CAREER: Pushing the Practicality of Secure Multiparty Computation

Overview. Recent advances in networking, hardware, data science, and artificial intelligence have improved personal life by leveraging useful but private information from individuals. It has raised a critical need in data security and privacy with the emergence of regulations like the General Data Protection Regulation and the California Consumer Privacy Act. Secure multi-party computation (MPC), which allows a set of mutually untrusted parties to compute any function jointly and reveals nothing except for the output, is crucial in enabling computation on datasets from multiple sources without sharing them. In recent years, MPC has found a few applications in deployment. However, many hurdles still exist in deploying MPC, and as a result, the adoption of MPC in practice is still lagging behind the surge in privacy needs.

This CAREER proposal aims to overcome major issues when pushing MPC to practice. To this end, the proposed research will focus on three aspects: efficiency, scalability, and security. In particular, the proposal features a powerful compiler that brings malicious security with a very small cost, a suite of MPC protocols designed to compute on large datasets, and foundations to quantify and improve the concrete security level of MPC protocols. The systems built from this proposal will enable programmers to use MPC protocols that meet practical deployment requirements. This CAREER proposal fits perfectly with the PI's long-term career goal of designing practical cryptographic protocols for enhancing the privacy and security of real systems.

Intellectual Merit. This proposal will make a huge leap forward in the efficiency of MPC with strong, malicious security. The proposed compiler can get malicious security with a very small cost while at the same time supporting a large set of generic and special-purpose protocols. To improve the data scalability of MPC, the proposed research will take a co-design approach between oblivious algorithms, including oblivious RAMs and structures, and recent advances in applied cryptography. Such integration brings a new land of opportunities to make MPC data structures practical. Finally, a systematic study on concrete security on MPC protocols will be initiated. Preliminary work from the PI has shown that misunderstanding in concrete security of MPC building blocks can result in a complete breakdown in security. The new foundations and tools will enable a rigorous way to formalize and design protocols with high concrete security.

Broader Impacts. This proposal will improve the practical aspects of MPC for academic research and industry deployments. The PI will integrate systems from this proposal to the EMP-toolkit open-source framework to maximize the impact of the proposed research. The PI will further evaluate the proposed research in real products through industry collaborations with [redacted] and [redacted]. Based on the proposed research, the PI will also make a competitive submission to the NIST standardization effort on threshold cryptography.

Besides the above, the proposal will also apply MPC to problems that arise in the legal domain. In particular, the PI, together with collaborators from Northwestern Law school, will build a system that can resolve a long-standing dilemma in confidential legal settlements, between individual interests and the needs of social justice. For example, such a system is applicable to non-disclosure agreements due to sexual harassment, where the use of MPC can protect individual victims and, at the same time, identify repeated offenders.

The proposed project will lead to a significant educational impact. The PI is active in advising female and minority students. As part of this proposal, an undergraduate research seminar will be organized to bring students with various backgrounds to computer science research. Undergraduate and graduate courses on practical MPC will also be developed based on hands-on projects, with the aim to equip next-generation future entrepreneurs with privacy-enhancement technologies.

1 Introduction

Recent advances in networking, hardware, data science, and artificial intelligence have improved personal life by leveraging useful but private information from individuals. Companies have data from their own users, but they are often prohibited from sharing the data with others, although data sharing could lead to even more convenience in our life. Data sharing is particularly difficult under recent regulations like the General Data Protection Regulation and the California Consumer Privacy Act. Can we resolve the irreconcilables between privacy and utility needs?

Secure multi-party computation (MPC), which allows a set of mutually untrusted parties to compute any function jointly without revealing anything except for the output, is crucial in enabling computation on datasets from multiple sources without sharing them. It allows parties to obtain useful information without revealing anything unnecessary, and thus a perfect tool to solve the above dilemma. Although MPC was originally proposed merely as a theoretical solution, recent advances have gradually pushed MPC towards practice. The practical aspects of MPC now sit in the intersection of cryptography, algorithm, system, and sometimes domain-specific knowledge. Despite the recent advances in MPC and the surging need for it, we still don't see many MPC-enabled applications being deployed. This is partially because most existing MPC protocols only focus on efficiency, particularly in the weak, semi-honest setting. Furthermore, most of them do not consider other important factors in practical deployment like scalability to big data, concrete security, usability, and so on.

