Course Project

1 Topic of the Project

The aim of the project is to show how simple functions can be computed privately using MPC.
The first part of the project is

- Implement a simple Yao protocol for two parties (with AES).
- You can choose from the following functions which is to be computed between two parties using the protocol
  - Sum of a set of values
  - Maximum of two sets of values
  - Minimum of two sets of values
  - Common elements in two sets of values
- The floating point to integer (and vice versa) conversion for this implementation (you do not have to implement this if you are working alone).

The second part of the project is

- Find a use case and discuss (drawing on what you have learned in SEL course) the social, ethical, and legal aspects of your use case (perhaps by considering the SEL aspects of the use case with and without the use of your implementation)

1.1 Some Details about Implementation

The implementation **must be done** in Python. Note that

- if you are working alone then your implementation should work on integers only. You can choose to implement in way that your implementation can process floating point numbers but this is not formally required.
- For the conversion part, your implementation must process decimal inputs with at least one digit before and after the decimal point. For example, 1.7, 0.4, 9.9 etc. The number of bits required to represent should be chosen in such way that it can process the floating points. Note that for

the addition function the result can have two digits before the decimal point when the inputs are having exactly one digits after decimal points. For example, the addition of 9.9 and 0.2 will be 10.1. Your script should work for such input.

- If you are working on the project alone then your implementation should be able to process integers of size *at least* 4-*bits*.
- Your implementation must be such that the inputs for Alice and Bob are given from two different terminals, and suitable outputs must be shown in both terminals. The inputs on two different terminals can be direct e.g. a number or set of comma separated numbers entered using keyboard. Alternately, your implementation can take the inputs for Alice and Bob from two different files. Your implementation should have the following outputs

  1. The message/circuit that Alice sends to Bob [this output is optional and only for your own clarity/understanding]
  2. OT between Alice and Bob [this output is optional and only for your own clarity/understanding]
  3. A function which verifies and returns 0/1 after successful or unsuccessful verification of the results (function outputs) obtained using Yao's protocol and in a normal way [this is mandatory]

Your implementation should (at least) work locally. This means that your implementation will not be checked (for correctness) on two different machine on a network but *only on a single computer*.

2 Use of Library

As mentioned earlier, you can use the implementation from Github repository garbled-circuit either as a library or as a guide to have your own implementation. The library is well documented and there are clear instructions on installation of dependencies, how to run the script with different options etc. It also clearly states the functionalities of each class and member functions.

For example the main function is called with the parameters

- `party` (i.e. name of the party Alice or Bob or local),
- `circuit` (i.e. path to the circuit file),
- `no_oblivious_transfer` (i.e. whether the oblivious transfer option is True or False),
- `m` (i.e. print mode is either in the form of a table or circuit).

The library uses json to describe the circuit. You can use the same. If you choose to describe circuits differently then you will have to write the script to read the circuit accordingly.

3 Documentation and Submission

You are required to submit a document together with your script. It should contain the following

- The version of Python used
- The choice of parameters and function
- Description of the circuit of the function
- The functionalities of your implementation with the name of the functions
- Clear instructions on how to run your script (including any dependencies other than those necessary for the github script) and how to interpret the outputs
- If you are sharing the script using github then a link to the repository
- If you are working in a group of two then the name of the other group member

The document must be in pdf format and should be named as firstname_lastname.pdf. If you choose to submit your code together with the pdf file then compress them together and follow the same nomenclature for the compressed file.

4 Grade Points

Out of 40% of the project's grade 15% is accounted for programming part. On a scale of 15 the evaluation of the programming will be as following for the programming of MPC

- For describing the circuit of the function 1 point
- For writing the pseudocode of Alice's communication to Bob 2 points
- For writing the pseudocode of Bob's communication to Alice 2 points
- For writing the working (python) script of the MPC of the chosen function 6 points
- For writing the documentation (describing the functionalities of each process in your script, how to call them, how to run the script, install any additional package) 3 points (in case of a team 2 points to a member)
- For proper scripting/coding style (writing readable comments in your script e.g as it is done in the github library script) 1 point

The marking scheme will be as following for the programming part of the floating point to integer (and vice versa)

- For writing the pseudocode to of the conversion from floating point to integer 2.5 points
- For writing the pseudocode to of the conversion from integer to floating point 2.5 points
- For writing the working (python) script for the above two functionalities 6 points
- For writing the documentation (describing the functionalities of each process in your process, how to call them, install any additional packages) 1 point
- For proper scripting/coding style (writing readable comments in your script e.g as it is done in the github library script) 1 point

Note that 1 point will be given to each individual in a team for the documentation of the overall script (e.g how to run the script, install any additional package necessary, functionalities of the common processes in the integrated script).

The pseudocode and circuit should be in the document accompanying the script.

Note that

- Submitting an implementation using any other language will result in no grade points corresponding to your implementation.
- If the instructions for executing your implementation is not clear enough then we will contact you (via moodle or email). If you fail to communicate regarding this and we can not verify you solution (by running the program) then you will receive no grade points for the corresponding part.
- If your program does not allow to give inputs from two terminals (for Alice and Bob) as specified in section 1.1 then you will not receive any grade points for the implementation part.

5 Project Rules

1. If you use text verbatim from a source, or if you take a picture, table or the like, you must use quotation marks and make it clear via a citation (including page number) what the original source is.
2. If you use any code (fragment) from existing source code available publicly then you must mention it in your report and put a reference to the source.
3. If we find copies of any kind (be these parts of source code or the written report) all students receive zero (these are the AAU rules).