Statement of Purpose

Wei Qi

December 20, 2022

I am interested in theoretical computer science in general and foundations of cryptography in particular. So far, my research has been devoted to understanding basic problems in foundations of public key encryption; in particular, I have studied how we can construct new primitives with useful features such as not having the key-escrow problem in identity based encryption (IBE). In particular, I enjoy proving tight positive and negative results that match to reveal what exactly we can do. It is the desire to solve a problem *thoroughly* that motivates me to do theoretical research. During my master's study, I have done research in public key cryptography and in particular registration based encryption (RBE)(more details below). The results of this project were accepted to TCC 2022, which I presented in Chicago a month ago. In the paper, my advisor and I proved a lower bound for the number of decryption updates in RBE. Additionally, I actually do enjoy implementing carefully designed algorithms very much. It is very satisfying for me to see an algorithm work as predicted by theory. During my master's program, I also did research outside theoretical computer science and that research was also fruitful. I studied microarchitecture level optimization and have one paper accepted to MICRO 2022, which is a top venue in microarchitecture. In the paper, we designed and implemented a novel microarchitectural optimization technique called speculative code compaction, which aggressively and speculatively eliminates dead code from hot code regions resident in the micro-op cache. However, despite of this, my main research interests still lie in theoretical computer science and cryptography and that is where I want to concentrate during my PhD study.

Now I will explain more details about the main project of my master's degree about registration based encryption. RBE was proposed to resolve the key-escrow problem of IBE [Garg et al., 2018]. In IBE, an authority has a master secret key that generates all users' decryption keys. Therefore, this authority can trivially attack any user by generating the corresponding decryption key. In RBE, every user will generate its own pair of public key and secret key and register its identity and public key to a public-coin third party. This avoids the key-escrow problem of IBE [Shamir, 1985, Boneh and Franklin, 2001]. However, from time to time, users of RBE need to get a decryption update from the third party and all known constructions require $\Omega(\log n)$ updates where n is the number of users. The need of decryption update is a major disadvantage of RBE and I was curious to understand if $\Omega(\log n)$ updates are really necessary. To solve this problem, we simplified it by assuming there is no update. It is easy to see that in that case there is a simple information theoretical argument that allows one to attack simply because a compact public parameter cannot remember enough information about all the keys. We then worked on a slightly more complicated version of the problem by assuming every user receives exactly one update immediately after registration. Using a slightly more advanced information theoretical argument (i.e., the chain rule for mutual information), we were still able to get an attack. Then, we were stuck when trying to solve the more general case where there is no constraint on when updates are issued. Inspired by known construction of RBE, where users are grouped into separate trees, I tried to create independence between different users by dividing them into blocks. After I told my idea to my advisor, we together came up with a more abstract description of the problem where we used directed acyclic graphs (DAGs) to model the times of updates for a user. Then, I contributed by identifying and proving the existence of an important combinatorial structure in such DAGs, which we later called "skipping sequences". Combining this combinatorial structure with the information theoretical tool, we were able to prove that $\Omega(\frac{\log n}{\log \log n})$ updates are needed. I was thrilled when I discovered the structure because I felt I finally understood the core of the problem. However, the problem is still not fully solved since we need to assume that the update times of a user are independent of the value of its public key and I am still working on how to get rid of this assumption. This property is, however, present in all known RBE constructions. Also, after this work, an important open question is whether this result is tight. Namely, can we actually construct RBE schemes whose number of decryption updates matches the lower bound we discovered? In an ongoing work with my advisor, we discovered some surprising positive results for RBE that achieve the tight bound of $\frac{\log n}{\log \log n}$. This is done by expanding the combinatorial ideas from the *lower bound* paper and turning those ideas into new positive results. This result is very appealing to me since it satisfies my desire of solving problems thoroughly and coming up with provably tight results.

I believe that cryptography (and theoretical computer science in general) is not just a beautiful theory but also possesses great practical value and potential. Thus, it is important to understand deeply the system where cryptography can actually be implemented and utilized. For me, the best way to learn about a subject is to do research. Therefore, although my focus is on theory, I also enjoy learning about computer itself and did research on microarchitecture. In particular, I studied microarchitectural optimization. The presence of data-dependent operations and irregular control-flow patterns that are not predictable at compile time results in significant wasteful computation. Luckily, modern processors also feature sophisticated and powerful value predictors, which allows one to predict invariants across long execution intervals correctly with high probability. Therefore, we can eliminate dead code at runtime within the processor based on dynamically predicted data and control invariants. In order to exploit this observation to further optimize sequential code at run-time, we designed and implemented speculative code compaction and tested it on gem5, an open-source system-level and processor simulator. This research experience is invaluable to me. It improved my understanding of microarchitecture and computer systems in general so that it is easier for me to understand the practical potential of cryptographic primitives and help me find practical questions about hardware security and theoretically study them.

I have experience in studying foundations of public key encryption and good understanding of microarchitecture. In the future I would like to continue studying problems in theoretical cryptography. But I am open to exploring research in other areas of theoretical computer science as well. I would be happy to implement prototypes of cryptographic primitives too.