2018 13th Asia Joint Conference on Information Security

Efficient Card-based Cryptographic Protocols for the Millionaires' Problem Using Private Input Operations

Hibiki Ono Yoshifumi Manabe
Faculty of Informatics,
Kogakuin University
Shinjuku, Tokyo 163-8677 Japan
manabe@cc.kogakuin.ac.jp

***Abstract*—This paper proposes new efficient card-based cryptographic protocols for the millionaires' problem using private input operations. The millionaires' problem is one of the fundamental problems in cryptography. Two players, Alice and Bob, want to know which of them is richer without revealing their actual amount of asset. Many cryptographic protocols were proposed to solve the problem. Card-based cryptographic protocols were proposed to execute cryptographic protocols using physical cards instead of computers. Though some card-based cryptographic protocols for the millionaires' problem were proposed, most of them use many cards whose number depends on the size of the amount of asset. Though Nakai et al. implicitly proposed a new protocol that uses a constant number of cards using private input operations, their protocol is not efficient since the number of rounds is $2n+1$, where n is the maximum number of bits of the asset. This paper shows new card-based protocols whose number of rounds is $n+1$. Another important feature of the proposed protocols is no-open property. No cards are opened until the end of the protocol.**

***Index Terms*—Multi-party secure computation, card-based cryptographic protocols, private input operations, millionaires' problem**

I. Introduction

The millionaires' problem [1] is one of the fundamental problems in cryptography. Two players, Alice and Bob, want to know which of them is richer without revealing their actual amount of asset. Many cryptographic protocols were proposed to solve the problem [2]–[7].

Card-based cryptographic protocols [8], [9] have been proposed in which physical cards are used instead of computers to securely calculate values. These protocols are useful when computers cannot be used. They are also useful among people who are not familiar with cryptography. den Boer [10] first showed a five card protocol to securely calculate logical AND of two inputs. Since then, many protocols have been proposed to calculate general logical functions [11]–[22] and specific computations such as computations on three inputs [23], [24], voting [25], [26], random permutation [27], [28], grouping [29], zero-knowledge proof [30], and the millionaires' problem.

If Yao's original protocol is executed by a card-based protocol, the number of cards used in the protocol depends on the size of assets [31]. Thus, a new protocol that the number of cards used in the protocol is a constant is necessary. This paper aims to obtain card-based protocols whose number of cards are a constant.

In order to securely calculate a function using physical cards, some primitives whose result is unknown are necessary. If every primitive execution result is known to a player, some information about private input values is known to the player by reversely executing the protocol in his/her mind using the final outputs. There are two types of primitives whose results are unknown to the players: randomizations and private operations.

Randomization is shuffling some number of cards so that the result is unknown to the players. Many recent protocols use random bisection cuts [13], which randomly execute swapping two decks of cards or doing nothing. The result must be unknown to the players. Several realization methods of the randomization are discussed [32].

The other type of primitives is private operations. Private operations are primitives executed by a player at some place where the other players cannot see. They are realized by executing the operation in a different room, under the table, and so on. There are three types of private operations: private randomization, private reveal operation, and private input operation.

Private randomization is a randomization executed by a player whose result is known to the player but unknown to the other players.

Private reveal operation is opening some cards at some place where the other players cannot see. The player who executes this type of operation must not know the private value, thus the cards must be privately randomized by another player before the private reveal.

With the combination of private randomizations and private reveals, Nakai et al. proposed a protocol to solve the millionaires' problem [31]. The number of cards used by their protocol is $3n+2$, where n is the maximum number of the bits of the asset.

Though the combination of the private randomizations and the private reveals is effective, it is better if the private reveal

978-1-5386-7380-5/18/$31.00 ©2018 IEEE
DOI 10.1109/AsiaJCIS.2018.00013

operations are not used. In order to execute the private reveals, the cards must be able to be relatively easily opened. Such "easy to open" cards might lead the players' mistakes during the protocol execution. If a player make a mistake to open a card that is not allowed, some private information is leaked to the player. If no cards are opened until the end of the protocol, that is, the time when the final output is provided, the cards can be made so that they are not easy to be opened, for example, the faces of the cards can be sealed. Such a protection is effective for preventing mistakes and cheats that a player sees the faces of the cards that are not allowed. Such a hard protection is not applicable when the players need to execute private reveal operations.

The last type of operations is private input operations. When a player has his/her own private input values that must not be disclosed to the other players, the player sets his/her input values at a place where the other players cannot see. Several works have been done to calculate logical functions using private input operations [33], [34]. Though these protocols also satisfy no-open property, the papers do not mention the property. This paper proposes new protocols for the millionaires' problem using private input operations.