Goal of this proposal. This proposal aims to solve the most imperative challenges in pushing MPC technologies to practical applications and presents a highly integrated plan between research, education, and outreach. Our goals include 1) design efficient, scalable MPC with concrete security that meets the efficiency and security needs from MPC-enabled applications. 2) build reliable systems for these protocols and evaluate them in a wide range of practical applications. 3) disseminate the research results through an education plan integrated with other activities. Below we summarize the main tasks in this proposal. See Figure 1 as an overview of their relationships.

Research Thrust 1: Next-Generation Malicious MPC. In the first research thrust, we will investigate ways to improve the concrete efficiency of malicious MPC that provides strong security guarantees. We propose a compiler that can transform any protocol secure with additive errors to full malicious security with a very small overhead. Applying this compiler with recent advances in MPC from our preliminary work will yield maliciously secure protocols with efficiency very close to their semi-honest counterparts. The compiler can be applied to many settings. For example, our preliminary result shows that in the two-party setting, we can get a malicious garbled-circuit protocol with merely 10% to 50% slowdown compared to the best semi-honest protocol. In this proposal, we will further study: 1) more optimization to this compiler, 2) instantiations for other MPC protocols, and 3) using the compiler to compute more complicated functions with mix-mode operations and those with a faster verification procedure.

Research Thrust 2: Cryptographic Protocols for Big Data. The second research thrust focuses on scaling MPC to compute on data of larger sizes. We will investigate the interplay between MPC and oblivious RAM (ORAM), a key building block for RAM-based MPC. Instead of isolating their design, we will study how MPC optimizations can influence the ORAM design and vice versa. Besides, we will apply the same philosophy to data structures. Finally, we will develop efficient and maliciously secure protocols suitable for big data.

Research Thrust 3: Concretely Secure MPC. We will study the concrete security of MPC in the last research thrust. Our goal here is to provide a rigorous way to argue the exact security level of a protocol so that implementations can choose instantiations properly. We will first analyze the concrete security of existing popular protocols, following our preliminary work that finds such an attack for garbling schemes. Then we will find ways to extend existing security definition frameworks, like the standalone model and UC model, to incorporate concrete security. Finally, we will build useful symmetric-key tools and basic protocols that are helpful in building end-to-end concretely secure protocols.

System Building, Evaluation and Application. System building plays a central role in this proposal, as the goal of this proposal is to push the practicality of MPC. We will build systems based on our research

