# The Six-Card Trick: Secure Computation of Three-Input Equality

Kazumasa Shinagawa[1,2] and Takaaki Mizuki[3]

[1] Tokyo Institute of Technology, Meguro, Japan
shinagawakazumasa@gmail.com
[2] Institute of Advanced Industrial Science and Technology (AIST), Kōtō, Japan
[3] Tohoku University, Sendai, Japan

**Abstract.** Secure computation enables parties having secret inputs to compute some function of their inputs without revealing inputs beyond the output. It is known that secure computation can be done by using a deck of physical cards. The *five-card trick* proposed by den Boer in 1989 is the first card-based protocol, which computes the logical AND function of two inputs. In this paper, we design a new protocol for the three-input equality function using six cards, which we call the *six-card trick*.

## 1 Introduction

Suppose that during the two-candidate election, Alice, Bob, and Charlie wish to talk about the candidates only if they are supporting the same candidate. Unfortunately, they do not know each other's supporting candidate. If their supporting candidates do not coincide, they wish to hide their supporting candidates from each other in order not to break their friendship. Due to this secrecy condition, just revealing supporting candidates to others does not work. How can we solve this problem?

*Secure computation*, which is one of cryptographic techniques, serves a solution to such a situation in which parties wish to compare secret information without leaking it. Specifically, it enables a set of parties to compute some function of their inputs without leaking the inputs beyond the output. In the above situation, for the inputs $a, b, c \in \{0, 1\}$ indicating the candidates that Alice, Bob, and Charlie support, respectively, it is sufficient to securely compute a three-input function $f$ such that $f(a, b, c) = 1$ if $a = b = c$ and 0 otherwise. When the output of the function is 1, they can start to talk about the election since in this case their supporting candidates are the same. When the output of the function is 0, they should talk about a topic which is not related to the election campaigns. In the latter case, thanks to the security of secure computation, their supporting candidates are hidden from each other. For example, Alice cannot obtain Bob's and Charlie's supporting candidates. The only thing obtained by Alice is that *at least one of Bob and Charlie supports the opposite candidate from hers*, but this is not problematic since it can be computable from the output of the function.

Secure computation is one of very active research areas in cryptography. In this paper, we focus on secure computation using a deck of cards, which is often called *card-based cryptography*, while most of secure computation is designed to be executed on computers. The reason why we choose card-based cryptography instead of secure computation running on computers, is that it is suitable to a situation in which *all parties gather together in the same place*. This is in contrast to secure computation running on computers which is designed to be executed among *parties whose locations are physically separated*. Card-based cryptography often provides a simple solution to such an everyday situation since the participants can watch each other to prevent malicious behavior. For example, in card-based cryptography, secret information is encoded by a sequence of face-down cards on a dinning table, instead of encryption as in the standard secure computation. This is sufficient to enforce honest behavior on parties since any malicious behavior can be detected by the parties. Moreover, due to its simplicity of model of card-based cryptography, it is easy to understand correctness and security even for non-experts, compared to secure computation running on computers. Therefore, this provides a solution such that all parties are convinced how their secret inputs are protected.

The first research on card-based cryptography is the *five-card trick* proposed by den Boer [2]. It provides a very simple solution to the case that two parties Alice and Bob having $a, b \in \{0, 1\}$, respectively, wish to securely compute the logical AND $a \wedge b$ (see Section 3). It has two nice properties. The first one is that it requires only a single shuffle, especially, a *random cut*. Since a random cut is accepted[1] as the most basic shuffle, this property means that it is the most efficient in terms of shuffle operation. We call a protocol that only requires a random cut a *single-cut protocol*. The second nice property is that at the end of an execution, all cards are face-up. We call such a protocol a *gabage-free protocol*.

Despite of these nice properties, as far as we know, no single-cut garbage-free protocols have been proposed except for the five-card trick. Are these properties special for the logical AND computation? In this paper, we show that the answer is NO by constructing a single-cut garbage-free protocol for the three-input equality function using six cards, which we call the *six-card trick*.

## 1.1 Related Works

Card-based protocol begins with the five-card trick proposed by den Boer [2] in 1989. Crépeau and Killian [1] showed that every function can be securely computed by applying a number of random cuts. A number of subsequent works [6, 9, 10, 13, 14, 16] improved upon the protocols [1] in terms of numbers of cards and shuffles. While these works [1, 6, 9, 13, 14, 16] aimed to achieve the feasibility and the efficiency of general secure computation, there is another line of research which focuses on a specific problem [2–4, 7, 8, 11, 12, 15]. Our work focuses on

[1] A random cut is securely implemented by a Hindu cut [17], while most of other shuffles do not have a (direct) secure implementation. Koch and Walzer [5] showed that every uniform and closed shuffles are reduced to a number of random cuts.

a specific problem, specifically, the three-input equality function for Boolean inputs.