Nakai et al.'s paper [31] says their protocol can be modified to use private input operations. They do not explicitly show the protocol, but the protocol will have the following properties.

- The number of cards is 4.
- The number of rounds is $2n + 1$.
- Two types of outputs, $a \geq b$ or $a < b$.
- Private reveal operations are used.

This paper shows two new card-based cryptographic protocols for the millionaires' problem. The first protocol has the following properties.

- The number of cards is 5.
- The number of rounds is $n + 1$.
- Three types of outputs, $a > b$, $a < b$ or $a = b$.
- Private reveal operations are not used, that is, no cards are opened until the end of the protocol.

The equality can be detected, the number of rounds is small, and no cards are opened until the end of the protocol.

The second protocol has the following properties.

- The number of cards is 4.
- The number of rounds is $n + 1$.
- Two types of outputs, $a \geq b$ or $a < b$.
- Private reveal operations are not used.

Though the second protocol cannot distinguish $a > b$ and $a = b$, the number of cards is decreased.

II. Preliminaries

This section gives the notation and basic definitions of card-based protocols. This paper is based on the standard two type card model. In the two type card model, there are two kinds of marks, ♣ and ♥. Cards of the same marks cannot be distinguished. In addition, the back of both types of cards is ?. It is impossible to determine the mark in the back of a given card with ?.

One bit of data is represented by two cards as follows: ♣♥ = 0 and ♥♣ = 1.

One pair of cards that represents one bit $x \in \{0, 1\}$, whose face is down, is called a commitment of x, and denoted as $commit(x)$. Note that if the two cards of $commit(x)$ are swapped, $commit(\bar{x})$ is obtained.

A linearly ordered card is called a sequence of cards. A sequence of cards S whose length is n is denoted as $S = s_1, s_2, \ldots, s_n$, where s_i is the i-th card of the sequence. $S = \underbrace{?}_{s_1}\underbrace{?}_{s_2}\underbrace{?}_{s_3}\ldots,\underbrace{?}_{s_n}$. $S_1||S_2$ is a concatenation of sequence S_1 and S_2.

All protocols are executed by multiple players. Throughout of this paper, all players are semi-honest, that is, they obey the rule of the protocols, but try to obtain private information. There is no collusion among players executing one protocol together. No player wants any other player to obtain his/her own private information.

Card-based protocols using private operations are evaluated by the following criteria.

- The number of cards used in the protocol.
- The number of rounds.

In the first round of a protocol, each player executes primitives which include sending cards to another player, but do not include receiving cards from another player. In the i-th ($i > 1$) round of the protocol, each player receives the cards sent from another player in the $(i-1)$-th round and executes primitives which includes sending cards to another player. The number of rounds of a protocol is the maximum number of rounds among all possible inputs. Since each primitive is relatively simple, handing cards between players and setting up so that the cards are not seen by the other players is the dominating time to execute private operations. Thus the number of rounds is the most appropriate criteria to evaluate the time complexity of card-based protocols using private operations.

III. Outline of Nakai et al.'s protocol

This section shows Nakai et al.'s protocol in [31], which is modified to use a constant number of cards using private input operations.

Assumption 1: Alice and Bob have their private input a and b, respectively. The number of bits of the input values is n, which is decided before the execution of the protocol. Let $a(i)$ and $b(i)$ be the i-th bit of a and b, where LSB(Least Significant Bit) is $a(1)$ and $b(1)$.

Note that the names of variables (cards) are changed from the original paper so that similar variables in our protocol have similar names. The protocol is executed bit by bit from LSB. The protocol uses two cards $s_1^{(i)}s_2^{(i)}$ to store intermediate states. The result up to i-th bit is stored in $s_1^{(i)}$. The final result, given by $s_1 = s_1^{(n)}$ is as follows:

- $a \geq b$ if $s_1 =$ ♣
- $a < b$ if $s_1 =$ ♥

During the execution, one bit data is sent by one card. If the data is 1, it is represented by $\boxed{\heartsuit}$ and if the data is 0, it is represented by $\boxed{\clubsuit}$. In the following protocol, single $\boxed{\heartsuit}$ means 1 and single $\boxed{\clubsuit}$ means 0.

Protocol 1: Nakai et al.'s protocol [31].

