Practical Secure Function Evaluation

Thomas Schneider*

thomas.schneider@informatik.stud.uni-erlangen.de

Abstract: This thesis focuses on the practical aspects of general two-party Secure Function Evaluation (SFE). A new SFE protocol that allows free evaluation of XOR gates and is provably secure against semi-honest adversaries in the random oracle model is given. Furthermore, the extension of SFE to private functions (PF-SFE) using universal circuits is considered. Based on a new practical universal circuit construction, FairplayPF is implemented that extends Fairplay, a well known SFE system, with PF-SFE.

Introduction

Consider the following situation. Several parties, each of which has a private input, wish to evaluate a function on their inputs. This need arises often indeed – almost any transaction or a communication over a network can be cast as evaluation of a function on the participant's inputs. For example, in an online auction, the bidders and the auctioneer are players who wish to evaluate the auction function, whose value is equal to the ID of the highest bidder. Other natural and important examples include financial transactions, voting, distributed database mining, etc.

A lot of the time, parties' inputs need to be hidden from the rest of the world. For example, in the case of the auction, unsuccessful bidders would want to preserve the privacy of their bids. Depending on the function, some information about the parties' inputs might be easily derived from the output of the function. For example, the winning bid of an auction might necessarily be revealed. The goal of secure computation is to ensure that no other information is leaked during the computation. Clearly, efficient methods of secure evaluation of functions are of great interest. The problem is often referred to as Secure Function Evaluation, or SFE.

Current Approaches and State of Knowledge

We first note that the general problem of SFE is a well-researched problem [LP, Yao82, Yao86]. That is, there exist well-defined protocols that allow secure computation of any

*This Diplomarbeit (masters thesis) is supervised by Prof. Volker Strehl (Department of Computer Science, Universität Erlangen-Nürnberg, Germany) and Vladimir Kolesnikov, PhD (Bell Labs, Murray Hill, NJ, USA).

function, based on certain reasonable physical or complexity assumptions.

A natural and efficient method is the so-called *Yao's garbled circuit* approach (see [LP] for an excellent presentation of the subject). There, the evaluated function is viewed as a binary circuit. During the evaluation, the signals on all wires of the circuit, except for the output wire, are garbled, and thus the evaluator (one of the participants of the computation) is limited to obtaining only the output of the computation.

This known general solution is often too inefficient to be applied in practice. Nevertheless, for a large class of useful functions (especially those with efficient circuit representation), this offers the most efficient solutions. Such problems include number comparison, auctions, evaluation of conditional statements, etc. [NPS99].

Continuing advances in available computational power and communication have made secure computation of many useful functions affordable. Several recent works approach the problem of general SFE from the practical angle, discuss and fine-tune the implementation details [KJGB06, MNPS04].

Contributions of this thesis[1]

This thesis focuses on the practical aspects of general two-party SFE. The currently best known approach for general SFE of Fairplay [MNPS04] is combined with the information-theoretically secure approach of Gate Evaluation Secret Sharing (GESS) [Kol05] to a new, practical method for general SFE. This new method results in substantial performance improvements of 50% for many important circuit structures like addition or number comparison. A proof of security in the semi-honest model is given that is based on the same assumptions as Fairplay, namely the existence of random oracles (RO).

In practice, there is often a need to not only protect the inputs, but the function being evaluated as well. One example is checking a passenger against the no-fly list (or, more generally, no-fly function of passenger's data). Here, a compromise of the function weakens the security of the system significantly. Other examples include credit checking or background- and medical history checking functions.

This well-known problem is called SFE of private functions (PF-SFE) and addressed by a large amount of work like [FAZ05, CCKM00, SYY99, Pin02]. In PF-SFE, the evaluated function is known only by one party and needs to be kept secret (i.e. everything besides the size, the number of inputs and the number of outputs is hidden from the other party). Full or even partial revelation of these functions opens vulnerabilities in the corresponding process, exploitable by dishonest participants (e.g. credit applicants), and should be prevented.

The problem of PF-SFE can be reduced to the "regular" SFE by parties evaluating a *Universal Circuit* (UC) instead of a circuit defining the evaluated function [SYY99, Pin02].

