Chapter 11
Secure Multiparty Computation via Oblivious Polynomial Evaluation

Mert Özarar
Middle East Technical University, Turkey

Attila Özgit
Middle East Technical University, Turkey

ABSTRACT

The number of opportunities for cooperative computation has exponentially been increasing with growing interaction via Internet technologies. These computations could occur between almost trusted partners, between partially trusted partners, or even between competitors. Most of the time, the communicating parties may not want to disclose their private data to the other principal while taking the advantage of collaboration, hence concentrating on the results rather than private data values. For performing such computations, one party must know inputs from all the participants; however, if none of the parties can be trusted enough to know all the inputs, privacy will become a primary concern. Hence, the techniques for Secure Multiparty Computation (SMC) are quite relevant and practical to overcome such kind of privacy gaps. The subject of SMC has evolved from earlier solutions of combinational logic circuits to the recent proposals of anonymity-enabled computation. In this chapter, the authors put together the significant research that has been carried out on SMC. They demonstrate the concept by concentrating on a specific technique called Oblivious Polynomial Evaluation (OPE) together with concrete examples. The authors put critical issues and challenges and the level of adaptation achieved before the researchers. They also provide some future research proposals based on the literature survey.

DOI: 10.4018/978-1-4666-4030-6.ch011

INTRODUCTION

The number of opportunities for cooperative computation has exponentially been increasing with growing interaction via Internet technologies. These computations could occur between trusted partners, between partially trusted partners, or even between competitors. Most of the time, the communicating parties may not want to disclose their private data to the other principal while taking the advantage of collaboration, hence concentrating on the results rather than private and perhaps useless data values. For example, two or more competing large organizations might jointly invest in a project that must satisfy all organizations' goals while preserving their private and valuable data (Du, 2001). For performing such computations, one party must know inputs from all the participants; however if none of the parties can be trusted enough to know all the inputs, privacy will become a primary concern. Hence the techniques for secure multiparty computation are quite relevant and practical to overcome the privacy gaps.

Secure Multiparty Computation

If multiple parties want to perform a computation based on their private inputs, but neither party is willing to disclose its own input to anybody else, then the basic problem is how to conduct such a computation while preserving the privacy of the inputs. This is referred to as Secure Multiparty Computation problem (SMC) in the literature.

For example, consider the following real life scenarios where SMC can directly be applicable;

1. Some hospitals situated in various different countries having their medical databases and patient's history stored on some remote database sites. They want to wish to jointly mine their patient's data for the purpose of medical research and prevention of data is to be maintained due to confidentiality of patients' records.
2. In a given exam, the results are privately shared with the students. No student wants to disclose its exam grade yet all the students want to calculate the average of the exam.
3. Let us assume that an airline company that has a reservation database for each country exists. If a person wishes to make a reservation from city A located in country X to a city B located in country Y, then we need to consult each intermediate countries databases. These databases provide only the queried details without disclosing their whole reservation database.

In general, a secure multiparty computation problem deals with computing any function on any input, in a distributed network where each participant holds one of the inputs, ensuring independence of the inputs, correctness of the computation, and that no more information is revealed to a participant in the computation other than that can be inferred from the participant's input and output (Du, 2002).

Currently, to solve the above problems, a common strategy is to assume the trustworthiness of the service providers, or to assume the existence of a trusted third party, which is risky in nowadays' dynamic and malicious environment. Consider a trusted party who collects all participants' data and then performs the computation and sends the results to the participants. Without having a trusted party, some communication among the participants is certainly required for any related computation; yet we do not know how to ensure that this communication does not disclose anything. Therefore, protocols that can support joint computations while protecting the participants' privacy are of growing importance.

In theory, the general secure multiparty computation problem is solvable (Yao, 1986; Goldreich, 1987; Goldreich, 2004) but using the solutions derived by these general results for special cases of multiparty computation can be impractical; special solutions should be developed