The Five-Card Trick Can Be Done with Four Cards

Takaaki Mizuki, Michihito Kumamoto, and Hideaki Sone

Cyberscience Center, Tohoku University,
Aramaki-Aza-Aoba 6-3, Aoba-ku, Sendai 980-8578, Japan
tm-paper+ac2012@g-mail.tohoku-university.jp

Abstract. The "five-card trick" invented by Boer allows Alice and Bob to securely compute the AND function of their secret inputs using five cards—three black cards and two red cards—with identical backs. This paper shows that such a secure computation can be done with only four cards. Specifically, we give a protocol to achieve a secure computation of AND using only four cards—two black and two red. Our protocol is optimal in the sense that the number of required cards is minimum.

1 Introduction

Assume that two *honest-but-curious* players Alice and Bob, who hold secret bits $a \in \{0,1\}$ and $b \in \{0,1\}$, respectively, wish to *securely compute* the AND function, that is, they want to learn the value of $a \wedge b$ without revealing more of their own secret bits than necessary. The "five-card trick" invented in 1989 by Boer [2] achieves such a secure computation of AND using five cards $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}$. Now, after over two decades since the invention of the five-card trick, this paper improves upon the result: we show that the same secure computation can be done using only four cards $\boxed{\clubsuit}\boxed{\clubsuit}\boxed{\heartsuit}\boxed{\heartsuit}$.

This paper begins with an overview of the five-card trick.

1.1 The five-card trick

The "five-card trick" by Boer [2] is an elegant secure AND computation protocol that uses three $\boxed{\clubsuit}$s and two $\boxed{\heartsuit}$s. Before going into the details of the protocol, we first mention the properties of cards appearing in this paper.

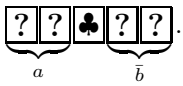
All cards of the same type ($\boxed{\clubsuit}$ or $\boxed{\heartsuit}$) are assumed to be indistinguishable from one another. We use $\boxed{?}$ to denote a card lying face down. We also assume that the back $\boxed{?}$ of each card is identical. To deal with Boolean values, we use the following encoding:

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

Given a bit $x \in \{0,1\}$, a pair of face-down cards $\boxed{?}\boxed{?}$ whose value is equal to x (according to the encoding rule (1) above) is called a *commitment to* x, and

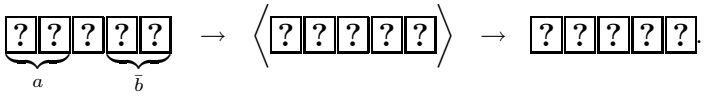
is expressed as

We now explain how to play the five cards in Boer's secure AND protocol. First, given two cards ♣♡ out of the five cards, Alice privately makes a commitment to her secret bit a (without Bob's knowing the order of the two cards); similarly, Bob makes a commitment to the negation $\bar{b}$ of his secret bit b. Then, with the remaining one card ♣, two commitments are put forth as follows:

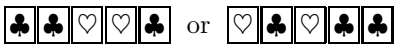


It should be noted that the three cards in the middle would be ♣♣♣ only when $a = b = 1$ (if the second and fourth cards from the left were turned over).

Next, Alice and Bob turn the centered card ♣ face down, and apply a *random cut*, which is denoted by $\langle\cdot\rangle$:



A random cut (also called a *random cyclic shuffling*) means that, as in the case of usual card games, a random number of leftmost cards are moved to the right without changing their order (of course, the random number must be unknown to Alice and Bob); to implement this, it suffices that Alice and Bob take turns cutting the deck until they are satisfied. Finally, Alice and Bob reveal all five cards. Then, the resulting sequence is either

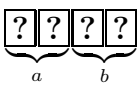
 (2)

apart from cyclic rotations, where either the three ♣s are "cyclically" consecutive or not. One can easily verify that the former case implies $a \wedge b = 1$, and the latter case implies $a \wedge b = 0$.

This is the *five-card trick*, a simple and elegant secure AND protocol.