[1] Parts of this thesis will be published by Vladimir Kolesnikov and Thomas Schneider on two international conferences: "A Practical Universal Circuit Construction and Secure Evaluation of Private Functions", Financial Cryptography and Data Security (FC08) and presumably ICALP08.

UC can be thought of as a "program execution circuit", capable of simulating any circuit C of certain size, given the description of C as input. Therefore, disclosing the UC does not reveal anything about C, except its size. At the same time, the SFE computes output correctly and C remains private, since the player holding C simply treats description of C as additional (private) input to SFE. This reduction is the most common (and often the most efficient) way of securely evaluating private functions [SYY99, Pin02].

We improve previous PF-SFE constructions by giving a new simple and efficient UC construction. Our universal circuit for simulating k gates has size $\sim 1.5k \log^2 k$ and depth $\sim k \log k$. It is up to 50% smaller than the best UC (of Valiant [Val76], of size $\sim 19k \log k$) for practical circuit sizes of up to ≈ 5000 gates. This improvement results in corresponding performance improvement of SFE of (small) private functions. Since, due to cost, only small circuits (i.e. < 5000 gates) are practical for PF-SFE, our construction appears to be the best fit for many practical PF-SFE.

General PF-SFE is implemented based on this UC construction and the Fairplay SFE system [KS].

When using the improved SFE protocol of this thesis to evaluate a universal circuit, PF-SFE can be improved to approximately 25% of the previously best known solution using Fairplay as underlying SFE protocol.

The results of this thesis substantially improve general SFE for many practical circuits and demonstrate practicability of general PF-SFE for "small" functions.

References

[CCKM00] Christian Cachin, Jan Camenisch, Joe Kilian, and Joy Müller. One-Round Secure Computation and Secure Autonomous Mobile Agents. In *ICALP '00*, pages 512–523, London, UK, 2000. Springer-Verlag.

[FAZ05] Keith Frikken, Mikhail Atallah, and Chen Zhang. Privacy-preserving credit checking. In *EC '05: Proceedings of the 6th ACM conference on Electronic commerce*, pages 147–154, New York, USA, 2005. ACM Press.

[KJGB06] Louis Kruger, Somesh Jha, Eu-Jin Goh, and Dan Boneh. Secure function evaluation with ordered binary decision diagrams. In *CCS*, pages 410–420. ACM Press, 2006.

[Kol05] Vladimir Kolesnikov. Gate Evaluation Secret Sharing and Secure One-Round Two-Party Computation. In *Advances in Cryptology – ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 136–155. Springer, 2005.

[KS] Vladimir Kolesnikov and Thomas Schneider. FairplayPF. `http://thomaschneider.de/FairplayPF`.

[LP] Yehuda Lindell and Benny Pinkas. A proof of Yao's protocol for secure two-party computation. Cryptology ePrint Archive, Report 2004/175.

[MNPS04] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. Fairplay — a secure two-party computation system. In *USENIX*, 2004.

[NPS99] Moni Naor, Benny Pinkas, and Reuben Sumner. Privacy Preserving Auctions and Mechanism Design. In *EC'99*, pages 129–139, 1999.

[Pin02] Benny Pinkas. Cryptographic techniques for privacy-preserving data mining. *SIGKDD Explor. Newsl.*, 4(2):12–19, 2002.

[SYY99] Tomas Sander, Adam Young, and Moti Yung. Non-Interactive CryptoComputing for NC^1. In *Proc. 40th IEEE Symp. on Foundations of Comp. Science*, pages 554–566, New York, 1999. IEEE.

[Val76] Leslie G. Valiant. Universal circuits (Preliminary Report). In *Proc. 8th ACM Symp. on Theory of Computing*, pages 196–203, NY, USA, 1976.

[Yao82] Andrew C. Yao. Protocols for Secure Computations. In *Proc. 23rd IEEE Symp. on Foundations of Comp. Science*, pages 160–164, Chicago, 1982.

[Yao86] Andrew C. Yao. How to Generate and Exchange Secrets. In *Proc. 27th IEEE Symp. on Foundat. of CS*, pages 162–167, Toronto, 1986.