1) Let $s_1^{(0)}s_2^{(0)} = \boxed{\clubsuit}\boxed{\heartsuit}$. Note that these cards are face down.
2) Execute the following procedure for $i = 1, \ldots n$.
   a) Alice privately selects a random bit r. If $r = 0$, Alice privately sets $x_1x_2 = s_1^{(i-1)}s_2^{(i-1)}$, otherwise privately sets $x_1x_2 = s_2^{(i-1)}s_1^{(i-1)}$. Alice privately sets $y = \boxed{\clubsuit}$ if $a(i) \oplus r = 0$ otherwise sets $y = \boxed{\heartsuit}$. Note that y is face down.
   b) Alice sends x_1, x_2, and y to Bob.
   c) Bob privately opens y. If ($y = \boxed{\clubsuit}$ and $b(i) = 0$) or ($y = \boxed{\heartsuit}$ and $b(i) = 1$), Bob privately replaces x_2 with one card that represents $b(i)$, otherwise (($y = \boxed{\clubsuit}$ and $b(i) = 1$) or ($y = \boxed{\heartsuit}$ and $b(i) = 0$)), Bob privately replaces x_1 with one card that represents $b(i)$.
   Note that the unused card is discarded.
   d) Bob hands x_1x_2 to Alice.
   e) Alice privately sets $s_1^{(i)}s_2^{(i)} = x_1x_2$ if $r = 0$, otherwise privately sets $s_1^{(i)}s_2^{(i)} = x_2x_1$.

The result is given by $s_1^{(n)}$.

Initially, two cards are used to set $s_1^{(0)}s_2^{(0)}$. After the initialization, Alice needs one $\boxed{\heartsuit}$ and one $\boxed{\clubsuit}$ to set an input value in each round. Thus, the total number of cards used in the protocol is 4. The number of rounds is $2n + 1$.

IV. New protocol for the millionaires' problem

First, we show a simple protocol to show the idea. Then the protocol is modified to decrease the number of rounds. The assumption of the protocol is just the same as Assumption 1.

The protocol is executed bit by bit from LSB. The result is given by two cards s_1s_2 as follows:

- $a > b$ if $s_1s_2 = \boxed{\heartsuit}\boxed{\clubsuit}$
- $b > a$ if $s_1s_2 = \boxed{\clubsuit}\boxed{\heartsuit}$
- $a = b$ if $s_1s_2 = \boxed{\clubsuit}\boxed{\clubsuit}$

Note that $s_1s_2 = \boxed{\heartsuit}\boxed{\heartsuit}$ never occurs by the protocol.

Let $s_1^{(i)}s_2^{(i)}$ be the result up to the i-th bit. Thus the final result $s_1s_2 = s_1^{(n)}s_2^{(n)}$.

Protocol 2: The protocol for the first bit: ($i = 1$)
(Input) $a(1)$, $b(1)$.
(Output) $s_1^{(1)}s_2^{(1)}$.

1) Alice hands $commit(a(1))$ to Bob.
2) Let the received card sequence be x_1x_2. Bob privately sets $s_1^{(1)}s_2^{(1)}$ by the following rule.
   - $s_1^{(1)} = x_1$, $s_2^{(1)} = \boxed{\clubsuit}$ if $b(1)$=0
   - $s_1^{(1)} = \boxed{\clubsuit}$, $s_2^{(1)} = x_2$ if $b(1)$=1

   Note that all cards are face down. Bob discards the unused card.

Protocol 3: The protocol for the i-th bit: ($i > 1$)
(Input) $s_1^{(i-1)}s_2^{(i-1)}$, $a(i)$, $b(i)$.
(Output) $s_1^{(i)}s_2^{(i)}$.

1) Alice privately sets card sequence S as follows:
   - $S = s_1^{(i-1)}s_2^{(i-1)}||\boxed{\clubsuit}\boxed{\heartsuit}$ if $a(i) = 0$
   - $S = \boxed{\heartsuit}\boxed{\clubsuit}||s_1^{(i-1)}s_2^{(i-1)}$ if $a(i) = 1$

   Note that all cards are face down. Alice hands S to Bob.
2) Bob privately selects left two cards if $b(i) = 0$, otherwise selects right two cards as the output $s_1^{(i)}s_2^{(i)}$. Bob discards the unused cards.

The reason the proposed protocol is simplified and has no open property is that the result that Alice wins equals to the commitment of $a(i)$. The result that Bob wins equals to the commitment of $b(i)$. Thus obtaining the comparison result is whether to replace the previous result with the commitment of input values.

Theorem 1: The protocol is correct, secure, and uses at most four cards at any instant.

Proof: Correctness: For the first bit, $x_1x_2 = \boxed{\heartsuit}\boxed{\clubsuit}$ if $a(1) = 1$. In this case, Bob sets $s_1^{(1)}s_2^{(1)} = \boxed{\heartsuit}\boxed{\clubsuit}$ if $b(1) = 0$ and $s_1^{(1)}s_2^{(1)} = \boxed{\clubsuit}\boxed{\clubsuit}$ if $b(1) = 1$. Thus, the result is correct when $a(1) = 1$.

