CS 758 Project:
Secure Computation with Playing Cards

Eddie Cheung Chris Hawthorne Patrick Lee

December 15, 2013

1 Abstract

There exist several card-based protocols for the secure computation of a boolean AND; currently the best known committed protocols use 6 cards. We present a new Las Vegas committed protocol that uses 5 cards, reducing the number of cards needed. Mizuki et al. have presented several card-based protocols that can be expressed as a permutation of the form xgx^{-1}. We show that any desired permutation h with the same cycle structure as g can be obtained using a permutation of the form xgx^{-1} and we give a technique to compute such a permutation.

2 Problem Statement

Alice and Bob are on a blind-date and they wish to determine if they share some particular mutual interest, but naturally both parties refuse to show interest first as they wish to avoid an embarrassing "No" from the other party. More formally, Alice and Bob are two *honest-but-curious* players, each with a secret bit, $a \in \{0, 1\}$ and $b \in \{0, 1\}$ respectively, they want to *securely* compute the **AND** of their bits, but do not want to reveal more information about their bits than necessary. For example, if Alice's bit is 1 then it is unavoidable that she learns the value of Bob's bit, but if Alice's bit is 0 then Alice should not be able to determine the value of Bob's bit.

In this paper we use two suits $\boxed{\clubsuit}$ and $\boxed{\heartsuit}$, and we assume all cards of the same suit are identical. We denote a face-down card as $\boxed{?}$. We further assume that the backs of all the cards are identical.

We define an encoding scheme by which one can encode a bit as a pair of cards:

$$0 \mapsto \boxed{\clubsuit}\boxed{\heartsuit}$$
$$1 \mapsto \boxed{\heartsuit}\boxed{\clubsuit}$$

For example, if Alice wishes to express that she does hold a particular interest, she would place the cards $\boxed{\heartsuit}\boxed{\clubsuit}$ on the table.

Given a protocol for performing a computation using playing cards, we say the protocol is in *committed format* if the output of the protocol is two face-down cards that encode the result in the above encoding scheme. We want protocols to be in committed format as it allows the output of a protocol to be used as the input for another without revealing intermediate results.

3 Previous Works

In 1989, Boer presented the first card based protocol to securely compute the AND of two bits using five playing cards [1]. Many other card-based protocols have since been developed. Table 1 lists two protocols that securely compute the AND in non-committed format. Table 2 lists several protocols that securely compute the AND in committed format. Table 3 lists several protocols that securely compute the XOR in committed format. Notably, Mizuki, Kumamoto, and Sone gave a protocol to securely compute the AND using 4 cards in non-committed format [4]; Mizuki and Sone gave a protocol to securely compute the AND in committed format using 6 cards and a protocol to securely compute the XOR in committed format using 4 cards [5].

The methods for computing AND and XOR securely provide the framework for doing any boolean expression securely. The negation of a committed bit can be easily computed by swapping the cards. It is easy to see how a OR can be computed using De Morgan's Law

$$A \vee B = \neg(\neg A \wedge \neg B).$$

Thus, any boolean expression can be securely computed because the set of operators {AND, NOT, OR} is adequate. A random bit can be generated by taking a committed bit and applying a random cyclic shift to the cards [7] (i.e. randomly decide to either switch the cards or leave them as they are). To obtain k copies of a committed bit we can use the scheme described in [2] that uses $2k+4$ cards, or the scheme of [5] that uses $2k+2$ cards.

4 Applications and Motivations

There are several applications and motivations of card-based protocols, some of which are listed below:

- Card-based protocols can be realized without the use of computers; computers can be expensive, complicated, bulky, and insecure, whereas playing cards are cheap, simple, portable and secure if the cards are unmarked.
- Card-based protocols can be used to create schemes for secure multi-party computation [6], [7].
- A card-based protocol can be used to securely generate permutations with no fixed points [2].

