UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Titel

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 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. Correctness is attained when the computed result equals the functions output based on the given input.

Since there are many areas of application where MPC would be 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 [TODO Zitat].
Even though the idea of these 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. [TODO Zitat] The main reason for this is that for many interesting problems they are not efficient enough to be used in practice when solved with general purpose frameworks. [TODO Zitat] For these problems special-purpose algorithms with faster execution time can be constructed. This allows them to be used in practice but involves much more work because there is no single solution that fits all as it is the case with general purpose frameworks.

One such problem is finding a stable matchhing. Matching algorithms are used to match members of two different sets to each other with regards to their respective preferences. 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 [TODO Zitat]. 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 and in some sense fairness.
One solution for computing a stable matching is the Gale-Shapley algorithm, developed by David Gale and Lloyd Shapley [TODO Zitat], which will also be the one we are interested in.
To provide a solution which is more applicable in reality without sacrificing security, Doerner et al. [TODO Zitat] 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. [TODO Zitat] to obliviously permutate 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. Honest-but-curious attackers 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. Even though the authors give a rough intuition provided in the form of explaining arguments as to why security holds in this setting, it has not been formally proven yet.

2 Current state of research

As mentioned above there are general purpose frameworks which are applicable to arbitrary functions but are not efficient enough to be used in practice. Over the years technical advances allowed for more feasible implementations leading to the development of various products, including tools like EMP-toolkit, Obliv-C or ObliVM. Even though there were huge improvements there are still limitations. In addition to efficiency problems Hastings et al. [TODO Zitat] for example point out lack of documentation and correctness erorrs [TODO Zitat].
One alternative way to securely compute a stable matching without expensive adjustments to classic protocols is to outsource the computation to 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 avoid this, even prior to Doerner et al. [TODO Zitat] specific-purose protocols to compute a stable matching that are carried out between the involved parties only were developed. Stable matching in the context of multiparty-computation comprises multiple different aspects, such as data access strategies or the underlying algorithm. 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. [TODO Zitat] which uses the Gale-Shapley algorithm, Blanton et al. [TODO

Zitat] for example use Breadth First Search.
Some more examples of work prior to Doerner et al. [TODO Zitat] are the protocols of Golle [TODO Zitat] with roughly O(n^5) public-key operations [TODO formatieren] or the previously best implementation of the Gale-Shapley algorithm with a runtime of over 33 hours, working on sets of only 512 members each [TODO Zitat].

3 Goals of the thesis

As I 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. [TODO Zitat] and provide a formal proof based on the Simulation Technique introduced by Goldreich [TODO richtig aufschreiben/Zitat]. It was later explained by Lindell in an extensive Tutorial [TODO Zitat] which will also be used as a basis for this thesis.
First I will take a closer look at the oblivious linked multi-list. According to the simulation technique I will establish an ideal model of it's functionality. The model includes a formal description of what the functionality takes as input and what the expected output is. 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 proof 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 can be learned about the input, respectively.
After proving security for the oblivious linked multi-list 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, 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) I can conclude that the combination of the algorithm and oblivious linked multi-lists is also secure.
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) MPC
   c) 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

	Week																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Organization	■																				
Research	■	■																			
Simulation Technique		■	■	■	■	■	■														
Ideal Models			■	■	■																
Security Proof					■	■	■	■	■	■	■	■	■	■	■	■	■				
Optional topics												▨	▨	▨	▨	▨	▨				
Writing				■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
Review				■				■				■				■			■	■	■
Finishing Up																				■	■

Date:

Prof. Dr. Johannes Blömer

Ella Vahle