Secure Two Party Sum with Yao's Garbled Circuit

Mattia D'Urso

February 11, 2021

1 Introduction

This project implements a two-party secure function evaluation[5] using Yao's garbled circuit [2] protocol. This project uses a modified version of Ojroques' garbled-circuit repository[3] in order to compute a secure sum between two parties.

In this project, two parties Alice and Bob compute the sum on their sets without sharing the value of each of their inputs with the opposit party. Alice is the circuit creator (the *garbler*) while Bob is the circuit evaluator. Alice creates the Yao circuit and sends it to Bob along with her encrypted inputs. Bob then computes the results and sends them back to Alice.

The circuit used in this project is composed respectively of a half adder[4] and seven full adders[1] in series.

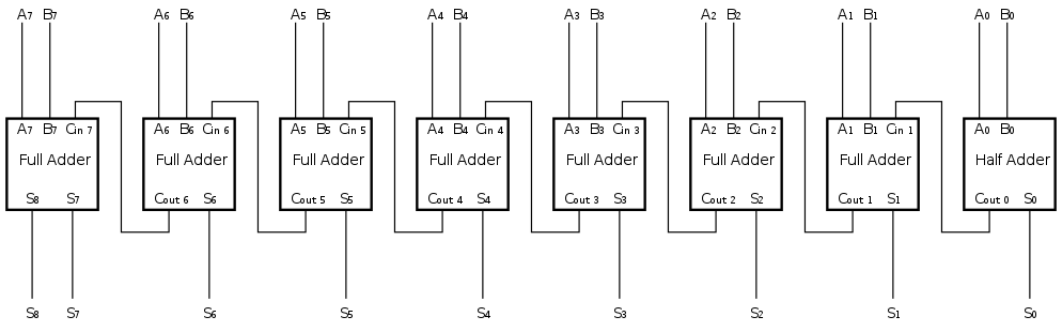

Figure 1: Representation of the circuit used

2 Yao's Protocol

The steps of Alice's algorithm are the following:

```
Alice(inputs, circuit):
    input_alice ← map_wires(input)
    circuit ← read_circuit(circuit)
    keys_alice ← rand_keys_AES()
    truth_table ← init_circuit(circuit)

    circuit_enc ← ENC_keys_a(truth_table)
    garbled_circuit ← garble(circuit_enc)

    keys_for_Bob ← select_keys_for_Bob(keys_a)
    result ← Bob(garbled_circuit, keys_for_Bob, Bob_input)

    return result
```

Algorithm 1: Alice algorithm

The steps of Bob's algorithm are the following:

```
Bob(garbled, keys_a, input):
    input_b ← map_wires(input)

    while end_circuit_is_evaluated do:
        decrypted_gate ← use_keys(keys_a)
        keys_b ← oblivion_transfer(decrypted_gate, bob_input)
    message ← decrypted_gate

    return message
```

Algorithm 2: Bob's algorithm

3 How to use

In order to correctly install or check the dependencies examine the original [3] documentation.

3.1 Input format

This project expects a sigle line of integers $\in \mathbb{N}$ separated from a space in each file *.txt*. This circuit is able to compute a secure two party sum up to 8-bit length inputs.

3.2 Circuit reppresentation

At the path *./src/input* you can write Alice and Bob's inputs respectively in the file *alice.txt* and *bob.xt*.

In order to check the truth table copy and paste this in the terminal

```
python main.py local -c circuits/add.json -m table
```

3.3 Execute the protocol

In order to run the protocol over the network you can copy and paste this in the **Bob's** terminal:

```
cd path_of_src
python main.py bob
```

and this in the **Alice's** terminal:

```
cd path_of_src
python main.py alice -c circuits/add.json
```

4 Inference

This project implements a really simple sum function as example of what this type of technology can do, it makes possible things that before were not feasible. A framework based on Yao's protocol is able to preserve the privacy of the people and at the same time to use sensitive data to compute functions or train machine learning models that are able to really help people, for example in a healthcare scenario.

References

[1] Full adder
https://it.wikipedia.org

[2] Garbled circuit
https://it.wikipedia.org

[3] github.com/ojroques/
https://github.com

[4] Half adder
https://it.wikipedia.org

[5] Secure two party computation
https://it.wikipedia.org