Modelling and Proving Security for a Secure MPC Protocol for Stable Matching

Bachelor's Thesis Proposal
by Ella Vahle

Arbeitsgruppe
Codes und Kryptographie

1 Introduction

Secure multiparty-computation (MPC) is a concept for securely computing (potentially probabilistic) functions with multiple parties involved. Each party has private input which must not be disclosed to the other parties. Based on these, the different parties jointly execute a protocol to obtain the function's result. In this context security typically refers to the privacy of parties' inputs and correctness. To achieve privacy the protocol must ensure that no information about one party's input is leaked to any other parties beyond the function's result. Correctness is attained when the computed result equals the functions output based on the given inputs. For these considerations it is important to determine the attacker model, i.e. what the attacker is capable of. Honest-but-curious (semi-honest) attackers are passive attackers and don't acitvely intefere with the protocol. They follow instructions correctly and act as expected but try to learn more from the execution than they are supposed to. This could for example be learning about one party's preferences based on the order they are accessed in. Malicious attackers on the other hand actively interfere with the protocol and can act differently than required. It is easy to see that active attackers are much more powerful, making active security a much stronger requirement than semi-honest security.

Since there are many areas of application where MPC is useful, general purpose frameworks were developed early on. With these, arbitrary functions can be converted into a format which is applicable to MPC protocols. These are then executed on for example boolean or arithmetic curcuits . Zitat
Even though the idea of these general purpose frameworks has been around for many decades already and it has been proven that they fulfill security requirements they are not used as much as one would expect. The main reason for this is that for many Zitat interesting problems general purpose frameworks create solutions that are not efficient enough to be used in practice. For these problems special-purpose algorithms with faster Zitat execution time can be constructed. This allows them to be used in practice but involves much more work because security needs to be proven for every single solution seperately.

One such problem is finding a stable matching. Matching algorithms are used to match members of two different sets to each other with regards to their respective preferences, i.e. a (potentially partial) order over the members of the other set. A stable matching is achieved when there are no two members (one from each set) that aren't matched to each other, such that they would rather be matched to each other than to their assigned matches.
Such algorithms are used to for example match residents to residency programs or students to public schools. Obviously the best solution would be that every member is assigned to their first preference, but in reality this is rarely possible. Stable matchings are a desirable solution in such cases to still achive a high level of member satisfaction.
One solution for computing a stable matching is the Gale-Shapley algorithm, developed by David Gale and Lloyd Shapley, which will also be the one we are interested in.
The algorithm could be run by a trusted third party. This way, both parties' inputs stay hidden to the other party and are only revealed to the commonly trusted party. Relying on a commonly trusted party has several disadvantages. In many cases it is not even possible to find such a party. But even if it is possible it can be hard to find one and entail a lot of efforts.
To neither sacrifice efficency nor rely on a third party, Doerner et al. introduce a new oblivious data strucure called "Oblivious Linked Multi-List" and make use of it during the execution of the Gale-Shapley algorithm. The data structure uses data arrays combined with a method by Zahur et al. to obliviously permute them. This provides the means to obliviously access data, thus hiding access patterns and the derivable information. The authors claim that using the Gale-Shapley algorithm combined with the Oblivious Linked Multi-List ensures security in the honest-but-curious attacker model while maintaining a relatively high level of efficiency compared to other solutions. Even though the authors give a rough intuition provided by arguing as to why security holds in this setting, it has not been formally proven yet.

2 Current state of research

As mentioned above, neither using a general purose framework nor outsourcing the computation to a third party is a desirable solution. To avoid this, even prior to Doerner et al. specific-purose protocols were developed, which compute a stable matching and are carried out between the involved parties only.
Stable matching in the context of multiparty-computation comprises multiple different aspects, such as data access strategies or the underlying algorithm making it an interesting problem both in theory and practice. Both of these are research topics of their own with several possible approaches and solutions. Naturally this yields various possibilities of combining them to solve the problem of securely computing a stable matching.
As opposed to the discussed approach by Doerner et al. which uses the Gale-Shapley algorithm, Blanton et al. for example use Breadth First Search.

As for the Gale-Shapley algorithm there are several existing approaches.
The usability problem of generic solutions for stable matching, in particular the Gale-Shapley algorithm, was also mentioned by Golle . Due to this, he proposed a "practical" solution with complexitiy of $O(n^5)$ asymmetric cryptographic operations, which however has not been implemented. It is based on matching authorities, and ensured to fulfill privacy and correctness requirements when semi-honesty can be guaranteed for a majority of performing matching authorities.

Zitat

Problems with Golle's solution were later found by Franklin et al. , resulting in two new solutions. One is based on Golle's approach one is based on garbled curcuits and a protocol by Naor Nissim for secure function evaluation, both with $O(n^4)$ public key operations and computation complexity respectively. The latter one has never been implemented either.

Zitat

Previously to Doener et al. the best approach was proposed by Zahur et al.. With a runtime of roughly 33 hours on input of 512 x 512 participants, it is over 40 times slower than the Doerner et al.'s protocol.