1.2 Our result and related work

In this paper, we reduce the number of required cards by one, compared to the five-card trick, as listed in Table 1. That is, given commitments



to Alice's bit a and Bob's bit b, our protocol needs no card other than the four cards constituting the two commitments, i.e., it can securely evaluate the value of $a \wedge b$ without the use of any additional card. Therefore, as long as one adopts

∘ *Secure AND in a non-committed format*

	# of card types	# of cards
Boer [2] (§1.1)	2	5
Ours (§2)	2	4

Table 1. The five-card trick and our protocol with their performance.

the encoding rule (1), our protocol is optimal in the sense that the number of required cards is minimum because at least four cards are necessary for the two inputs a and b.

Since the invention of the five-card trick, there have been several *card-based protocols* for secure computation, as listed in Table 2. All these protocols produce their output (say, $a \wedge b$) in a *committed format*, i.e., their output is described as a sequence like

that follows the encoding rule (1) (and Alice and Bob have no knowledge about the value than their own secret bits). In contrast, the five-card trick and our protocol (given in Section 2) output the value of $a \wedge b$ in a *non-committed format*; the format of the output $a \wedge b$ differs from the format of inputs a and b, namely the encoding rule (1) (recall the resulting sequences (2), which are completely revealed to the public at the end of the protocol).

∘ *Secure AND in a committed format*

	# of card types	# of cards	avg. # of trials
Crépeau-Kilian [3]	4	10	6
Niemi-Renvall [7]	2	12	2.5
Stiglic [10]	2	8	2
Mizuki-Sone [4]	2	6	1

∘ *Secure XOR in a committed format*

	# of card types	# of cards	avg. # of trials
Crépeau-Kilian [3]	4	14	6
Mizuki-Uchiike-Sone [5]	2	10	2
Mizuki-Sone [4]	2	4	1

Table 2. The "committed format" protocols.

Thus, all the card-based protocols are categorized into two types: "non-committed format" protocols (Table 1) and "committed format" protocols (Table 2); this paper addresses the former. Note that in Table 2 every protocol whose average number of trials is more than 1 is a Las Vegas algorithm.

While card-based protocols might fall within the area of *cryptography without computers* [7], *recreational cryptography* [1] or *human-centric cryptography* [6], we

believe that this type of research will help professional cryptographers intuitively explain to nonspecialists the nature of their constructed cryptographic protocols (e.g. [9]). That is, card-based protocols would help ordinary people understand what secure computations are, or, more fundamentally, what cryptography is. Furthermore, it should be noted that some of the card-based protocols are implemented and used in online games [8].

The remainder of this paper is organized as follows. In Section 2, we give a description of our four-card secure AND protocol. In Section 3, we show the correctness of our protocol, that is, we prove that our protocol securely computes the AND function. This paper concludes in Section 4 with an open question.

2 Description of Our Protocol

In this section, we design a new card-based protocol that securely computes the AND function using only four cards $\clubsuit\clubsuit\heartsuit\heartsuit$.

In Section 2.1, we first introduce the "random bisection cut" [4] used in our protocol. We then describe our protocol in Section 2.2.

2.1 Random bisection cuts

As seen in Section 1.1, applying a random cut to a sequence of face-down cards results in a sequence such that a random number of leftmost cards are moved to the right without changing their order. Whereas, a "random bisection cut" [4] works differently, as follows.

Given a deck of (an even number of) face-down cards, say

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

bisect it and randomly switch the resulting two decks; such a card shuffling operation is called a *random bisection cut*. For the example above, a random bisection cut, denoted by $[\cdot\,\|\,\cdot]$, works as

$$\underbrace{?\,?}_{a}\,\underbrace{?\,?}_{b} \;\rightarrow\; \left[?\,?\,\middle\|\,?\,?\right] \;\rightarrow\; ?\,?\,?\,?,$$

where the resulting deck of the four cards is either

$$\underbrace{?\,?}_{a}\,\underbrace{?\,?}_{b} \text{ or } \underbrace{?\,?}_{b}\,\underbrace{?\,?}_{a},$$

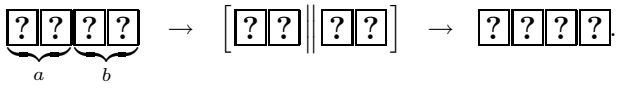
and each case occurs with probability of exactly 1/2.