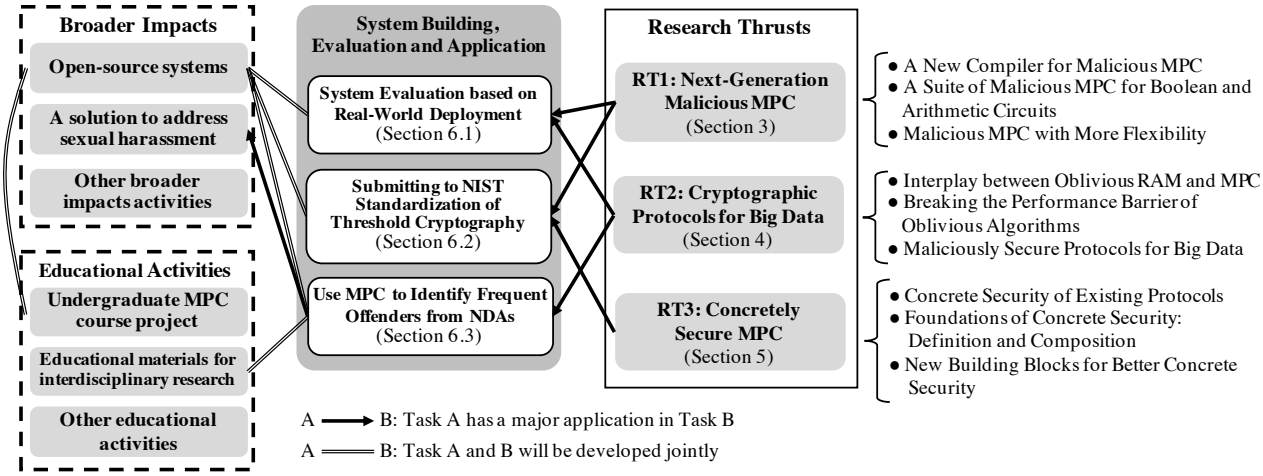

Figure 1: **Overview of this proposal.** This figure shows our main tasks in research, system building, educational activities and other broader impacts. Their relationships are shown in lines and arrows.

outcomes and evaluate them in real use cases (letters of collaboration are attached).

- We will work with our collaborators from academia (Professor Jennie Rogers, Northwestern) and industry ([redacted] and [redacted]) to integrate and evaluate our protocols in real systems and products. See Section 6.1.
- We will prepare a competitive submission to the National Institute of Standards and Technology (NIST) standardization effort on threshold cryptography, based on the research outcome here. See Section 6.2.
- We will explore the intersection between law and MPC with our collaborator (Professor Alyson Carrel, Northwestern Law) and the PI's Ph.D. student (Peter Chan, PhD+JD joint program). We will design and develop a system that uses MPC to discover from confidential settlements important information like frequent sex offenders who are otherwise undetectable. See Section 6.3.

We will use both generic metrics (like execution time) and application-dependent metrics (like usability) to evaluate our research and systems.

Educational Activities. We will integrate our research and system building with our educational plan. See Section 7 for details.

- *Curriculum development.* We will design undergraduate and graduate courses where MPC and related techniques will be introduced. The undergraduate course will include a course project where students use MPC to build useful applications, aiming to inspire students' interests in privacy-enhancement technologies; the graduate course will enable systemization of state-of-the-art techniques in MPC.
- *Educational materials for interdisciplinary research.* We will develop MPC introductory materials suitable for people with different backgrounds, especially those without a background in computer science. This effort goes hand-in-hand with our undergraduate course development, our collaboration in law, as well as other industry collaborations.
- *Undergraduate research seminar.* We will organize an undergraduate research seminar to promote undergraduate students doing research. We will improve the format of the seminar from the pilot run that we did in the prior year, with emphasis on inviting female and minority speakers.

Intellectual Merit. This proposal will make a huge leap forward in the efficiency of MPC with strong, malicious security. The proposed compiler can get malicious security with a very small, sometimes almost optimal, cost while at the same time supporting a large set of generic and special-purpose protocols. To improve the data scalability of MPC, the proposed research will take a co-design approach between oblivious algorithms, including oblivious RAMs and structures, and recent advances in applied cryptography. Such integration brings a new land of opportunities to make MPC data structures practical. Finally, a systematic study on concrete security on MPC protocols will be initiated. Preliminary work from the PI has shown that misunderstanding in concrete security of MPC building blocks can result in a complete breakdown in security. The new foundations and tools will enable a rigorous way to formalize and design protocols with

high concrete security.

Broader Impacts. This proposal will make huge broader impacts as joint outcomes of our research, system building, education, and more activities for open-sourcing and fostering diversity (in Section 8). Below, we summarize all these impacts in three aspects.

- *Impacts on individual.* This project will allow students at different levels, affiliated with Northwestern or not, to understand and use MPC tools with ease. It will expedite the development of MPC-enabled applications with our open-source framework.
- *Impacts on industry.* The outcome of this proposal will push privacy-preserving techniques for a wider deployment in industry. Our integrated effort from this proposal, in research, system, and education, will significantly accelerate the adoption of MPC in real products and the adoption of these MPC-enabled applications.
- *Impacts on society.* Our system that applies MPC to compute on sexual harassment non-disclosure agreements, if deployed, will enhance the interests to sexual harassment victims and bring significant benefit to our society at large.