Table 1: Secure AND in a non-committed format; taken from [3]

	# of suits	# of cards	Avg. # of trials
den Boer [1]	2	5	1
Mizuki-Kumamoto-Sone [4]	2	4	1

Table 2: Secure AND in a committed format; taken from [3]

	# of suits	# of cards	Avg. # of trials
Crépeau-Kilian [2]	4	10	6
Niemi-Renvall [6]	2	12	2.5
Stiglic [7]	2	8	2
Mizuki-Sone [5]	2	6	1

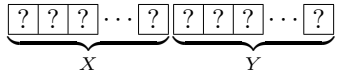
Table 3: Secure XOR in a committed format; taken from [3]

	# of colors	# of cards	Avg. # of trials
Crépeau-Kilian [2]	4	14	6
Mizuki-Uchiike-Sone [8]	2	10	2
Mizuki-Sone [5]	2	4	1

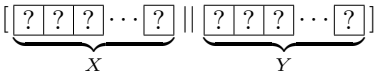
- Playing card games in solitary can be achieved by modelling an opponent as a probabilistic boolean circuit and realizing the opponent using card-based protocols [2].
- Card-based protocols can be used to obtain zero-knowledge proofs for satisfiability by modelling the satisfiability problem as a boolean circuit and using card-based protocols to execute the circuit [6],[7].
- Mizuki presented a card-based protocol for secure two-party voting and card-based protocols for secure half-adders and full-adders [3].
- Card-based protocols tend to be very simple, and can be used as a teaching aid to illustrate the principles of cryptography to non-specialists [4].

5 6-Card Committed AND

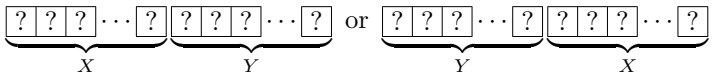
As a motivating example, we will present Mizuki and Sone's 6-card committed AND protocol [5]. We begin by formally defining a "random bisection cut." Given a sequence of $2m$ face-down cards,



with $|X| = |Y| = m$, we define a *random bisection cut* (denoted by $[\,\cdot \,||\, \cdot\,]$) as



which results in one of two possibilities:

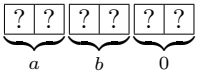


each with probability $\frac{1}{2}$. (i.e. Switch X and Y with probability $\frac{1}{2}$, and leave the cards where they are with probability $\frac{1}{2}$. Later in this paper, we will relax the definition to allow X and Y to have different sizes.

A useful property of the random bisection cut is that Alice and Bob can take turns secretly applying random bisection cuts, and the result will be a random bisection cut whose outcome is unknown to both Alice and Bob.

Mizuki and Sone's protocol can be summarized as follows:

1. Have Alice and Bob encode their bits as $\boxed{\clubsuit}\boxed{\heartsuit} = 0$ or $\boxed{\heartsuit}\boxed{\clubsuit} = 1$.
2. Place their bits facedown along with a 0 bit.



3. Apply the permutation (243).
4. Apply a random bisection cut. Note that after this step, neither player knows whose cards are where; they only know that the sequence of cards must be in one of two possible states.
5. Apply the permutation (234).
6. Reveal the first two cards. If $\boxed{\clubsuit}\boxed{\heartsuit}$ is revealed, output the last two cards. Otherwise, output the middle two cards.

The reader can verify by a straightforward case analysis that this protocol is correct.

6 Conjugation

Mizuki et al. presented several card-based protocols of the following form, where g is some permutation that is physically easy to apply, and x is any permutation:

- Apply x^{-1}
- Apply g with probability $\frac{1}{2}$; else apply the identity permutation
- Apply x

For example, the protocol in the previous section takes this form, where g is the bisection cut and x is (243).

The permutation resulting from this sequence of operations will be the identity with probability $\frac{1}{2}$ and xgx^{-1} with probability $\frac{1}{2}$. It is typically easy to see why the protocol works once we notice the possible resulting permutations, but the choice of x required to get these resulting permutations is not obvious; we present a general approach for finding x derived from group theory.

6.1 The Result

Suppose we wish to apply some $h \in S_n$ to the cards with probability $\frac{1}{2}$, and otherwise apply the identity permutation. Assume we have found some $g \in S_n$ such that g and h have the same cycle structure, and g is physically easy to apply to a deck of cards. If we could find some permutation x such that $h = xgx^{-1}$, then we could simply tell the players to do the following:

- Apply x^{-1}
- Apply g with probability $\frac{1}{2}$; else apply the identity permutation
- Apply x

which would have the effect of applying h to the cards with probability $\frac{1}{2}$, and otherwise applying the identity permutation. We can thus achieve the desired outcome without forcing the players to compute h directly.

We thus wish to find x such that $xgx^{-1} = h$.

