Research Statement

Challenges in Cryptography for Big Data We live in a world with ever-expanding volumes and varieties of data. Our ability to collect and analyze the vast amounts of data greatly increases our understanding of the world. In particular, the growing prevalence of big data technology enables us to e.g., decode human DNA in minutes, find cures for cancer, accurately predict human behavior, and so much more.

While on the one hand big data technology yields great promises, on the other hand, realizing such applications while preserving individual privacy poses new technical challenges for the current cryptographic technology. For instance, cryptographic schemes have traditionally been developed in the circuit model of computation, and this model provides a good testing ground for the feasibility of solutions. Nevertheless, in order to improve practical applicability, it is crucial to develop schemes that incur acceptable overhead when applied to the realistic *random access machine (RAM)* model of computation.

Secure Multi-Party Computation for RAM Programs Secure multi-party computation (MPC) enables parties with private data to collaboratively compute a global function over their private data while keeping those data private. It is is one of the main pillars of cryptography and almost all known cryptographic primitives like encryption, zero-knowledge proofs, program obfuscation, etc. can be formulated as special cases of MPC. Apart from its theoretical importance, MPC has a wide range of practical applications, varying from simple tasks such as coin tossing to more complex ones like electronic auctions, electronic voting, privacy-preserving data mining, and much more.

In the past few decades, both theoretical and practical improvements have been pushing the limits of the overall efficiency of MPC protocols. However, most of these constructions are devised only for circuits and securely computing a RAM program requires the process of first converting it into a circuit. This conversion is unimaginable in the context of big data applications where the size of the circuit can be exponential in the running time of the original RAM program. Therefore, it is paramount that we realize RAM-friendly secure computation techniques that do not suffer from these inefficiencies.

I am particularly interested in the security model in the presence of malicious adversaries, where the adversarial parties may deviate arbitrarily from the protocol execution in the attempt to cheat. This model, compared to the honest-but-curious adversary model, provides a higher security guarantee and is more realistic in practice. I present the *first* constant-round two-party RAM computation protocol secure against malicious adversaries [3]. The protocol allows for multiple RAM programs being executed on a persistent database, and only makes black-box use of one-way functions. In this work I provide a novel cut-and-choose technique that can be seen as an adaptation of the traditional cut-and-choose technique for the RAM setting.

I also study secure RAM computation in the multi-party setting, where the round complexity of all known protocols grows linearly in the running time of the program being computed, and all the round-efficient results are limited to the two-party setting. I present the *first* constant-round multi-party secure computation protocol for RAM programs [2], which allows execution of multiple programs on the same persistent database, and only makes black-box use of one-way functions. I also show how to extend the honest-but-curious results to the malicious setting.

Laconic Oblivious Transfer and its Applications Typical secure multi-party computation solutions require that both the computational complexity and communication complexity scale with the size of the input dataset, which makes it generally unsuitable for even moderate dataset sizes. Over the past few decades, substantial effort has been devoted towards realizing cryptographic primitives that overcome these challenges. This includes works on fully-homomorphic encryption (FHE) and on the RAM setting of oblivious RAM and secure RAM computation. Protocols based on FHE generally have a favorable communication complexity and are basically non-interactive, yet incur a prohibitively large computational overhead (dependent on the dataset size). On the other hand, protocols for the RAM model generally have a favorable computational overhead, but lack in terms of communication efficiency (that grows with the program running time). Can we achieve the best of both worlds?

I address the above question and make concrete positive progress. I introduce a new tool called *laconic oblivious transfer* (or laconic OT for short) [1] that helps to strike a balance between the two seemingly opposing goals. I show various applications of this novel technique to secure multi-party computation on large inputs with better computational complexity and communication complexity. Additionally, I present the *first* multi-hop homomorphic encryption scheme for RAM programs. I believe that laconic OT is a powerful primitive, and I expect it to find other applications in secure computation, especially in the context of big data. I would also like to realize laconic OT based on other standard computational assumptions, such as learning with errors (LWE), or semi-honest OT and hash functions.

Future Work The need to harden or protect computation has become urgent in the face of the proliferation of digital data and the growing demand of collecting and analyzing such data. Considerable recent efforts have been devoted to developing cryptographic schemes that are acceptably efficient when applied to programs that are (as most programs are) designed for machines with random access memory. I would love to continue addressing the challenges of RAM-model cryptography in a variety of topics. Examples include secure distributed computation, secure database and memory access, secure delegation of computation, homomorphic encryption, program obfuscation, leakage resilient computation, and so on. Realizing these constructions, without relinquishing the efficiency of RAM programs, often poses considerable technical hurdles. I believe this will bring us novel ideas and innovative approaches in cryptography.

References

[1] Chongwon Cho, Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Antigoni Polychroniadou. Laconic oblivious transfer and its applications. In *Annual International Cryptology Conference*, pages 33–65. Springer, 2017.

[2] Sanjam Garg, Divya Gupta, Peihan Miao, and Omkant Pandey. Secure multiparty RAM computation in constant rounds. In *Theory of Cryptography Conference*, pages 491–520. Springer, 2016.

[3] Peihan Miao. Cut-and-choose for garbled RAM. *IACR Cryptology ePrint Archive*, 2016:907, 2016.