2 Background and Preliminary Work

2.1 Background

The concept of MPC dates back to the 1980s [149, 65]. After years of improvements, MPC started to attract much attention from industry recently (e.g., [5, 1, 4, 3]; see [2] for a list of companies). There is a massive need for MPC, but the number of MPC-enabled products in practice is still small. This proposal will hunt for solutions to the following most imperative challenges in pushing MPC technologies to practical applications.

1. **Challenges in Efficiency.** The efficiency of MPC protocols is often a hurdle in practical deployment. In particular, MPC protocols with malicious security, where the security holds even if the adversary behaves arbitrarily, are strongly desired. Prior works from the PI and other researchers have made huge progress in this direction but the efficiency is still far from satisfactory, particularly with a dishonest majority. In this proposal, we will design a flexible compiler that can significantly reduce such overhead.
2. **Challenges in Data Scalability.** Many MPC protocols are designed to compute functions represented in circuits and thus inefficient when accessing a private index to a large data. Oblivious RAMs and oblivious algorithms [66] can be used to build efficient protocols friendly to big data (e.g., RAM-based MPC), but even the state-of-the-art systems with such support still incur a considerable cost when the data is large. The proposed work will bring down the cost, both asymptotically and concretely, for computing on big data with a suite of new generic and special-purpose RAM-based MPC protocols and data structures.
3. **Challenges in Security.** Almost all existing security proofs in the MPC literature are asymptotic in the security bound. However in practice, MPC systems need to be instantiated with concrete parameters and thus need concrete security analysis. Our preliminary work [70] found that misunderstanding of concrete security can lead to practical attacks on important MPC building blocks. To deploy MPC with confidence, proofs with concrete security analysis are indispensable. In our proposed work, we will systematically study how to define, design, and efficiently implement MPC protocols with concrete security.

2.2 Our Preliminary Work

We describe works from the PI related to this proposal, which serve as important building blocks and preliminary exploration of the directions in this proposal.

Authenticated garbling for malicious MPC. In 2017, we proposed the authenticated garbling framework [140, 141] for malicious MPC tolerating all-but-one corruption. With follow-up extensions from us [88, 147] and others [75, 135, 155], this framework is still the most efficient in its setting for computing Boolean circuits and has been used as a key tool in numerous follow-up works [89, 21, 41, 136, 154].

Efficient oblivious RAMs and algorithms for MPC. In our past effort [145, 143, 116, 108, 137, 152, 142], we pioneered the research to improve the efficiency of RAM-based MPC by designing oblivious RAMs and algorithms with small circuit complexity. For example, we proposed Circuit ORAM [137], the first ORAM that achieves optimal circuit complexity for a wide range of block sizes. As a whole, our prior work improved

the efficiency of RAM-based MPC by at least $100\times$.

Efficient random correlation generation. Recent works on Pseudorandom Correlation Generator (PCG) [21, 23, 22] lead to a new way to generate correlated oblivious transfer (COT), oblivious linear evaluation (OLE) and vector OLE (VOLE) with very small communication cost (but intensive computation). In our very recent work [148, 146], partially funded by an ongoing DARPA project [1], we designed efficient protocols for COT and VOLE. Under 50Mbps network, our implementation can output a COT correlation in roughly 20 nanoseconds and can output a VOLE correlation (on a 61-bit field) in 100 nanoseconds.

MPC meets symmetric-key cryptography. In our recent series of work [71, 70], we studied how existing MPC protocols instantiate hash functions and found an attack to a garbling scheme implemented in many MPC frameworks. The attack is fast, parallelizable, and economical, making it a real threat when deploying the protocol in practice. We also showed how to rescue the scheme without sacrificing much performance.

Our open-source effort. Our team is a strong believer in open source and has spent a huge amount of effort in maintaining our EMP-toolkit [138] open-source framework for MPC protocols in different settings. EMP-toolkit has been used in many academic projects outside of our team [11, 82, 96, 110, 18, 11, 115, 34, 28, 78, 154, 128] and industry pioneers from [redacted], [redacted], and [redacted].