$x_1x_2 = \boxed{\clubsuit}\boxed{\heartsuit}$ if $a(1) = 0$. In this case, Bob sets $s_1^{(1)}s_2^{(1)} = \boxed{\clubsuit}\boxed{\clubsuit}$ if $b(1) = 0$ and $s_1^{(1)}s_2^{(1)} = \boxed{\clubsuit}\boxed{\heartsuit}$ if $b(1) = 1$. Thus, the result is correct when $a(1) = 0$.

For the i-th ($i > 1$) bit, the desired output is as follows:

$$s_1^{(i)}s_2^{(i)} = \begin{cases} \boxed{\heartsuit}\boxed{\clubsuit} & \text{if } a(i) = 1 \text{ and } b(i) = 0 \\ \boxed{\clubsuit}\boxed{\heartsuit} & \text{if } a(i) = 0 \text{ and } b(i) = 1 \\ s_1^{(i-1)}s_2^{(i-1)} & \text{if } a(i) = b(i) \end{cases} \quad (1)$$

When $a(i) = 0$, $S = s_1^{(i-1)}s_2^{(i-1)}||\boxed{\clubsuit}\boxed{\heartsuit}$ is handed to Bob. Bob selects left two cards if $b(i) = 0$, thus the result is $s_1^{(i-1)}s_2^{(i-1)}$ and the output is correct. Bob selects right two cards if $b(i) = 1$, thus the result is $\boxed{\clubsuit}\boxed{\heartsuit}$ and the output is correct.

When $a(i) = 1$, $S = \boxed{\heartsuit}\boxed{\clubsuit}||s_1^{(i-1)}s_2^{(i-1)}$ is handed to Bob. Bob selects left two cards if $b(i) = 0$, thus the result is $\boxed{\heartsuit}\boxed{\clubsuit}$ and the output is correct. Bob selects right two cards if $b(i) = 1$, thus the result is $s_1^{(i-1)}s_2^{(i-1)}$ and the output is correct.

At the end of the protocol, no information other than the comparison result is obtained from the output cards. During the execution of the protocol, no cards are opened. Thus, Alice and Bob obtain no information other than the final comparison result.

At most four cards are used at any instant of the protocol. ■

A note is necessary for the number of cards. Four cards are sufficient if the marks of the discarded cards can be easily erased by some machine without revealing the faces and a new mark can be printed again and again. If the marks of discarded cards cannot be erased, a careful treatment of discarded cards is necessary. The discarded cards must be mixed with the other cards (no one knows the number of cards of each mark) and new cards necessary for the next round, one ♥ and one ♣, must be picked up. Alice uses these cards to set S. The marks of the discarded cards leak the privacy of Alice or Bob. For example, when $i = 1$ and the discarded card is ♥, it means that $a(1) = b(1)$.

If there are very few number of cards so that the above mixing of discarded cards is impossible, the discarded cards must be handed to TTP(Trusted Third Party) and new cards must be handed to the players from the TTP. TTP knows the secret information from the marks of the discarded cards, but TTP does not disclose the information. In the case, the TTP must initially have at least five cards, three ♣ cards and two ♥ cards, though four cards are used at any instant. When $i = 1$, two ♣ cards and one ♥ card are necessary. When $i > 1$, one ♣ card and one ♥ card are necessary for Alice to set S. The discarded card when $i = 1$ might be one ♥ or one ♣. If the discarded card is ♥ (that is, $a(1) = b(1)$), another ♣ card is necessary for $i = 2$. If $a(2) > b(2)$, the discarded cards for $i = 2$ are two ♣s. Thus another ♥ card is necessary for $i = 3$. This is the worst case and three ♣ and two ♥ cards are necessary. Thus in the worst case, five cards are necessary.

V. Round-efficient protocol

The protocol in the previous section can be modified to decrease the number of rounds. The protocol is initiated by Alice for every i. Thus, the number of rounds is $2n-1$, because the cards must be handed back from Bob to Alice at the end of procedure for each i other than the case $i = n$. Since the protocol is symmetric between Alice and Bob, the protocol can also be initiated from Bob.

The previous section's protocol for $i > 1$ is used only when i is odd. If i is even the protocol below is used instead.

Protocol 4: The protocol for the i-th bit: ($i > 1$ and i is even)

(Input) $s_1^{(i-1)}s_2^{(i-1)}$, $a(i)$, $b(i)$.