Although at first glance a random bisection cut seems to be a little bit less natural operation compared to a (normal) random cut, we hope that people will feel a random bisection cut to be an easy-to-implement operation some day. If Alice and Bob are not familiar with playing cards, then they may hold each of the two bisected decks together using a clip before shuffling the two decks. Alternatively, they may put each of the two decks into an envelope (without changing the order of the cards), and shuffle the two envelopes.

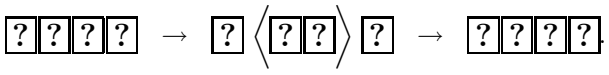
2.2 The protocol

Given a commitment to Alice's bit a and a commitment to Bob's bit b, our four-card AND protocol proceeds as follows.

1. Apply a random bisection cut:

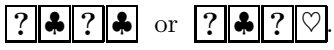


2. Apply a random cut to the two cards in the middle, namely the second and third cards:



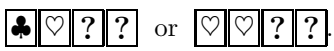
3. Reveal the second card.

   (a) If the face-up second card is ♣, then open the fourth card. We now have either



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

   (b) If the face-up second card is ♡, then open the first card. We now have either



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

As described above, our protocol makes one random bisection cut (in step 1) and one random cut (in step 2). After those cuts, two cards are eventually revealed, namely either (a) the second and fourth cards, or (b) the first and second cards, depending on the result of revealing the second card in step 3. Note that if the two face-up cards are the same type (♣♣ or ♡♡), then we have $a \wedge b = 1$; otherwise, we have $a \wedge b = 0$.

We show why our protocol works in the next section.

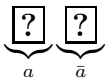
3 Correctness of Our Protocol

In this section, we prove that the protocol given in the previous section securely computes $a \wedge b$. First, in Section 3.1 we intuitively explain why our protocol works. Then, we verify the correctness of our protocol in Section 3.2.

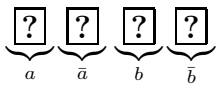
3.1 An intuitive sketch

Given a commitment

note that each individual card constituting the commitment inherently has the value of the bit a, that is, one can also write

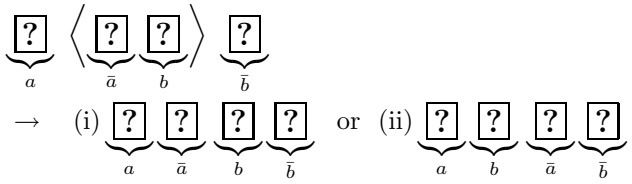


where an encoding rule for a single card is taken: $\boxed{\clubsuit}$ expresses 0, and $\boxed{\heartsuit}$ expresses 1. Based on such a single-card encoding, commitments to a and b can be expressed as:

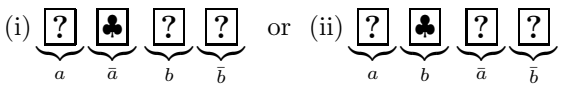


(3)

Now, for the expression (3), skip step 1 in our protocol and directly apply step 2. That is, apply a random cut to the second and third cards:



Next, applying step 3, reveal the second card. Assume that the face-up second card is $\boxed{\clubsuit}$ (as in step 3(a)), i.e.,



(4)

Then, it means that either (i) $\bar{a} = 0$ or (ii) $b = 0$ (because $\boxed{\clubsuit} = 0$). If (i) $\bar{a} = 0$, then $a = 1$ and hence $a \wedge b = b$; if (ii) $b = 0$, then $a \wedge b = 0 = b$. Therefore, in either case, we have $a \wedge b = b$, and hence one can notice that the value of $a \wedge b = b$ can be obtained by revealing the fourth card

in the sequence (4). Actually, in step 3(a), the fourth card is opened. If it is $\boxed{\clubsuit}$, then $\bar{b} = \boxed{\clubsuit} = 0$ and hence $a \wedge b = b = 1$; if it is $\boxed{\heartsuit}$, then $\bar{b} = \boxed{\heartsuit} = 1$ and hence $a \wedge b = b = 0$.