There are two types of card-based protocols: “committed format” protocols (e.g. [1, 9]) and “non-committed format” protocols (e.g. [2, 8]). The former produces a sequence of face-down cards that follows the input encoding. Thus, an output sequence of a committed format protocol can be used to be inputted to another protocol. In contrast, the latter outputs the result value directly. A committed format protocol implies a non-committed format protocol for the same function, by just turning over the output sequence. On the other hand, in general, non-committed format protocols can be more efficient in terms of the number of cards and shuffles, compared to committed format protocols. Our protocol, the six-card trick, is a non-committed format protocol for the three-input equality function.

### 1.2 Organization

In Section 2, we present basic definitions of card-based cryptography. In Section 3, we review a previous work called the five-card trick. In Section 4, we design our protocol for the three-input equality function, which we call the six-card protocol. In Section 6, we conclude this paper.

## 2 Card-based Cryptography

In this section, we present basic definitions of card-based cryptography. Although the first three subections provide very common definitions of card-based cryptography, the last subsection defines new properties, which we call *single-cut* and *garbage-free.*

### 2.1 Basic Setting of Card-based Protocols

Suppose that Alice, Bob, and Charlie having secret inputs $a, b, c \in \{0, 1\}$, respectively, wish to compute some function on their inputs without revealing the inputs beyond the output. (In the two-party case as in Section 3, Charlie is ignored.) They are in the same room together. There are a deck of cards and a flat space (e.g. dinner table) in the room. Cards will be arranged on the flat space. The deck of cards in our use contains two types of cards, $\boxed{\clubsuit}$ and $\boxed{\heartsuit}$, whose back sides are the same pattern $\boxed{?}$. All cards having the same type ($\boxed{\clubsuit}$ or $\boxed{\heartsuit}$) are indistinguishable. We say that $\boxed{\clubsuit}$ and $\boxed{\heartsuit}$ lying on the flat space are *face-up cards* and $\boxed{?}$ is a *face-down card*.

### 2.2 Commitment

We use the following encoding rule for Boolean inputs:

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

For a bit $x \in \{0,1\}$, two face-down cards $\boxed{?}\boxed{?}$ having a value $x$ according to the above encoding is called a *commitment* to $x$, denoted by

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

## 2.3 Random Cut

A *random cut* is a random cyclic shift operation. Let $X$ be a sequence of face-down cards $(x_0, x_1, \cdots, x_{n-1})$ as follows:

$$X = \underset{x_0}{\boxed{?}}\underset{x_1}{\boxed{?}} \cdots \underset{x_{n-1}}{\boxed{?}}.$$

For a sequence of cards $X$, a random cut generates a randomly shifted sequence $(x_i, x_{i+1}, \cdots, x_{i+n-1})$ for a uniformly distributed random integer $i$, $0 \le i \le n-1$, denoted by $\langle \cdot \rangle$, as follows:

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

It must satisfy that the random number $i$ is hidden from all parties. Ueda et al. [17] experimentally showed that a random cut is securely implementable by applying *Hindu cut*, which is a kind of shuffling operations widely used in card games.

## 2.4 Single-cut and Garbage-free Protocols

We say that a card-based protocol is *single-cut* if it requires only one random cut and does not require other shuffles. We say that a card-based protocol is *garbage-free* if at the end of the protocol, all cards are face-up.

In this paper, we focus on single-cut and garbage-free protocols of the following type:

1. Each party having two cards $\boxed{\clubsuit}\boxed{\heartsuit}$ privately places them in the face-down format according to his/her input. These pairs of cards (possibly with additional cards) are arranged according to a predetermined permutation.
2. Apply a random cut to the sequence of cards.
3. Turn over all of the cards. The output can be obtained from the pattern of the resulting sequence.

For ease of explanation, we define a term *cyclic set*. For a sequence of face-up cards, we define its cyclic set as a set of all its cyclic shifted sequences. For example, the cyclic set of $\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}$ is the set of three sequences:

$$\left\{ \boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit}, \boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}, \boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit} \right\}.$$

We say that a *single-cut garbage-free protocol computes a function* $f : \{0,1\}^n \to \{0,1\}$ if there exists two cyclic sets $C_0$ and $C_1$ such that for every input $x \in \{0,1\}^n$, a sequence of cards after the final step of the protocol is always contained in $C_{f(x)}$. For such a protocol, a proof of security is trivial since all cards are face-down except at the end of the protocol.