(Output) $s_1^{(i)}s_2^{(i)}$.

1) Bob privately sets card sequence S as follows:
   - $S = s_1^{(i-1)}s_2^{(i-1)}||$♥♣ if $b(i) = 0$
   - $S =$ ♣♥$||s_1^{(i-1)}s_2^{(i-1)}$ if $b(i) = 1$

   Note that all cards are face down. Bob hands S to Alice.
2) Alice privately selects left two cards if $a(i) = 0$, otherwise selects right two cards as the output $s_1^{(i)}s_2^{(i)}$. Alice discards the unused cards.

The modified protocol is executed as follows.

- Alice executes step (1) of $i = 1$.
- Alice hands S to Bob.
- Bob executes step (2) of $i = 1$ and then step (1) of $i = 2$.
- Bob hands S to Alice.
- Alice executes step (2) of $i = 2$ and then step (1) of $i = 3$.
- ...
- If n is even(odd), Alice(Bob) executes step (2) of $i = n$.

Handing cards is executed once for each i, thus the number of rounds is $n + 1$.

Example 1: Suppose that $n = 3$, $a = 110$ and $b = 010$.

The protocol is executed as follows. Note that all cards are face down.

1) $i = 1$: Alice hands $commit(0) =$ ♣♥ to Bob since $a(1) = 0$.
2) Let the received card be x_1x_2. Since $b(1) = 0$, Bob selects $x_1 =$ ♣, and sets $s_1^{(1)}s_2^{(1)} =$ ♣♣. Note that the result is correct for $i = 1$.
   $i = 2$: Bob sets $S =$ ♣♥$||s_1^{(1)}s_2^{(1)}$, which is ♣♥♣♣, since $b(2) = 1$. Bob hands S to Alice.
3) Alice selects right two cards of S, since $a(2) = 1$. The result, $s_1^{(2)}s_2^{(2)} =$ ♣♣. Note that the result is correct for $i = 2$.
   $i = 3$: Alice sets $S =$ ♥♣$||s_1^{(2)}s_2^{(2)}$, which is ♥♣♣♣, since $a(3) = 1$. Alice hands S to Bob.
4) Bob selects left two cards of S, since $b(3) = 0$. The result, $s_1^{(3)}s_2^{(3)} =$ ♥♣.
   The final result, $s_1^{(3)}s_2^{(3)}$, means that $a > b$ since the cards are ♥♣.

Theorem 2: The modified protocol is correct and secure.

Proof: The proof of the security and the number of cards used by the protocol is just the same as Theorem 1. We just show the correctness of the protocol when $i > 1$ and i is even. The desired output is the same as the equation (1).

When $b(i) = 0$, $S = s_1^{(i-1)}s_2^{(i-1)}||$♥♣ is handed to Alice. Alice selects left two cards if $a(i) = 0$, thus the result is $s_1^{(i-1)}s_2^{(i-1)}$ and the output is correct. Alice selects right two cards if $a(i) = 1$, thus the result is ♥♣ and the output is correct.

When $b(i) = 1$, $S =$ ♣♥$||s_1^{(i-1)}s_2^{(i-1)}$ is handed to Alice. Alice selects left two cards if $a(i) = 0$, thus the result is ♣♥ and the output is correct. Alice selects right two cards if $a(i) = 1$, thus the result is $s_1^{(i-1)}s_2^{(i-1)}$ and the output is correct. ■

VI. Protocol with fewer number of cards

This section shows another protocol with the following properties.

- The number of cards is 4.
- The number of rounds is $n + 1$.

- Two types of outputs, $a \geq b$ or $a < b$.
- Private reveal operations are not used.

For the protocol in the previous section, two cards are necessary to obtain three different outputs. If the output is one bit ($a \geq b$ or $a < b$), one card is sufficient to remember, thus the total number of cards can be decreased.

The assumption of the protocol is just the same as Assumption 1.

The protocol is executed bit by bit from LSB. The result is given by one card s as follows:

- $a \geq b$ if $s =$  .
- $a < b$ if $s =$  .

Let $s^{(i)}$ be the result up to the i-th bit. Thus the final result $s = s^{(n)}$.

Protocol 5:

The protocol for the first bit: $(i = 1)$:
(Input) $a(1)$, $b(1)$.
(Output) $s^{(1)}$.

1) Alice privately sets $x_1x_2 =$  if $a(1) = 1$, otherwise sets $x_1x_2 =$  . Note that all cards are face down. Alice hands x_1x_2 to Bob.
2) Bob privately sets $s^{(1)} = x_1$ if $b(1) = 1$, otherwise sets $s^{(1)} = x_2$. Bob discards the unused card.