Zitat

Zitat

3 Goals of the thesis

As already mentioned, the solution proposed by Doerner et al. is claimed to be secure but it's security has not been formally proven yet. The goal of this thesis is to formalize the rough intuition given by Doerner et al. and provide a formal proof based on the Simulation Technique introduced by Goldreich . The Simulation Technique was later explained by Lindell in an extensive Tutorial which will also be used as a basis for this thesis.

Zitat

Zitat, richtig auf-schreiben

Zitat

First we will consider the oblivious linked multi-list. According to the simulation technique I will model the notion of an oblivious linked multi-list as an ideal functionality, i.e. a formal model of the oblivious linked multi-list's expected behaviour. Then I will examine how Doerner et al. realize this functionality in detail. The questions answered will include: How do they transform the input? Which operations are performed on the data? I will then proceed to formally prove security including correctness and privacy. This will be done by showing that it works correctly, meaning that their output equals the ideal model's output and showing that nothing (beyond the funcion's output) can be learned about the input, respectively. The proof for privacy is based on the simulation paradigm, which states that if nothing can be learned about the private inputs in the ideal model and the real and ideal world's executions can not be distinguished, nothing can be learned in the real world either..

Zitat

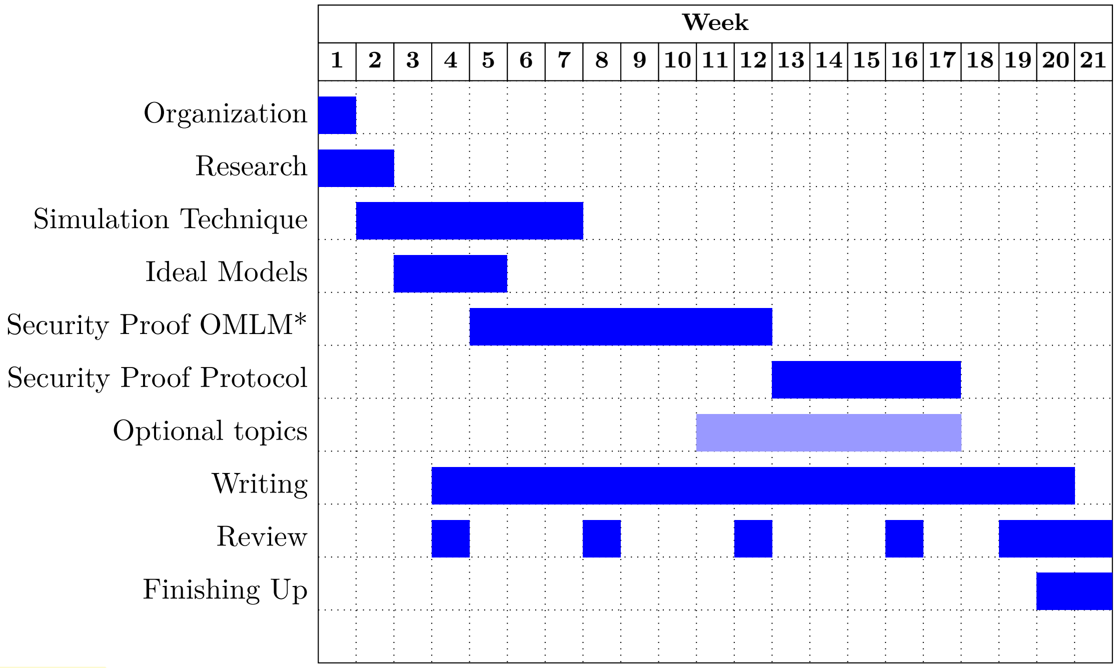
After proving security of their proposed oblivious linked multi-list implementation I will use it as a blackbox when showing the whole protocol's security. To prove the protocol's security I will carry out the same steps as I did with the oblivious linked multi-list, i.e. establishing an ideal model of a stable matching functionality, examining Doerner et al.'s construction and then formally proving security. By showing that the protocol is secure when using the oblivious linked multi-list (which was previously proven to be secure) as

an ideal blackbox I can conclude that the combination of the algorithm and oblivious linked multi-lists is also secure, using the famous composition theorem by Goldreich.
Another advantage that this modular approach has, becomes relevant when the protocol turns out not to be secure in the specified setting. Splitting the protocol into two primitives allows me to point out where things go wrong more specifically and adjust the attacker model to potentially show security in a weaker security model.
It is also possible that the proof turns out to be more straight forward than expected. A reasonable course of action could be to go a step further and investigate how the proven security transfers to other security models, such as an active attacker scenario. Here again I would be making use of the modular approach, benefiting from it's advantages.

4 Preliminary outline of the thesis

1. Introduction
2. Definitions and notation
    a) General
    b) Secure MPC
        i. Simulation Technique
3. Prior knowledge
    a) ORAM
    b) Gale-Shapley
4. Oblivious Linked Multi-lists
    a) Ideal model
    b) Construction
    c) Security
    d) (optional) Security under different security model (i.e. stronger or weaker depending on the prior results)
5. Applying Oblivious Linked Multi-lists to Gale-Shapley
    a) Ideal model
    b) Construction
    c) Security
    d) (optional) Security under different security model (i.e. stronger or weaker)
6. Conclusion
7. References

5 Work plan



*OMLM = Oblivious Linked Multi-List

Date:

Prof. Dr. Johannes Blömer

Ella Vahle

References