622.755 – Introduction to Cybersecurity Summer Term, 2022/23

Project Write Up

Lorenzo Zanolin 12245822

1 Outline

The report is organized as follows: Section 2 offers an overview of Yao's protocol, including some insights of my implementation of the Secure Multiparty Computation (SMPC) adder [3]. In Section 3, I present a practical application of my implementation in real-world scenarios. Section 4 delves into an analysis of the social, ethical, and legal considerations associated with this implementation. Finally, Section 5 provides a comprehensive summary of the entire report, encapsulating its contents.

2 Yao's protocol

This project covers the Yao's protocol[6]; more precisely the *Secure Multi-Party Computation*. This protocol allows two parties, Alice who knows x and Bob who knows y, to compute jointly the value of $f(x, y)$ in a way that does not reveal to each side more information than can be deduced from $f(x, y)$[5]. In this scenario Alice is the garbler, while Bob is the evaluator. Another important role is the use of the OT[5], which is responsible to let Bob knows his encrypted input. An example of functioning is represented in figure 1.

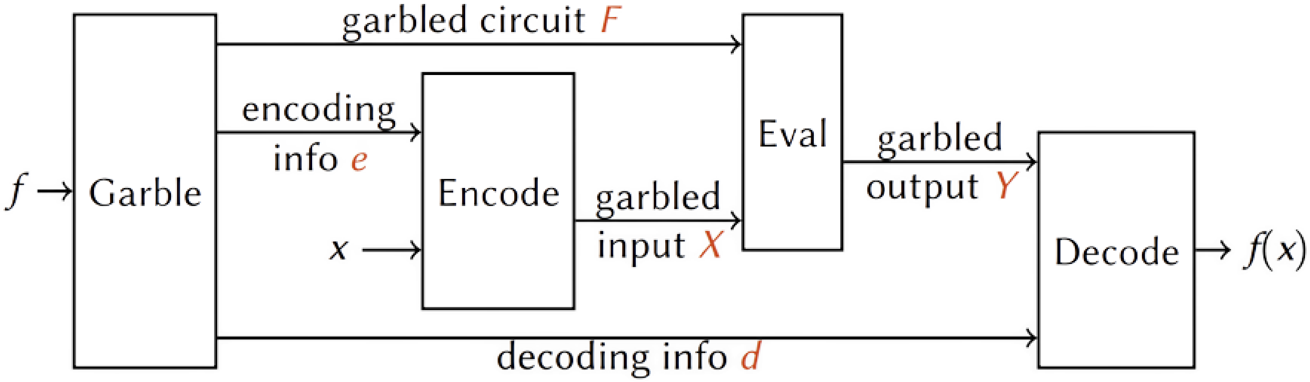

Figure 1: Steps of the SMPC

[1] There are two principles that must be respected[4]:

[1] 1 was taken from `https://web.engr.oregonstate.edu/~rosulekm/cryptabit/1-overview.pdf`
2 was taken from `https://upload.wikimedia.org/wikipedia/commons/1/14/Half-adder.svg`
3 was taken from `https://upload.wikimedia.org/wikipedia/commons/a/a9/Full-adder.svg`.

- *privacy*: nothing is learned from the protocol other than the output;
- *correctness*: the output is distributed according to the prescribed functionality.

The request was to implement a program for which two user can sum up their set of values without sharing them with the opposing party; in this case we decided to create a 8-bit adder circuit. The circuit uses 7 full adders, 1 half adder and 1 if-then-else, represented in figures 2 3 4, concatenated together; the implementation of the entire circuit is represented in figure 5. In this implementation, as you can read in my full review at `https://github.com/lorenzozanolin/garbledCircuit`, *privacy* is respected since neither Alice or Bob can understand the set of the other. The same holds for the *correctness*: assuming that both actors behave always honestly, the result obtained through Yao's protocol is guaranteed to be the same as that obtained through a standard computation, and this is verified by a procedure inside the program.