## 3 Five-card Trick

Before presenting our construction of the six-card trick, we review the seminal work called the *five-card trick* proposed by den Boer [2]. This is a secure two-party protocol computing the logical AND function $a \wedge b$ for inputs $a, b \in \{0, 1\}$. As the name suggests, it only requires five cards: $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$.

Now we are ready to explain the five-card trick. Suppose that Alice and Bob having $a, b \in \{0, 1\}$, respectively, wish to securely compute the logical AND function. The five-card trick [2] proceeds 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 $\boxed{\heartsuit}$ are arranged as follows:

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

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

2. A random cut is applied to the five cards as follows:

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

3. Turn over all cards; then, we can consider cyclic sets of two sequences:

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

   The left cases, three (cyclically) consecutive $\boxed{\heartsuit}$'s, imply $a \wedge b = 1$ and the right imply $a \wedge b = 0$.

The correctness of the five-card trick can be observed by Table 1 showing all possible sequences after Step 1. As described in Section 2.4, the security of the five-card trick is trivial because all cards are face-down except at the end of protocol.

| $(a, b)$ | sequence |
|---|---|
| $(0, 0)$ | $\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}$ |
| $(0, 1)$ | $\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit}$ |
| $(1, 0)$ | $\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}$ |
| $(1, 1)$ | $\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit}$ |

**Table 1.** All possibilities of the sequence after Step 1.

## 4 Six-Card Trick

In this section, we present the six-card trick.

The six-card trick is a protocol that securely computes the three-input equality function that takes three bits as inputs and outputs 1 if they are the same and 0 otherwise. As the name suggests, it only requires six cards: $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}$. Suppose that Alice, Bob, and Charlie have $a, b, c \in \{0,1\}$, respectively, and wish to securely compute the three-input equality function. The protocol proceeds as follows:

1. Alice, Bob, and Charlie privately arrange commitments to the inputs $a$, $b$, and $c$, respectively:
$$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{b}\underbrace{\boxed{?}\boxed{?}}_{c}.$$
2. The six cards are arranged as follows:
$$\begin{tikzcd}
\boxed{?} & \boxed{?} \arrow[drr] & \boxed{?} & \boxed{?} \arrow[drr] & \boxed{?} & \boxed{?} \arrow[dllll] \\
\boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?}.
\end{tikzcd}$$
3. A random cut is applied to the six cards as follows:
$$\left\langle \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?} \right\rangle \rightarrow \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}.$$
4. Turn over all cards; then, we can consider cyclic sets of two sequences:
$$\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\clubsuit} \text{ or } \boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\clubsuit}.$$

   The output is 0 if it is the former case and 1 otherwise.

The correctness of the six-card trick can be easily observed by Tables 2 and 3 which show all possibilities of the sequence after Steps 1 and 2, respectively. As described in Section 2.4, the security of the six-card trick is trivial because all cards are face-down except at the end of protocol.

## 5 Open Problems

In Section 4, we obtained a six-card protocol for the three-input equality function. A natural question arises: Can we construct a $2n$-card protocol for the $n$-input equality function? Indeed, we obtain a four-card protocol for the two-input equality function. For inputs $a, b \in \{0,1\}$, the protocol just applies a random cut to the following sequence:

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

| $(a,b,c)$ | sequence |
|---|---|
| $(0,0,0)$ | ♣♡♣♡♣♡ |
| $(0,0,1)$ | ♣♡♣♡♡♣ |
| $(0,1,0)$ | ♣♡♡♣♣♡ |
| $(0,1,1)$ | ♣♡♡♣♡♣ |
| $(1,0,0)$ | ♡♣♣♡♣♡ |
| $(1,0,1)$ | ♡♣♣♡♡♣ |
| $(1,1,0)$ | ♡♣♡♣♣♡ |
| $(1,1,1)$ | ♡♣♡♣♡♣ |

**Table 2.** All possibilities of the sequence after Step 1.

and outputs 1 if the resulting face-up sequence is ♣♡♣♡, and 0 otherwise[2]. Thus, the answer of the above question is YES for $n \in \{2,3\}$.

Unfortunately, by using a computer, we found that there is no eight-card protocol for the four-input equality function. Thus, the answer of the above question is NO in general. Our conjecture is that the answer of the above question is NO for all $n \geq 4$. We left it as an open question.

Other open problems are as follows:

- Construct a single-cut and garbage-free protocol for the $n$-input equality function using more than $2n$ cards.
- Find single-cut and garbage-free protocols for other interesting functions.

## 6 Conclusion

In this paper, we designed the six-card trick, which is a single-cut garbage-free protocol for the three-input equality function. We leave several questions as open problems.

## Acknowledgments

This work was supported in part by JSPS KAKENHI Grant Numbers 17J01169 and 17K00001. The authors would like to thank Osamu Watanabe for his valuable comments.