The protocol for the i-th bit: $(i > 1)$ when i is even:
(Input) $s^{(i-1)}$, $a(i)$, $b(i)$.
(Output) $s^{(i)}$.

1) Bob privately sets card sequence S as follows:
   - $S = s^{(i-1)}||$ if $b(i) = 1$
   - $S =$  $||s^{(i-1)}$ if $b(i) = 0$

   Note that all cards are face down. Bob hands S to Alice.
2) Alice privately selects the left card if $a(i) = 1$, otherwise selects the right card as the output $s^{(i)}$. Alice discards the unused card.

The protocol for the i-th bit: $(i > 1)$ when i is odd:
(Input) $s^{(i-1)}$, $a(i)$, $b(i)$.
(Output) $s^{(i)}$.

1) Alice privately sets card sequence S as follows:
   - $S = s^{(i-1)}||$ if $a(i) = 1$
   - $S =$  $||s^{(i-1)}$ if $a(i) = 0$

   Note that all cards are face down. Alice hands S to Bob.
2) Bob privately selects the left card if $b(i) = 1$, otherwise selects the right card as the output $s^{(i)}$. Bob discards the unused card.

An example for the same input is shown.

Example 2: Suppose that $n = 3$, $a = 110$ and $b = 010$.

The protocol is executed as follows. Note that all cards are face down.

1) $i = 1$: Alice hands  to Bob since $a(1) = 0$.
2) Let the received card be x_1x_2. Since $b(1) = 0$, Bob selects $x_2 =$  as $s^{(1)}$. Note that the result is correct for $i = 1$.
   $i = 2$: Bob sets $S = s^{(1)}||$, which is  , since $b(2) = 1$. Bob hands S to Alice.
3) Alice selects the left card of S, since $a(2) = 1$. The result, $s^{(2)} =$  . Note that the result is correct for $i = 2$.
   $i = 3$: Alice sets $S = s^{(2)}||$, which is  , since $a(3) = 1$. Alice hands S to Bob.
4) Bob selects the right card of S, since $b(3) = 0$. The result, $s^{(3)} =$  .
   The final result, $s^{(3)}$, means that $a \geq b$ since the card is .

Theorem 3: Protocol 5 is correct and secure.

Proof: Proof of the security of the protocol is just the same as the previous protocol. Alice and Bob open no cards other than the final output. Thus they obtains no information other than the final output.

Next we show the correctness.

For the first bit $(i = 1)$, Alice sets  if $a(1) = 1$. In this case, the result $s^{(1)} =$  regardless whether $b(1) = 0$ or 1. The result is correct since $a(1) \geq b(1)$.

When $a(1) = 0$, Alice sets  . In this case, Bob selects  if $b(1) = 1$. The output is correct since $a(1) < b(1)$. Bob selects  if $b(1) = 0$. The output is correct since $a(1) \geq b(1)$.

For the i-th bit $(i > 1)$, the desired output is as follows:

$$s^{(i)} = \begin{cases} \text{  } & \text{if } a(i) = 1 \text{ and } b(i) = 0 \\ \text{  } & \text{if } a(i) = 0 \text{ and } b(i) = 1 \\ s^{(i-1)} & \text{if } a(i) = b(i) \end{cases} \tag{2}$$

First consider the case when $i(> 1)$ is even. Bob sets $S = s^{(i-1)}||$  if $b(i) = 1$. Alice selects the left card and obtains $s^{(i-1)}$ if $a(i) = 1$. This output is correct. Alice selects the right card and obtains  if $a(i) = 0$. This output is correct.

Bob sets $S =$  $||s^{(i-1)}$ if $b(i) = 0$. Alice selects the left card and obtains if $a(i) = 1$. This output is correct. Alice selects the right card and obtains $s^{(i-1)}$ if $a(i) = 0$. This output is correct.

Next consider the case when $i(> 1)$ is odd. Alice sets $S = s^{(i-1)}||$  if $a(i) = 1$. Bob selects the left card and obtains $s^{(i-1)}$ if $b(i) = 1$. This output is correct. Bob selects the right card and obtains  if $b(i) = 0$. This output is correct.

Alice sets $S =$ $||s^{(i-1)}$ if $a(i) = 0$. Bob selects the left card and obtains  if $b(i) = 1$. This output is correct. Bob selects the right card and obtains $s^{(i-1)}$ if $b(i) = 0$. This output is correct. ■

At any instant of the protocol, at most three cards are used. Thus, if there are many cards, at most three cards are used, by a discussion similar to the previous protocol. If there are very few number of cards and TTP hands cards to Alice and Bob, the number of cards necessary for the execution is 4.