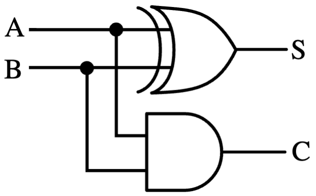

Figure 2: Half Adder

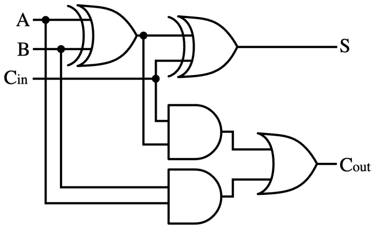

Figure 3: Full Adder

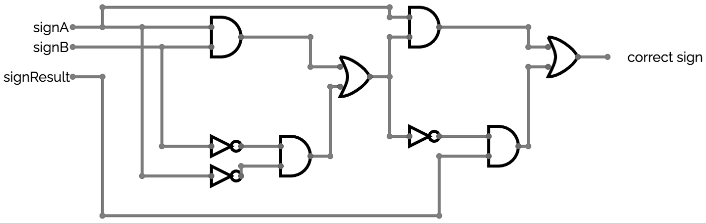

Figure 4: If then else

3 Real word application

Secure multiparty computation protocols permit to compliantly, securely, and privately compute on distributed data without ever exposing or moving it. It permits parties to collaborate without compromising sensitive information to each other or third parties, enhancing trust and minimizing the episodes of data breaches. An important field where privacy is crucial is *healthcare*. Medical institutions frequently require access to patient data from other healthcare providers to furnish better patient care; this informations

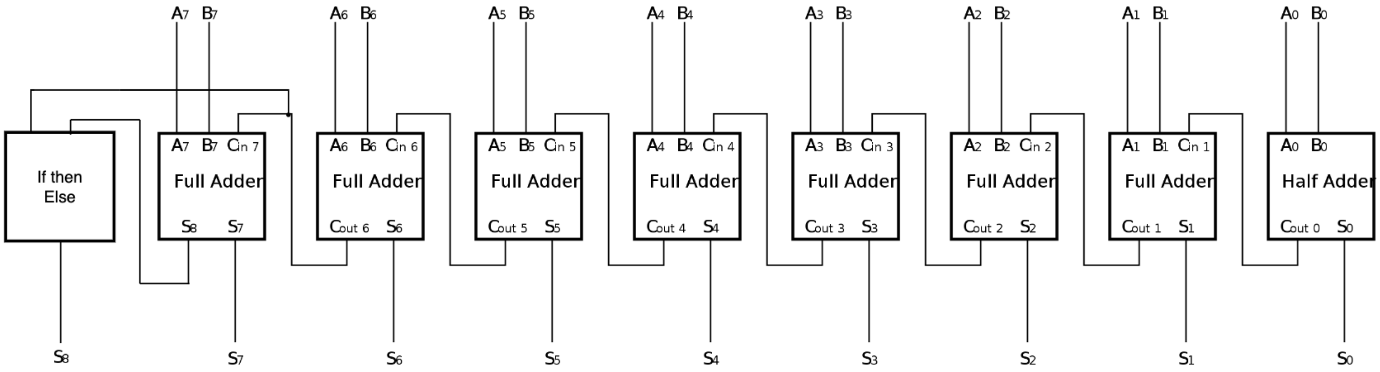

Figure 5: 8-bit Adder

must be kept private [2]. SMPC is useful because it allows to perform a joint function, such as a *statistical analysis*, on patient data while keeping it private. An example can be found in the paper [1], in which it is shown how Yao's garbled circuit is used to compute how many patients from different institutions have high Charlson index or have visited an ED for at least 4 times within a year.

3.1 Average salary calculator

In the *sum* case, an example could be the following: imagine you are an employee in a corporation and you want to know the average salary without revealing to anyone your income. In this scenario you can the use the *SMPC* protocol to get everyone's data and compute the sum; then you can proceed calculating the mean. As example, think of a big corporation with multiple locations. We have four employees, respectively Alice, Bob, Eve and Charlie; they work in two different locations and they want to compute the average salary among them. In my implementation, there are two parties (that represent each one a single location) and both of them have a pair of integers that represent the salary. Thus, once the `main.py` is called, the sum of the sets is computed; now the final operation is to divide the result by the number of participants, to obtain the average. In this example no one can understand values of other employees, thus *privacy* is respected; same holds for the *correctness* since the result obtained through Yao's protocol is guaranteed to be the same as that obtained through a standard computation.

