INFO-F-514 - Course Report
Secure computation

Université Libre de Bruxelles

Hakim Boulahya

June 9, 2018

1 Introduction

In this paper we will propose formal definitions of secure computation, also refered as secure multiparty computation. We will also gives known example in the litterature that are defined following the logical of secure computation. We will then recall state of the art techniques and protocols that allow to resolve secure computations.

2 Secure computation

A secure multi-part computation problem, is a problem where a computation, or a result, must be computed but the input that each party must used is confidential and not share between all parties. Such problem can be defined as a function $f(\cdot)$, that takes n parameters. The idea is to be able to compute the function $f(x_0, .., x_n)$ where the input x_i can only be accessed by the party i. The final results is accessible to everyone. cite f(.)

3 Problems

There exists multiple problems that used the secure computation definition. For example the millionaires problem [Yao82], is the problem that for two millionaires they both want to know which one of them is the richer, but they don't want to know the difference. In this problem, the computation function is the usual comparison $<$, and the inputs are the incomes of the individuals.

Oblivious Transfer Another problem is the Oblivious Transfer introduced by Rabin during in 1982 [Rab]. The Oblivious Transfer has many application and has been first introduced has a protocol to resolve the Exchange Of Secret problem. In the context of secure computation, The 1-out-of-2 Oblivious Transfer (OT_2^1), an another approach to the original Oblivious Transfer, is the problem that for a sender and a receiver, one of two message must be sent from the receiver to the sender. The message receive can be chosen by the receiver. Two constraints are that the sender must never know which message has been chosen, and the receiver must not know the content of the other message.

1-out-of-n Oblivious Transfer (OT_n^1) is an extension of OT_2^1, where the sender has n messages to send and the receiver must choose one of theme. Those two protocols are theorically equivalent has proven in [Cre88, Cac98].

4 Original Oblivious Transfer protocol

As mentionned in the previous section,

5 1-out-of-2 Oblivious Transfer protocol

On possible protocol for the OT_2^1 problem is by using a pair of key using the RSA protocol, first proposed in [EGL85]. Let Alice be the sender and Bob the receiver. Alice has two messages m_0, m_1, and in addition to that a public RSA key (e, d, n). The protocol is a multi-step communication between the two parties using the RSA public key of Alice.

The first step is for Alice to send the public key and two random values, that is the public key (e, n) and two random values x_0, x_1 contains in the domain $[1, n-1]$. Now that Bob has those inputs, he will generates on his side two other random values. The first one is the bit b which value is either 0 or 1, and is used to choose which random inputs received from Alice, that is x_b would be either x_0 or x_1. The second generated random value of Bob is a value k in the domain $[1, n-1]$.

The second step is for Bob to return his response. Since we don't want Alice to know which value has been chosen, Bob will encrypt the value x_b by blinding it using the random value k that he generated. That is Bob will send to Alice the value $v = (x_b + k^e) \mod n$. Upon receival of v, Alice will decrypt v two times, by removing the random values. That is, Alice will have two values k_0, k_1 where $k_i = (v - x_i)^d \mod n$.

Finally, the last step if for Alice to send back the real message. Since v was blinded by Bob with the value k, Alice doesn't know which random x_i has been chosen. By computing the two k_i based on both values, one of them will be identical to the k value of Bob. The last inputs that Alice will send to Bob are the two messages m'_0, m'_1 where $m'_i = m_i + k_i$. Upon receival, Bob will have to decrpyt the message with k, that is $m_b = m'_b - k$. Since Bob only has the k_i value associate to his message, he will not be able to decrypt the other message.

Give propoer def of RSA public key

explain rsa key pair (add a reminder section)

OT12 vs OT equivalence (check article that OT12 -¿ OT but not OT12)

why second step of Bob we have k^e ?

Proof of OT12 et OT equivalement Crepeau page 351 et Event page 640

6 Secure computation with OT_2^1

References

[Cac98] Christian Cachin. On the foundations of oblivious transfer. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, and Kaisa Nyberg, editors, *Advances in Cryptology — EUROCRYPT'98*, volume 1403, pages 361–374. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[Cre88] Claude Crepeau. Equivalence Between Two Flavours of Oblivious Transfers. In G. Goos, J. Hartmanis, D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. Seegmüller, J. Stoer, N. Wirth, and Carl Pomerance, editors, *Advances in Cryptology — CRYPTO '87*, volume 293, pages 350–354. Springer Berlin Heidelberg, Berlin, Heidelberg, 1988.

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, June 1985.

[Rab] Michael O. Rabin. How to Exchange Secrets with Oblivious Transfer.

[Yao82] Andrew C. Yao. Protocols for secure computations. pages 160–164. IEEE, November 1982.