underbrace{\boxed{?}\boxed{?}}_{x_2}\cdots\underbrace{\boxed{?}\boxed{?}}_{x_s},$$

such that $\bigoplus_{i=1}^{s} x_i = x$ is called an *s-shared commitment* to x.

Using this new concept, we can construct novel scuff-proof XOR and AND protocols. Our protocols can maintain secrecy even if at most t cards are flawed. The details are omitted in this LNCS paper due to the page limitation.

6 Conclusion

In this paper, we considered realistic situations in card-based cryptography where some players possibly act maliciously, backs of cards are rotationally symmetric, or some cards possibly have scuff marks. We then proposed methods to maintain the secrecy of players' private inputs even under such severe conditions.

Acknowledgments

We thank the anonymous referees whose comments helped us improve the presentation of the paper. This work was supported by JSPS KAKENHI Grant No. 23700007.

References

[1] R. Cramer, I. Damgård, and J. Nielsen, Secure Multiparty Computation and Secret Sharing – An Information Theoretic Approach, book draft, May 11, 2013.

[2] C. Crépeau and J. Kilian, "Discreet solitary games," Proc. CRYPTO '93, Lecture Notes in Computer Science, vol. 773, pp. 319–330, Springer-Verlag, 1994.

[3] B. den Boer, "More efficient match-making and satisfiability: the five card trick," Proc. EUROCRYPT '89, Lecture Notes in Computer Science, vol. 434, pp. 208–217, Springer-Verlag, 1990.

[4] O. Goldreich, "Foundations of Cryptography II: Basic Applications," Cambridge University Press, Cambridge, 2004.

[5] T. Mizuki, I. K. Asiedu, and H. Sone, "Voting with a logarithmic number of cards," Proc. Unconventional Computation and Natural Computation (UCNC 2013), Lecture Notes in Computer Science, Springer-Verlag, vol. 7956, pp. 162–173, 2013.

[6] T. Mizuki and H. Shizuya, "A formalization of card-based cryptographic protocols via abstract machine," International Journal of Information Security, vol. 13, no. 1, pp. 15–23, 2014.

[7] T. Mizuki and H. Sone, "Six-card secure AND and four-card secure XOR," Proc. Frontiers in Algorithmics (FAW 2009), Lecture Notes in Computer Science, vol. 5598, pp. 358–369, Springer-Verlag, 2009.

[8] T. Mizuki, M. Kumamoto, and H. Sone, "The five-card trick can be done with four cards," Proc. ASIACRYPT 2012, Lecture Notes in Computer Science, Springer-Verlag, vol. 7658, pp. 598–606, 2012.

[9] T. Mizuki, F. Uchiike, and H. Sone, "Securely computing XOR with 10 cards," Australasian Journal of Combinatorics, vol. 36, pp. 279–293, 2006.

[10] V. Niemi and A. Renvall, "Secure multiparty computations without computers," Theoretical Computer Science, vol. 191, pp. 173–183, 1998.

[11] T. Nishida, T. Mizuki, and H. Sone, "Securely computing the three-input majority function with eight cards," Proc. Theory and Practice of Natural Computing (TPNC 2013), Lecture Notes in Computer Science, Springer-Verlag, vol. 8273, pp. 193-204, 2013.

[12] T. Schneider, "Engineering Secure Two-Party Computation Protocols," Springer-Verlag, Berlin Heidelberg, 2012.

[13] A. Stiglic, "Computations with a deck of cards," Theoretical Computer Science, vol. 259, pp. 671–678, 2001.