Cryptography without Computers

Computations with a Deck of Cards

Heiko Stamer <HeikoStamer@gmx.net>

December 28, 2015 (Workshop at 32C3)

We discuss card-based protocols [MSh14] for secure computation of boolean operations, i.e. cryptography without computers [NR98, Sti01]. Our main focus are recently developed protocols for AND and XOR [MSo09] and their implementation as building blocks for *secure multiparty computation* within LibTMCG [1].

Introduction

Consider a deck with only black ($\boxed{\clubsuit}$) and red ($\boxed{\heartsuit}$) suits, where all cards of the same suit are identical. The back side (face-down) of each card is also identically shaped and denoted by $\boxed{?}$. We use the following encoding for a bit by a sequence of two cards:

$$0 \mapsto \boxed{\clubsuit}\boxed{\heartsuit} \qquad \text{and} \qquad 1 \mapsto \boxed{\heartsuit}\boxed{\clubsuit}$$

For convenience we take the notations and some text fragments from Mizuki and Shizuya [MSh14]. A pair of face-down cards is called a commitment to x, and is written as

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

For n parties and their private inputs $x_0, \ldots, x_{n-1}$ we want securely compute a function

$$f(x_0, \ldots, x_{n-1})$$

such that the input values $x_i \in \{0, 1\}$ remain private and no trusted third party must be employed. We only consider the so-called *semi-honest* adversary model, i.e. all parties follow the protocol correctly, however, maybe they are curious and want to learn the other input values.

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

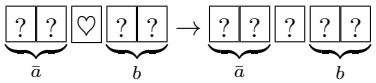
$$\underbrace{\boxed{?}\boxed{?}}_{x} \rightarrow \overbrace{\boxed{?}\boxed{?}}^{\rightleftharpoons} \rightarrow \underbrace{\boxed{?}\boxed{?}}_{\bar{x}}$$

In addition, there exists protocols (e.g. [CK93, NR98]) for copying a commitment. By using these techniques we can realize branches and complex data flows in a computation. Hence, without loss of generallity we will consider only two party protocols in the remaining sections.

den Boer's Five-card Trick

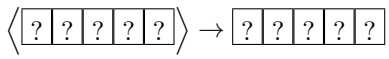
First of all, we sketch a simple protocol by den Boer [dBo89] as a starting lesson. Assume that Alice, holding a bit $a \in \{0,1\}$, and Bob, holding a bit $b \in \{0,1\}$, want to securely compute $a \wedge b$, i.e., they wish to learn only the value of $a \wedge b$. The five-card trick [3] works as 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 an additional card are placed



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

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



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

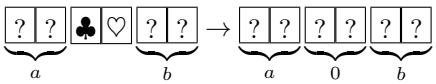
3. Reveal all of the five cards; then, we have either three (cyclically) consecutive $\boxed{\heartsuit}$ or not. The former case implies $a \wedge b = 1$ and the latter implies $a \wedge b = 0$.

Note that no output in committed form is produced by the protocol. That makes consecutive computations harder.

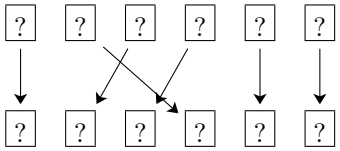
Mizuki-Sone's six-card AND protocol

Given commitments to a and b together with two additional cards, this protocol [MSo09] produces as output a commitment to $a \wedge b$, as follows.

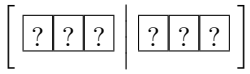
1. In addition to the two commitments of Alice and Bob, arrange a commitment to 0:



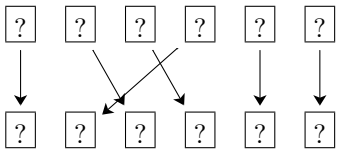
2. Rearrange the order of the sequence such that the second card is placed at the fourth position, the third card is placed at the second position, and the fourth card is placed at the third position:



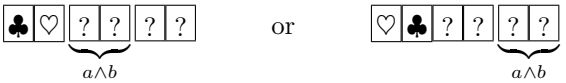
3. Bisect the sequence of six cards, and switch both parts randomly (so-called *random bisection cut* [MSo09] denoted by $[\cdot|\cdot]$):



4. Rearrange the order of the sequence such that the second card is placed at the third position, the third card is placed at the fourth position, and the fourth card is placed at the second position:



5. Reveal the first two cards from the left; then, their value tells us the position of the desired commitment to $a \wedge b$:

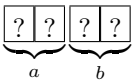


Some arguments for correctness and security of this protocol can be found in the corresponding papers [MSo09, MSh14].

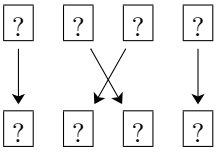
Mizuki-Sone's four-card XOR protocol

This protocol [MSo09] produces a commitment to $a \oplus b$ without any additional card, as follows.

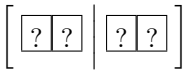
1. Arrange the commitments of Alice and Bob as



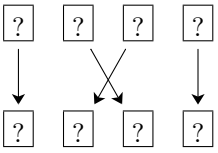
2. Rearrange the order of the sequence as



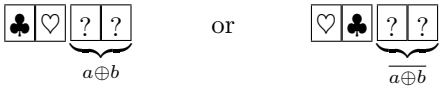
3. Apply a random bisection cut



4. Rearrange the order of the sequence again as



5. Reveal the leftmost two cards; then, their value determines the outcome:



Note that the secure NOT computation (swapping the remaining two cards) can transform a commitment to $\overline{a \oplus b}$ into one to $a \oplus b$.

References

[dBo89] Bert 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.

[CK93] Claude Crépeau and Joe Kilian. Discreet Solitary Games. Proc. CRYPTO '93, Lecture Notes in Computer Science, vol. 773, pp. 319–330. Springer-Verlag, 1994.

[NR98] Valtteri Niemi and Ari Renvall. Secure multiparty computations without computers. Theoretical Computer Science, vol. 191, pp. 173–183, 1998.

[Sti01] Anton Stiglic. Computations with a deck of cards. Theoretical Computer Science, vol. 259 (1-2), pp. 671–678, 2001.

[MSo09] Takaaki Mizuki and Hideaki Sone. Six-card secure AND and four-card secure XOR. Proc. FAW 2009, Lecture Notes in Computer Science, vol. 5598, pp. 358–369, Springer-Verlag, 2009.

[MSh14] Takaaki Mizuki and Hiroki Shizuya. Practical Card-based Cryptography. Proc. FUN 2014, Lecture Notes in Computer Science, vol. 8496, pp. 313–324, Springer-Verlag, 2014.

[1] Heiko Stamer. LibTMCG – a C++ library for creating secure and fair online card games. `http://www.nongnu.org/libtmcg/`