622.755 – Introduction to Cybersecurity Summer Term, 2022/23

Project Write Up

Lorenzo Zanolin 12245822

1 Outline

Please write a general overview explaining the structure of your report. This can be brief (one paragraph), but it should make it clear how the report is structured. The overall length of your report must not exceed 5 pages (including figures, algorithms, tables, and references). This report is structured as follows: I will provide a brief overview of the the written project in Section **??**. Then I will explain how I would suggest you use your time to achieve a good write up in Section **??**, and finally I will explain and illustrate the marking scheme in Section **??**.

2 Yao's protocol

This project covers the Yao's protocol[3]; 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 $f(x, y)$[2]. In this scenario Alice is the garbler, while Bob is the evaluator. Another important role is the use of the OT[2], which is responsible to let Bob knows his encrypted input. An example of functioning is represented in figure 1.

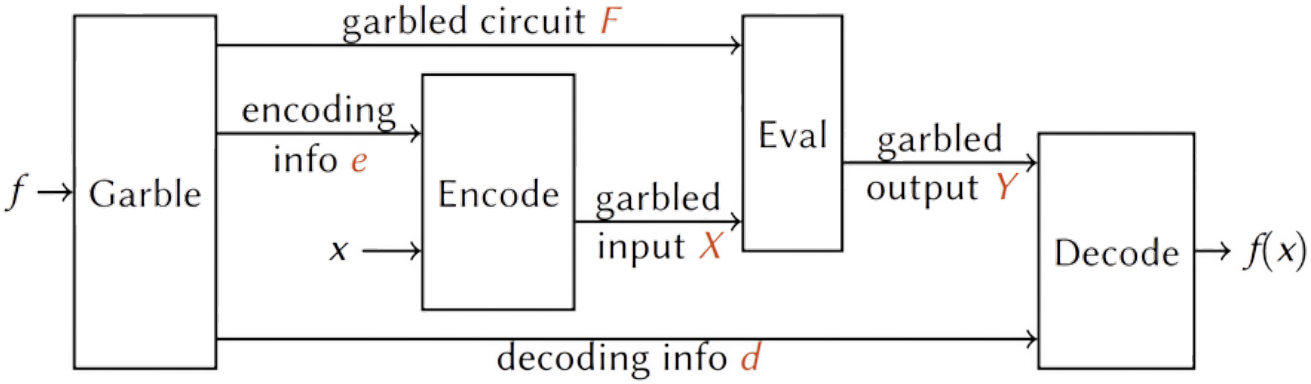

Figure 1: Steps of the SMPC

[1]

[1] 1 was taken from `https://web.engr.oregonstate.edu/~rosulekm/cryptabit/1-overview.pdf`

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

- *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.

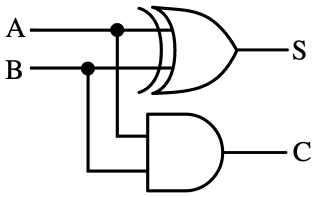

Figure 2: Half Adder

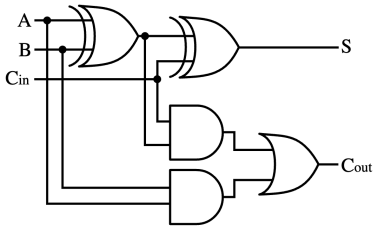

Figure 3: Full Adder

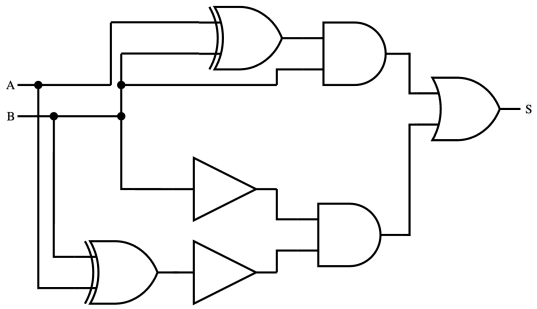

Figure 4: If then else

Image 2 was taken from `https://upload.wikimedia.org/wikipedia/commons/1/14/Half-adder.svg` Image 3 was taken from `https://upload.wikimedia.org/wikipedia/commons/a/a9/Full-adder.svg`.

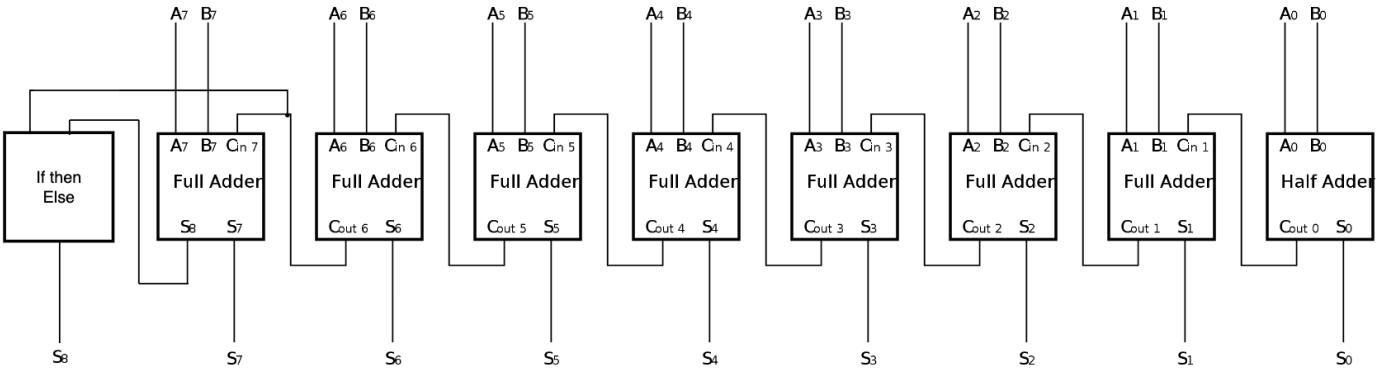

Figure 5: 8-bit Adder

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 *healtcare*. Medical institutions frequently require access to patient data from other healthcare providers to provide better patient care; this informations must be kept private. SMPC is useful because it allows to perform a joint function, such as a *statistical analysis*, on patient data while keeping it private.

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 proceede by calculating the mean. 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. For more details, we can follow the General Data Protection Regulation (GDPR) which gives enhanced data protection measures by placing increased responsibilities on organizations that collect and process such data. These guidelines ensure that appropriate security measures are in place to safeguard personal data from unauthorized access, loss, or misuse.

5 Final considerations

References

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

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

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