2.3 PI's Qualifications

The PI has a unique strength in bringing together theory and practice in applied cryptography, and has worked on all layers of applied cryptography in system and applications [116, 144, 142, 134, 108], protocols [97, 79, 69, 68, 87, 140, 141, 139, 152, 137, 143, 145] and foundation [32, 71, 70]. The PI is one of the leading forces in pushing MPC to practice by means of open-source systems; the open-source framework EMP-toolkit has been used widely in academic and industry research, as well as start-up companies building real products. The PI's work on malicious MPC received a best paper award in ACM CCS (2017) and is still leading the efficiency in its target setting. The PI's work on zero-knowledge proofs [87] is a key ingredient in Picnic, a post-quantum signature scheme in submission to NIST Post-Quantum Cryptography Standardization [153], currently an alternate candidate in Round 3.

3 RT1: Next-Generation Malicious MPC

MPC protocols with malicious security have attracted huge attention due to its strong security guarantee. To design a malicious MPC, we usually start with a semi-honest protocol as a stepping stone and strengthen it to malicious security in some way that introduces extra overhead. Although the efficiency of malicious MPC in the honest majority setting has been made very close to the semi-honest security [57, 36, 69, 101, 8, 58, 43, 44, 54, 17], the overhead in the dishonest majority setting is still substantially higher.

There are many generic compilers that work for a family of semi-honest protocols, including the GMW compiler [65], the IPS compiler [84, 85, 77], the AMD compiler [60, 59, 61], and others. Other works focus on the concrete efficiency including, for example, cut-and-choose garbled circuits [102, 129, 130, 98, 103, 80, 100, 26, 6, 139, 81, 105, 106, 126, 119, 118, 56, 55, 120, 95], BMR-style garbled circuits [13, 42, 37, 104, 107, 140, 141, 88, 147, 74], SPDZ [117, 46, 93, 45, 90, 91], and many others. Our first research thrust is to design a set of maliciously secure protocols with efficiency close, in many cases optimally close, to those with semi-honest security, with emphasis to the dishonest majority setting.

3.1 A New Compiler for Malicious MPC

In our new compiler, we consider a special class of semi-honest MPC protocols that are secure up to additive errors against malicious adversaries (inspired by the AMD framework [60, 59, 61]). This means that a malicious adversary cannot learn the honest parties' inputs but can influence the correctness of the computation by adding an error of its own choice to any wire. Intuitively, it means that the adversary can influence the computation of each gate with an error independent of the honest parties' inputs.

To get a full malicious MPC, our compiler needs: 1) an MPC protocol secure up to additive errors (Π_1), and 2) a triple-generation protocol that is also secure up to additive errors (Π_2). Suppose we want to securely

[1] SIEVE: Securing Information for Encrypted Verification and Evaluation. The PI will study practical zero-knowledge proofs in this DARPA project. The proposed project here instead focuses on practical MPC.

evaluate function f that has C gates when represented in field $\mathbb{F}$. Our compiler takes the following steps.

1. Use Π_1 to evaluate f and obtain for each wire a sharing of the wire value.
2. Use a subfield VOLE (sVOLE) protocol (e.g., the ones [148, 146] from our preliminary work discussed in Section 2.2) to authenticate all wire sharings from the first step.
3. Use Π_2 and an sVOLE to generate ℓ number of authenticated triples that are secure up to additive errors.
4. Use the above authenticated triples to check the correctness of authenticated wire sharings in step 2 via a probabilistic triple-checking process to ensure the execution of Π_1 is correct.
5. If the above check does not abort, reveal the sharings of the output wires to the parties.