4 Ethical, Legal and Social aspects

4.1 Ethical considerations

SMPC is useful for protecting user data and guaranteeing their anonymity, but at the same time it is important that this whole process does not lead to disadvantageous situations for some of the parties involved. Suppose we examine statistical surveys, where calculations are made to determine various metrics, such as the average salary of a particular group. It is common for such surveys to encounter unbalanced data, where certain subsets of data have significantly different properties than the others. Take as example a scenario where a company has multiple locations across the globe, where employees in less developed regions earn far less than their counterparts in more prosperous countries. In such a case, if the number of employees in the more prosperous regions is significantly greater than those in the lower-paid regions, the mean salary calculation could potentially be biased, leading to unreliable results. The problem here is intrinsic to the operation itself, SMPC is not really the guilty part; to overcome it, it should exist a control that checks whether the variation Δ of the inputs is below a predefined threshold.

4.2 Legal considerations

It's important that all the computations done must preserve confidentiality, let us analyze the 8-bit adder case. By construction of my solution, all inputs (sensible data) are encrypted using *AES-CBC* as encryption scheme; thus all data that flows is encrypted. By definition, encryption is used to assure confidentiality. To delve into further details, we can refer to the *General Data Protection Regulation* (GDPR). This regulation provides comprehensive data protection measures and imposes additional responsibilities on organisations involved in the collection and processing of such data. These guidelines ensure that appropriate security measures are implemented to protect personal data from unauthorised access, loss, or misuse. Some individuals or organisations may be hesitant to share their data, even in a secure and privacy-preserving manner, due to concerns about privacy breaches. There could be a need for clear communication and education (on how this technique works) to address these concerns and build confidence in the technology.

4.3 Social considerations

A significant social consideration when it comes to utilising secure multiparty computation (MPC) is the potential impact on trust and collaboration among participating parties. While MPC offers the advantage of secure computation without exposing sensitive inputs, it becomes crucial to address the issue of potential dishonest behavior among participants, as it could lead to unreliable results. In some cases, economic interests may come into play, where inputs are intentionally biased to manipulate the outcome; the challenge lies in identifying and addressing such situations since all data is encrypted, making it difficult to detect any manipulation or biased inputs. Finding ways to ensure the integrity of participants and establishing mechanisms for accountability become crucial aspects to consider in order to maintain the trustworthiness and reliability of MPC in practice.

5 Final considerations

In conclusion, in this paper we briefly presented the Yao's Secure computation analysing an implementation of it. We further explored the practical applications of this technology in real-world scenarios, highlighting its significant potential. However, it is essential to pay attention when utilising this technology, as emphasised in Section 4. Due to the encryption of all information used, once the computation is performed, the data cannot be reversed; therefore, careful attention must be taken during the process of selection of inputs to ensure accurate and reliable results.

References

[1] Xiao Dong, David A Randolph, Chenkai Weng, Abel N Kho, Jennie M Rogers, and Xiao Wang. Developing high performance secure multi-party computation protocols in healthcare: A case study of patient risk stratification. *AMIA Jt Summits Transl Sci Proc*, 2021:200–209, 2021.

[2] Wenliang Du and Mikhail J. Atallah. Secure multi-party computation problems and their applications: A review and open problems. NSPW '01, page 13–22, New York, NY, USA, 2001. Association for Computing Machinery.

[3] Daniel Escudero. An introduction to secret-sharing-based secure multiparty computation. Cryptology ePrint Archive, Paper 2022/062, 2022. `https://eprint.iacr.org/2022/062`.

[4] Yehuda Lindell and Benny Pinkas. A proof of security of yao's protocol for two-party computation. *Journal of Cryptology*, 22, 2009.

[5] Moni Naor and Benny Pinkas. Computationally secure oblivious transfer. *Journal of Cryptology*, 18, 2005.

[6] Andrew C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 160–164, 1982.