When $i = 1$, two $\heartsuit$s and one $\clubsuit$ are necessary for Alice to set x_1x_2. At the end of $i = 1$, two cards are unused. When $i > 1$, Alice (or Bob) needs to have one $\heartsuit$ and one $\clubsuit$ to set S. In the worst case, two unused cards at the end of $i = 1$ are two $\heartsuit$s, thus one new $\clubsuit$ is necessary. This is the worst case and four cards are necessary.

VII. CONCLUSION

This paper showed new card-based cryptographic protocols for the millionaires' problem with a small number of rounds. The result shows the effectiveness of private input operations. Very few works have been done for card-based protocols with private input operations. Especially, no-open property seems to be impossible without private input operations. We think that no-open property is a very important property for card-based cryptographic protocols. Obtaining no-open card-based cryptographic protocols for the other fundamental problems is an open problem.

REFERENCES

[1] A. C. Yao, "Protocols for secure computations," in *Proc. of 23rd Symposium on Foundations of Computer Science (FOCS)*. IEEE, 1982, pp. 160–164.

[2] I. Ioannidis and A. Grama, "An efficient protocol for yao's millionaires' problem," in *Proc. of the 36th Annual Hawaii International Conference on System Sciences*. IEEE, 2003, pp. 1–6.

[3] L. Shundong, W. Daoshun, D. Yiqi, and L. Ping, "Symmetric cryptographic solution to yao's millionairess' problem and an evaluation of secure multiparty computations," *Information Sciences*, vol. 178, no. 1, pp. 244–255, 2008.

[4] S.-D. Li, Y.-Q. Dai, and Q.-Y. You, "Efficient solution to yao's millionaires' problem." *Dianzi Xuebao(Acta Electronica Sinica)*, vol. 33, no. 5, pp. 769–773, 2005.

[5] H.-Y. Lin and W.-G. Tzeng, "An efficient solution to the millionaires' problem based on homomorphic encryption," in *Proc. of International Conference on Applied Cryptography and Network Security, LNCS Vol. 3531*. Springer, 2005, pp. 456–466.

[6] D. Malkhi, N. Nisan, B. Pinkas, Y. Sella *et al.*, "Fairplay-secure two-party computation system." in *USENIX Security Symposium*, vol. 4. San Diego, CA, USA, 2004, p. 9.

[7] M. Jakobsson and A. Juels, "Mix and match: Secure function evaluation via ciphertexts," in *Proc. of Asiacrypt 2000, LNCS Vol.1976*. Springer, 2000, pp. 162–177.

[8] T. Mizuki, "Secure multi-party protocols using a deck of cards," *IEICE Fundamental Review*, pp. 179–187, 2016, (In Japanese).

[9] T. Mizuki and H. Shizuya, "Computational model of card-based cryptographic protocols and its applications," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 100, no. 1, pp. 3–11, 2017.

[10] B. den Boer, "More efficient match-making and satisfiability the five card trick," in *Proc. of EUROCRYPT '89, LNCS Vol. 434*, 1990, pp. 208–217.

[11] T. Mizuki, "Card-based protocols for securely computing the conjunction of multiple variables," *Theoretical Computer Science*, vol. 622, pp. 34–44, 2016.

[12] D. Francis, S. R. Aljunid, T. Nishida, Y. Hayashi, T. Mizuki, and H. Sone, "Necessary and sufficient numbers of cards for securely computing two-bit output functions," in *Proc. of Second International Conference on Cryptology and Malicious Security(Mycrypt 2016), LNCS Vol. 10311*, 2017, pp. 193–211.

[13] T. Mizuki and H. Sone, "Six-card secure and and four-card secure xor," in *Proc. of 3rd International Workshop on Frontiers in Algorithms(FAW 2009), LNCS Vol. 5598*, 2009, pp. 358–369.

[14] T. Mizuki, M. Kumamoto, and H. Sone, "The five-card trick can be done with four cards," *Proc. of Asiacrypt 2012, LNCS Vol.7658*, pp. 598–606, 2012.

[15] C. Crépeau and J. Kilian, "Discreet solitary games," in *Proc. of 13th Crypto, LNCS Vol. 773*, 1993, pp. 319–330.

[16] V. Niemi and A. Renvall, "Secure multiparty computations without computers," *Theoretical Computer Science*, vol. 191, no. 1, pp. 173–183, 1998.

[17] A. Stiglic, "Computations with a deck of cards," *Theoretical Computer Science*, vol. 259, no. 1, pp. 671–678, 2001.