Recall the following results from group theory:

Proposition 6.1. *Suppose $\sigma = (a_0 \ldots a_{k-1}) \in S_n$ is a cycle, $\tau \in S_n$ an arbitrary permutation. Then $\tau\sigma\tau^{-1} = (\tau(a_0) \ldots \tau(a_{k-1}))$.*

Which extends easily to products of disjoint cycles, and thus to cycle decompositions:

Corollary 6.2. *Suppose $\sigma \in S_n$ has cycle decomposition*

$$(a_{0,0} \ldots a_{0,k_0})\,(a_{1,0} \ldots a_{1,k_1}) \ldots (a_{\ell,0} \ldots a_{\ell,k_\ell})$$

Then

$$\tau\sigma\tau^{-1} = (\tau(a_{0,0}) \ldots \tau(a_{0,k_0}))\,(\tau(a_{1,0}) \ldots \tau(a_{1,k_1})) \ldots (\tau(a_{\ell,0}) \ldots \tau(a_{\ell,k_\ell}))$$

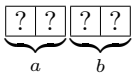
Example 6.3. Suppose $\sigma = (1\,4\,3)(2\,5)$, $\tau = (1\,2)(3\,4)$ in S_5. Then

$$\begin{aligned} \tau\sigma\tau^{-1} &= (\tau(1)\,\tau(4)\,\tau(3))(\tau(2)\,\tau(5)) \\ &= (2\,3\,4)(1\,5) \end{aligned}$$

and indeed one can check by hand that this equality holds.

We can use this to go in the other direction: we can find $x \in S_n$ such that $xgx^{-1} = h$ by simply looking at where we need to send the cycle structure of g to get that of h. (Note that in general this x will not be unique.)

Example 6.4. Suppose Alice and Bob encode their bits in the standard format and place them as follows:



We wish to either negate both of their bits with probability $\frac{1}{2}$. This can be expressed as applying $h = (1\,2)(3\,4)$ with probability $\frac{1}{2}$. Notice that h has the same cycle structure as the familiar bisection cut, which can be expressed as $g = (1\,3)(2\,4)$. Applying the above strategy, we find that $x = (2\,3)$ satisfies $xgx^{-1} = h$. We can thus tell players to flip cards 2 and 3, apply a random bisection cut, and flip cards 2 and 3 again; this will result in either Alice's bit followed by Bob's bit or the negation of Alice's bit followed by the negation of Bob's bit, as desired.

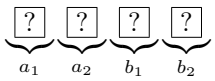
This gives us a general tool for converting complicated but useful permutations into simpler ones of the same cycle structure.

7 Las Vegas 5-card Committed AND

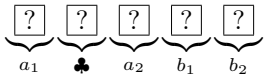
In this section we present our Las Vegas protocol which securely computes the AND of bits a and b using five cards ♣♣♣♡♡. As stated above use ♣♡ to encode 0 and ♡♣ to encode 1.

7.1 The Protocol

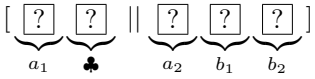
1. Have Alice and Bob encode their bits in the standard format:



2. Place the cards face-down on the table around a face-down club as follows:



3. Apply a random bisection cut to the cards with the cards divided as follows:



4. Reveal the card at position 1.
   If the card is ♣ then

– output the cards at position 2 and position 3 $\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}$

Else the card is $\boxed{\heartsuit}$ then

– we restart the protocol and do another trial.

7.2 Correctness

The correctness of the protocol will be shown through case analysis.

At the end of the protocol, the first card is revealed. If the first card is $\boxed{\heartsuit}$, then we restart the protocol. Suppose instead that the first card is $\boxed{\clubsuit}$. Depending on the outcome of the bisection cut, we fall into one of two cases:

Case 1. Suppose the bisection cut was not applied. Then the cards are

$$\underbrace{\boxed{?}}_{a_1}\underbrace{\boxed{?}}_{\clubsuit}\underbrace{\boxed{?}}_{a_2}\underbrace{\boxed{?}}_{b_1}\underbrace{\boxed{?}}_{b_2}$$

Then a_1 is $\boxed{\clubsuit}$, and we know that Alice's bit is 0; so the result of the AND should be 0. Since a_1 is $\boxed{\clubsuit}$, we know that a_2 is $\boxed{\heartsuit}$; thus cards 2 and 3 are

$$\boxed{\clubsuit}\boxed{\heartsuit}$$

which encodes 0. So outputting cards 2 and 3 yields the correct output.

Case 2. Suppose the bisection cut was applied. Then the cards are

$$\underbrace{\boxed{?}}_{a_2}\underbrace{\boxed{?}}_{b_1}\underbrace{\boxed{?}}_{b_2}\underbrace{\boxed{?}}_{a_1}\underbrace{\boxed{?}}_{\clubsuit}$$

So a_2 is $\boxed{\clubsuit}$. So we know that Alice's bit is 1, and the result of the AND should be Bob's bit. But the cards encoding Bob's bit are in positions 2 and 3; so outputting cards 2 and 3 yields the correct output.