Thus, steps 2 and 3(a) surely compute the value of $a \wedge b$. One can similarly verify the claim for the case of step 3(b). Therefore, steps 2 and 3 can provide at

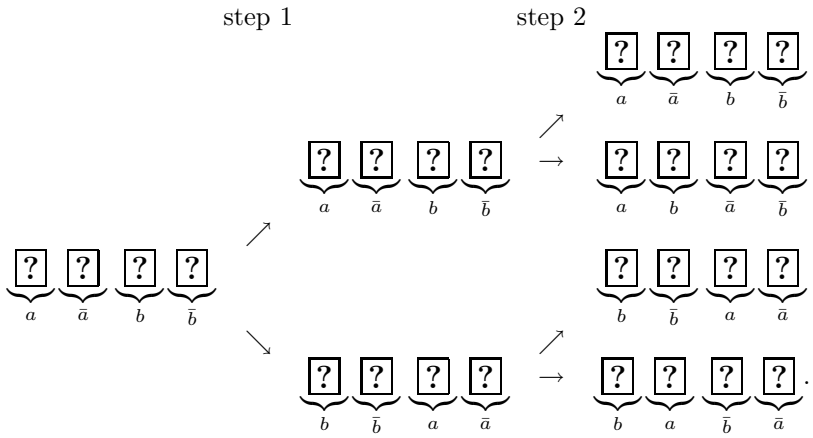
least the value of $a \wedge b$. However, they also leak some secret information about a and b; indeed, for example, when the second card revealed in step 3 was $\clubsuit$, we have $\bar{a} = 0$ or $b = 0$ (as seen above), and hence the fact of $(a, b) \neq (0, 1)$ has been disclosed. Therefore, since executing only steps 2 and 3 is not secure, our protocol applies a random bisection cut in step 1 to guarantee secrecy, as intuitively explained below.

Note first that adding step 1 never affects the computation outcome of executing steps 2 and 3: after applying a random bisection cut in step 1, we have either

$$\underbrace{?\,?}_{a}\,\underbrace{?\,?}_{b} \quad \text{or} \quad \underbrace{?\,?}_{b}\,\underbrace{?\,?}_{a}, \tag{5}$$

and then applying steps 2 and 3 to the sequence (5) always provides the value of $a \wedge b$ as shown above (because $a \wedge b = b \wedge a$).

To see that the secrecy is preserved by introducing a random bisection cut in step 1, we enumerate all possibilities:



Therefore, opening the second card (in the rightmost sequence) means that one of $\bar{a}$, b, $\bar{b}$ and a is randomly revealed. Hence, opening the second card never leaks any information about each of a and b.

3.2 Proof of correctness

In this subsection, we prove that our protocol works correctly.

Recall that a random bisection cut $[\,?\,?\,\|\,?\,?\,]$ in step 1 and a random cut $?\,\langle\,?\,?\,\rangle\,?$ in step 2 are applied to the two commitments; one can enumerate, as in Table 3, all possibilities of the four cards after each of steps 1 and 2. Note that the cases $(a, b) = (0, 1)$ and $(a, b) = (1, 0)$ both fall in the same status category after step 1 (and after step 2, of course).

Consider the actual execution of our protocol based on Table 3. After step 3, all possibilities can be enumerated as shown in Table 4. (Remember that the second card is opened in step 3, and that if it is $\clubsuit$, then the fourth card is

(a,b)	initial	after step 1	after step 2
$(0,0)$	♣♡♣♡	♣♡♣♡	♣♣♡♡ or ♣♡♣♡
$(0,1)$	♣♡♡♣	♣♡♡♣ or ♡♣♣♡	♡♣♣♡ or ♣♡♡♣
$(1,0)$	♡♣♣♡	same as $(0,1)$	same as $(0,1)$
$(1,1)$	♡♣♡♣	♡♣♡♣	♡♣♡♣ or ♡♡♣♣

Table 3. All possibilities of the four cards after each of steps 1 and 2.

opened; otherwise, the first one is opened.) Table 4 immediately implies that our protocol surely computes the value of $a \wedge b$—if the two revealed cards are the same type, then $a \wedge b = 1$; otherwise, $a \wedge b = 0$.

(a,b)	initial	after step 3
$(0,0)$	♣♡♣♡	?♣?♡ or ♣♡??
$(0,1)$	♣♡♡♣	?♣?♡ or ♣♡??
$(1,0)$	♡♣♣♡	?♣?♡ or ♣♡??
$(1,1)$	♡♣♡♣	?♣?♣ or ♡♡??

