Multi Party Computation - Active Adversaries

Computer Science Lab
July 7, 2015

Vitor Enes, José Bacelar Almeida, and Bernardo Portela

University of Minho, Portugal

Abstract. This project aimed at implementing a secure MPC protocol against active adversaries.
We implemented a protocol called SPDZ that belongs to the family of protocols in the preprocessing model. This protocol pushes all the expensive public-key machinery to the preprocessing phase. Then, in the online phase, it only uses cheap primitives, which gives us extremely efficient computations. In addition, this phase is actively secure against a dishonest majority of corrupted players.
This work is focused on the online phase.

Keywords: multi party computation, SPDZ

1 Introduction

Although secure multiparty computation was invented almost thirty years ago, only in the recent years these protocols were implemented and tested in practice. They can be divided in two camps essentially: based on Yao circuits or based on secret sharing. The ones based on Yao circuits are mainly focused on two party computations, but the ones based on secret sharing can be applied to a more general number of players.

The protocols based on secret sharing can be divided on those which consider only honest-but-curious adversaries and those which consider active adversaries. Often these protocols are presented in the preprocessing model in which is possible to produce random data that will be consumed during the online phase. SPDZ fits this model and provides full active security against a dishonest majority.

The goal of this project is to implement the online phase of SPDZ assuming a trusted dealer. This dealer provides the preprocessed data that would be generated in the offline phase.

Report structure. Bearing this in mind, this report will fistly address the theory behind this work in section 2; It will then in 3 explain the system's architecture and implementation choices; And, in section 4 a final appreciation of the work will be made, together with some suggestions for improvement.

2 SPDZ

As stated before, SPDZ is divided in two phases: offline and online. One key aspect of the offline phase is that it can occur without knowing neither the function to be computed

nor the inputs. This is good because both may not be known until the online phases starts. This asynchrony allows us to have very efficient and secure computations, as long as there is preprocessed data available.

The protocol also supports full reactive computations: after one function is evaluated, another can be executed depending on the output of the first. This can go forever until the preprocessed data runs out.

3 Implementation

4 Conclusions and Future Work

References