Thus the protocol outputs the correct response in the cases where it runs to completion. So the protocol is correct.

7.3 Security

Let A and B be the random variables corresponding to Alice's bit and Bob's bit, respectively; let F be the random variable that is 1 if the card turned over at the end of the protocol is $\boxed{\heartsuit}$, and 0 otherwise. Suppose we have some prior probability mass function on the values of (A, B). To show secrecy, we show that $P[(A, B) = (a, b) \mid F = f] = P[(A, B) = (a, b)]$ for any $a, b, f \in \{0, 1\}$. By Bayes' theorem, it suffices to show that $P[F = f \mid (A, B) = (a, b)] = P[F = f]$ for all $a, b, f \in \{0, 1\}$.

By examining the possible positions the cards could be in, we see that F is equal to $1 - A$ with probability $\frac{1}{2}$ and equal to A with probability $\frac{1}{2}$. A quick case analysis shows that whether A is 0 or 1, F is uniformly distributed on the set $\{0, 1\}$. So indeed we have that $P[F = f \mid (A, B) = (a, b)] = P[F = f] = \frac{1}{2}$ for any $a, b, f \in \{0, 1\}$.

So no information is given to the observers beyond what they know prior to running the protocol. So secrecy is maintained.

8 Open Problems

There are several open problems:

- We presented a Las Vegas algorithm to securely compute the AND of two bits using five cards. Whether this algorithm can be made deterministic remains open.
- Most card-based protocols are based upon cyclic permutations of a deck of cards ([1],[7]), or a bisection cut ([4],[3]). Niemi and Renvall suggested using more general shuffling operations based upon dihedral group [6]. It remains open whether more general type of shuffle operations can be used to create improved results.
- All card-based protocols assume that the parties are honest-but-curious, that is all parties properly execute the protocol. However, a party may be malicious and may execute the protocol improperly; the impact of such a malicious party has yet to be explored.
- The parties have no method of verifying that the protocol was executed properly. Such a verification step is desirable for many applications such as voting. It is an open problem whether redundancies can be added to the protocols to allow verifiable execution.
- A tangentially related open problem is the following:
  Card-based protocols are ideal for applications for which a computer is too cumbersome, such as equality testing (socialist millionaires), computing the maximum of a set of numbers (secret auctions), and secret sharing (Shamir threshold scheme). Naïve card-based protocols for these application require too many cards. *Efficient* card based-protocols that use a small number of cards and a small number of operations have yet to be discovered.

References

[1] Bert Boer. More efficient match-making and satisfiability the five card trick. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology — EUROCRYPT '89*, volume 434 of *Lecture Notes in Computer Science*, pages 208–217. Springer Berlin Heidelberg.

[2] Claude Crépeau and Joe Kilian. Discreet solitary games. In Douglas R. Stinson, editor, *Advances in Cryptology — CRYPTO' 93*, volume 773 of *Lecture Notes in Computer Science*, pages 319–330. Springer Berlin Heidelberg.

[3] Takaaki Mizuki, Isaac Kobina Asiedu, and Hideaki Sone. Voting with a logarithmic number of cards. In Giancarlo Mauri, Alberto Dennunzio, Luca Manzoni, and AntonioE. Porreca, editors, *Unconventional Computation and Natural Computation*, volume 7956 of *Lecture Notes in Computer Science*, pages 162–173. Springer Berlin Heidelberg.

[4] Takaaki Mizuki, Michihito Kumamoto, and Hideaki Sone. The five-card trick can be done with four cards. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 598–606. Springer Berlin Heidelberg.

[5] Takaaki Mizuki and Hideaki Sone. Six-card secure and and four-card secure xor. In Xiaotie Deng, JohnE. Hopcroft, and Jinyun Xue, editors, *Frontiers in Algorithmics*, volume 5598 of *Lecture Notes in Computer Science*, pages 358–369. Springer Berlin Heidelberg.

[6] Valtteri Niemi and Ari Renvall. Secure multiparty computations without computers. *Theoretical Computer Science*, 191(1-2):173–183.

[7] Anton Stiglic. Computations with a deck of cards. *Theoretical Computer Science*, 259(1–2):671 – 678, 2001.

[8] Fumishige Uchiike Takaaki Mizuki and Hideaki Sone. Securely computing xor with 10 cards. In *Australasian Journal of Combinatorics*, volume 36, pages 279–293.