Table 4. All possibilities of the four cards after step 3.

To verify the secrecy of the protocol, it suffices to show that

$$\Pr[(0,0) \,|\, \boxed{?}\boxed{\clubsuit}\boxed{?}\boxed{\heartsuit}] = \Pr[(0,0) \,|\, \boxed{\clubsuit}\boxed{\heartsuit}\boxed{?}\boxed{?}] = \Pr[(0,0) \,|\, a \wedge b = 0],$$

$$\Pr[(0,1) \,|\, \boxed{?}\boxed{\clubsuit}\boxed{?}\boxed{\heartsuit}] = \Pr[(0,1) \,|\, \boxed{\clubsuit}\boxed{\heartsuit}\boxed{?}\boxed{?}] = \Pr[(0,1) \,|\, a \wedge b = 0],$$

and

$$\Pr[(1,0) \,|\, \boxed{?}\boxed{\clubsuit}\boxed{?}\boxed{\heartsuit}] = \Pr[(1,0) \,|\, \boxed{\clubsuit}\boxed{\heartsuit}\boxed{?}\boxed{?}] = \Pr[(1,0) \,|\, a \wedge b = 0].$$

Let $\Pr[a = 0] = p$ and $\Pr[b = 0] = q$. Then, we have $\Pr[(0,0) \,|\, a \wedge b = 0] = pq/(p + q - pq)$. On the other hand,

$$\Pr[(0,0) \,|\, \boxed{?}\boxed{\clubsuit}\boxed{?}\boxed{\heartsuit}] = \frac{pq(1/2)}{pq(1/2) + p(1-q)(1/2) + (1-p)q(1/2)},$$

as desired. For all the remaining cases, one can easily check the equality.

4 Conclusions

In this paper, we presented a four-card secure AND protocol whose output is in a non-committed format. Since the existing protocol, namely the five-card trick, requires five cards, we have succeeded in reducing the number of required cards by one. Our protocol is optimal in the sense that at least four cards are required for commitments to the inputs a and b.

Note that the OR function can also be securely computed by using four cards, say, according to de Morgan's law $a \vee b = \overline{\bar{a} \wedge \bar{b}}$.

This paper showed a secure AND computation in a non-committed format using only four cards. For the committed format case, the best known protocol [4] requires six cards as seen in Table 2. An intriguing open question is whether there exists a "committed format" AND protocol that requires fewer than six cards.

Acknowledgments

We thank the anonymous referees whose comments helped us to improve the presentation of the paper. This work was supported by JSPS KAKENHI Grant Number 23700007.

References

1. J. Balogh, J. A. Csirik, Y. Ishai, and E. Kushilevitz, "Private computation using a PEZ dispenser," Theoretical Computer Science, vol. 306, pp. 69–84, 2003.
2. 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.
3. C. Crépeau and J. Kilian, "Discreet solitary games," Proc. CRYPTO '93, Lecture Notes in Computer Science, vol. 773, pp. 319–330, Springer-Verlag, 1994.
4. 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.
5. T. Mizuki, F. Uchiike, and H. Sone, "Securely computing XOR with 10 cards," Australasian Journal of Combinatorics, vol. 36, pp.279-293, 2006.
6. T. Moran and M. Naor, "Polling with physical envelopes: a rigorous analysis of a human-centric protocol," Proc. EUROCRYPT 2006, Lecture Notes in Computer Science, vol. 4004, pp. 88–108, Springer-Verlag, 2006.
7. V. Niemi and A. Renvall, "Secure multiparty computations without computers," Theoretical Computer Science, vol. 191, pp. 173–183, 1998.
8. H. Stamer, "Efficient electronic gambling: an extended implementation of the toolbox for mental card games," Proc. Western European Workshop on Research in Cryptology (WEWoRC 2005), Lecture Notes in Informatics, vol. P-74, pp. 1–12, 2005.
9. S. Stamm and M. Jakobsson, "Privacy-preserving polling using playing cards," IACR Eprint archive, 2005.
10. A. Stiglic, "Computations with a deck of cards," Theoretical Computer Science, vol. 259, pp. 671–678, 2001.