Our sVOLE protocols [148, 146] on random input need less than 1 bit of communication per correlation regardless of the underlying field size, therefore the main cost is authenticating of non-random elements and calling Π_1 and Π_2. Overall, the main cost includes: running Π_1 on C gates, running Π_2 for ℓ triples, and each party sending $3\ell + C$ field elements to a delegate (which can be any party). Using state-of-the-art randomized checking process and a statistical security parameter $\rho = 40$, ℓ is essentially C for large fields and $2C$ for small fields including the Boolean circuits as long as the circuit is reasonably large. This proposal will formally prove the security of this method and further improve its efficiency.

Relation to prior efforts. To ensure high efficiency for a large set of protocols, our compiler takes the best ideas of many prior works from both categories. For example, as mentioned above, the idea of using protocols secure up to additive errors are from the AMD compiler [60, 59, 61]; the way we eliminate additive errors is inspired from the SPDZ-style protocols [46, 117], the idea of share authentication and checking was rooted from the distributed garbling [42, 140]. However, this does not overshadow the significance of our compiler. As outlined below, our preliminary exploration shows that this compiler is applicable to many settings and significantly more efficient than prior works aiming at concrete efficiency. For example, applying the original AMD compiler to the garbled-circuit protocol would incur an intolerable cost because the AMD compiler uses the underlying semi-honest MPC in a non-black-box way. We will explore the full potential of this framework in the proposed project.

Practical considerations to our compiler. Before going into concrete instantiations, we first discuss some preliminary optimization ideas to the compiler that we plan to study in the proposed projects. All of them can significantly improve the efficiency of our compiler when applied to any protocol.

1. *Optimally cache-efficient triple checking.* The probabilistic triple-checking procedure [117], even after optimization [8], requires randomly shuffling all triples. This procedure is inherently cache-inefficient because accesses are random over a large range, which becomes a big overhead. We propose an improved checking procedure based on random cyclic shifts and can achieve optimal memory-access locality. In the proposed research, we will analyze this new randomized triple-checking procedure and explore its efficient implementation with the goal of being cache-oblivious. That is, the protocol can take advantage of the cache without the need to know the cache size.
2. *Authenticating values in the multi-party setting.* In our preliminary VOLE protocols [148, 146], we can authenticate one bit in about 20 nanoseconds and one 61-bit field element in about 100 nanoseconds, where LPN encoding is the bottleneck. When authenticating to n parties, such LPN encoding would need to be performed for n times, and cause the execution time to increase by a factor of n. This is not necessary as one can potentially use the same LPN coding matrix in all protocols, and compress all LPN encoding into one due to the linearity of LPN assumptions. In the proposed research, we will explore this idea to a further extent, considering how it can affect the LPN assumption.

3.2 A Suite of Malicious MPC for Boolean and Arithmetic Circuits

Now using our new compiler to a protocol secure up to additive errors, we can get protocols that outperform state-of-the-art protocols in almost all settings considered in prior works.

Secret-sharing based malicious MPC. It is well-known that secret-sharing based protocols like GMW is secure up to additive errors, and thus immediately applicable to our compiler. In the binary case, with our recent COT protocol [148], this would lead to a malicious MPC where each party just sends every other party roughly 9 bits per gate. In the arithmetic setting, our protocol would lead to something very similar to SPDZ. However, we estimate the concrete cost is much better than SPDZ, after incorporating recent advances in

Bandwidth	500 Mbps			2 Gbps		
Protocol	Half-gate	Authen-GC	New Result	Half-gate	Authen-GC	New Result
Execution time (ns/gate)	512	4300	550	128	1200	200

Table 1: **Our compiler applied to the case of constant-round two-party Boolean circuits.** Performance assumes the circuit has more than $200,000$ gates. We compare our compiler to the best semi-honest garbling (Half-gate) [151] and the most efficient authenticated garbling (Authen-GC) [147].

OLE [47, 63] that can enable an efficient instantiation of Π_1 and Π_2. This is particularly the case when the number of parties is large, because most communication complexity in our protocol is linear to the number of parties, while for SPDZ only the online phase can enjoy such benefit. To obtain a Π_2 with additively errors, we only need an OLE with additive errors. Recent OLE-based MPC [73, 52] is another direction achieving similar goals. Compared to them, we need less and weaker OLEs (but also need cheap VOLEs). In the proposed work, we will design such additive-error OLE with better efficiency than existing OLE protocols and will explore the performance comparison with SPDZ and other protocols.