[18] A. Koch, S. Walzer, and K. Härtel, "Card-based cryptographic protocols using a minimal number of cards," in *Proc. of Asiacrypt 2015, LNCS Vol. 9452*, 2015, pp. 783–807.

[19] A. Nishimura, T. Nishida, Y. Hayashi, T. Mizuki, and H. Sone, "Five-card secure computations using unequal division shuffle," in *Proc. of 4th International Conference on Theory and Practice of Natural Computing(TNPC 2015), LNCS Vol. 9477*, 2015, pp. 109–120.

[20] T. Nishida, Y. Hayashi, T. Mizuki, and H. Sone, "Card-based protocols for any boolean function," in *Proc. of 15th International Conference on Theory and Applications of Models of Computation(TAMC 2015), LNCS Vol. 9076*, 2015, pp. 110–121.

[21] A. Nishimura, T. Nishida, Y. Hayashi, T. Mizuki, and H. Sone, "Card-based protocols using unequal division shuffles," *Soft Computing*, pp. 1–11, 2017.

[22] J. Kastner, A. Koch, S. Walzer, D. Miyahara, Y. Hayashi, T. Mizuki, and H. Sone, "The minimum number of cards in practical card-based protocols," in *Proc. of 23rd International Conference on the Theory and Applications of Cryptology and Information Security(ASIACRYPT2017), Part III, LNCS Vol. 10626*, 2017, pp. 126–155.

[23] T. Nishida, Y. Hayashi, T. Mizuki, and H. Sone, "Securely computing three-input functions with eight cards," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 98, no. 6, pp. 1145–1152, 2015.

[24] T. Nishida, T. Mizuki, and H. Sone, "Securely computing the three-input majority function with eight cards," in *Proc. of 2nd International Conference on Theory and Practice of Natural Computing(TPNC 2013), LNCS Vol. 8273*, 2013, pp. 193–204.

[25] T. Mizuki, I. K. Asiedu, and H. Sone, "Voting with a logarithmic number of cards," in *Proc. of International Conference on Unconventional Computing and Natural Computation (UCNC 2013), LNCS Vol. 7956*, 2013, pp. 162–173.

[26] 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 *Proc. of 10th International Conference on Information Theoretic Security (ICITS 2017), LNCS Vol. 10681*, 2017, pp. 153–165.

[27] T. Ibaraki and Y. Manabe, "A more efficient card-based protocol for generating a random permutation without fixed points," in *Proc. of 3rd International Conference on Mathematics and Computers in Sciences and in Industry (MCSI 2016)*, 2016, pp. 252–257.

[28] R. Ishikawa, E. Chida, and T. Mizuki, "Efficient card-based protocols for generating a hidden random permutation without fixed points," in *Proc. of 14th International Conference on Unconventional Computation and Natural Computation(UCNC 2015), LNCS Vol. 9252*, 2015, pp. 215–226.

[29] Y. Hashimoto, K. Shinagawa, K. Nuida, M. Inamura, and G. Hanaoka, "Secure grouping protocol using a deck of cards," in *Proc. of 10th International Conference on Information Theoretic Security(ICITS 2017), LNCS Vol. 10681*, 2017, pp. 135–152.

[30] T. Sasaki, T. Mizuki, and H. Sone, "Card-based zero-knowledge proof for sudoku," in *LIPIcs-Leibniz International Proceedings in Informatics*, vol. 100. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[31] T. Nakai, Y. Tokushige, Y. Misawa, M. Iwamoto, and K. Ohta, "Efficient card-based cryptographic protocols for millionaires' problem utilizing private permutations," in *Proc. of International Conference on Cryptology and Network Security(CANS 2016), LNCS Vol. 10052*, 2016, pp. 500–517.

[32] I. Ueda, A. Nishimura, Y. Hayashi, T. Mizuki, and H. Sone, "How to implement a random bisection cut," in *Proc. of 5th International Conference on Theory and Practice of Natural Computing (TPNC 2016), LNCS Vol. 10071*, 2016, pp. 58–69.

[33] K. Kurosawa and T. Shinozaki, "Compact card protocol," in *Proc. of 2017 Symposium on Cryptography and Information Security(SCIS 2017)*, 2017, pp. 1A2–6, (In Japanese).

[34] S. Shirouchi, T. Nakai, M. Iwamoto, and K. Ohta, "Efficient card-based cryptographic protocols for logic gates utilizing private permutations," in *Proc. of 2017 Symposium on Cryptography and Information Security(SCIS 2017)*, 2017, pp. 1A2–2, (In Japanese).