[2] This technique was implicitly used in [1].

| $(a,b,c)$ | sequence |
|---|---|
| $(0,0,0)$ | ♣♡♣♡♣♡ |
| $(0,0,1)$ | ♣♣♣♡♡♡ |
| $(0,1,0)$ | ♣♡♡♡♣♣ |
| $(0,1,1)$ | ♣♣♡♡♡♣ |
| $(1,0,0)$ | ♡♡♣♣♣♡ |
| $(1,0,1)$ | ♡♣♣♣♡♡ |
| $(1,1,0)$ | ♡♡♡♣♣♣ |
| $(1,1,1)$ | ♡♣♡♣♡♣ |

**Table 3.** All possibilities of the sequence after Step 2.

## References

1. C. Crépeau and J. Kilian. Discreet solitary games. In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 319–330, 1993.
2. B. den Boer. More efficient match-making and satisfiability: *The Five Card Trick*. In *Advances in Cryptology - EUROCRYPT '89, Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, April 10-13, 1989, Proceedings*, pages 208–217, 1989.
3. Y. Hashimoto, K. Shinagawa, K. Nuida, M. Inamura, and G. Hanaoka. Secure grouping protocol using a deck of cards. In *Information Theoretic Security - 10th International Conference, ICITS 2017, Hong Kong, China, November 29 - December 2, 2017, Proceedings*, pages 135–152, 2017.
4. R. Ishikawa, E. Chida, and T. Mizuki. Efficient card-based protocols for generating a hidden random permutation without fixed points. In *Unconventional Computation and Natural Computation - 14th International Conference, UCNC 2015, Auckland, New Zealand, August 30 - September 3, 2015, Proceedings*, pages 215–226, 2015.
5. A. Koch and S. Walzer. Foundations for actively secure card-based cryptography. *IACR Cryptology ePrint Archive*, 2017:423, 2017.
6. A. Koch, S. Walzer, and K. Härtel. Card-based cryptographic protocols using a minimal number of cards. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I*, pages 783–807, 2015.
7. T. Mizuki, I. K. Asiedu, and H. Sone. Voting with a logarithmic number of cards. In *Unconventional Computation and Natural Computation - 12th International Conference, UCNC 2013, Milan, Italy, July 1-5, 2013. Proceedings*, pages 162–173, 2013.
8. T. Mizuki, M. Kumamoto, and H. Sone. The five-card trick can be done with four cards. In *Advances in Cryptology - ASIACRYPT 2012 - 18th International*

*Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, pages 598–606, 2012.

9. T. Mizuki and H. Sone. Six-card secure AND and four-card secure XOR. In *Frontiers in Algorithmics, Third International Workshop, FAW 2009, Hefei, China, June 20-23, 2009. Proceedings*, pages 358–369, 2009.
10. T. Mizuki, F. Uchiike, and H. Sone. Securely computing XOR with 10 cards. *The Australasian Journal of Combinatorics*, 36:279–293, 2006.
11. T. Nakai, S. Shirouchi, M. Iwamoto, and K. Ohta. Four cards are sufficient for a card-based three-input voting protocol utilizing private permutations. In *Information Theoretic Security - 10th International Conference, ICITS 2017, Hong Kong, China, November 29 - December 2, 2017, Proceedings*, pages 153–165, 2017.
12. T. Nakai, Y. Tokushige, Y. Misawa, M. Iwamoto, and K. Ohta. Efficient card-based cryptographic protocols for millionaires' problem utilizing private permutations. In *Cryptology and Network Security - 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings*, pages 500–517, 2016.
13. V. Niemi and A. Renvall. Secure multiparty computations without computers. *Theor. Comput. Sci.*, 191(1-2):173–183, 1998.
14. T. Nishida, Y. Hayashi, T. Mizuki, and H. Sone. Card-based protocols for any boolean function. In *Theory and Applications of Models of Computation - 12th Annual Conference, TAMC 2015, Singapore, May 18-20, 2015, Proceedings*, pages 110–121, 2015.
15. T. Nishida, T. Mizuki, and H. Sone. Securely computing the three-input majority function with eight cards. In *Theory and Practice of Natural Computing - Second International Conference, TPNC 2013, Cáceres, Spain, December 3-5, 2013, Proceedings*, pages 193–204, 2013.
16. A. Stiglic. Computations with a deck of cards. *Theor. Comput. Sci.*, 259(1-2):671–678, 2001.
17. I. Ueda, A. Nishimura, Y. Hayashi, T. Mizuki, and H. Sone. How to implement a random bisection cut. In *Theory and Practice of Natural Computing - 5th International Conference, TPNC 2016, Sendai, Japan, December 12-13, 2016, Proceedings*, pages 58–69, 2016.