Constant-round malicious MPC. What's more interesting is when we apply the compiler to constant-round protocols. Hazay et al. [76] showed an unoptimized distributed garbling protocol is secure up to additive errors, and we believe it also holds to our optimally efficient distributed garbling protocol [88] when combined with our recent COT protocol [148]. In Table 1, we present some preliminary results that we made in recent months of ongoing development. We can observe a $7\times$ improvement in execution time compared to the best prior work, and the performance is very close to the semi-honest setting for a 500 Mbps network. We believe similar improvement also holds for the multi-party setting with our recent improved distributed garbling [147]. In the proposed work, we will study this optimized distributed garbling protocol to obtain concretely efficient, constant-round malicious MPC. We will also explore other distributed garbling based on key-homomorphic PRFs [16] that requires significantly less communication for a large number of parties.

For constant-round malicious MPC for arithmetic circuits, even the most recent work by Makri and Wood et al. [112] is not concretely efficient. We could easily obtain a concretely efficient constant-round malicious MPC for arithmetic circuits, simply by proving that distributed arithmetic garbling is secure up to additive errors. We will study concrete instantiations of this protocol in the proposed work.

3.3 Malicious MPC with More Flexibility

In the above discussion, we focus on improving "classical" malicious MPC protocols, which are already highly impactful to real-world applications (See Section 6). Our compiler does not stop here, and we demonstrate its flexibility to a greater extent where prior generic approaches cannot cover.

Malicious mix-mode MPC. Mix-mode MPC is very useful to further reduce the cost of MPC as different parts of a computation task can be computed in different manners more efficiently. We will study two types of mix-mode MPC: 1) mixing Boolean and arithmetic circuits and 2) mixing joint MPC computation and local computation. The first is useful in many applications where both logical operations and arithmetic computations are needed. The second point refers to a setting where local computation can be used to accelerate the overall task. In the malicious case, we would need to prove in zero-knowledge that the local computation is done correctly and connect them with our MPC protocol. In the proposed project, we will study how to modify Π_1 and Π_2 to fulfill our requirements.

Efficiently verifiable functions. Here we consider extension in another direction. For many functions, computing the function usually takes more time than verifying the results. This kind of function was referred to as efficiently verifiable functions by Hazay et al. [76], who also proposed a protocol that can efficiently compute these functions with one-bit leakage. Using our compiler, we can support the same set of functions with high efficiency but without any leakage. In particular, we will use an additive-secure protocol to execute the computation phase and the verification phase. However, now we only need to check the correctness of the verification phase using our efficient additive-secure triples. This idea can be easily generalized to any number of parties and computing Boolean or arithmetic circuits. In the next section, we will further use this idea to obtain efficient malicious secure RAM computation.

4 RT2: Cryptographic Protocols for Big Data

Most MPC protocols, including ones in Section 3, assume that the function to compute in MPC is represented as a circuit. However, in practice, complicated tasks are often developed in the random-access machine (RAM) model, where dynamic access to an array is supported. The generic transformation from a RAM program running in time T leads to a circuit of size $O(T^3 \log T)$ [39, 124] and is thus not scalable even for medium-sized data. MPC in the RAM model was later proposed [122, 67], with the hope of supporting big data efficiently. Intuitively, we can use an oblivious RAM (ORAM) [66], which is executed in an MPC protocol, to transform sensitive accesses to non-sensitive accesses. Due to the high cost of MPC, most prior works [62, 137, 143, 92, 53, 152], including some from the PI, focus on reducing the circuit complexity of the ORAM. Other works can further improve the efficiency with the observation that for a specific task, it is possible to use better oblivious algorithms and structures that are cheaper than a generic ORAM [150, 145, 19, 114, 50, 116, 121]. However, all of the above focus on minimizing the circuit and thus use MPC as a black-box. Such an approach is simple to design but rules out the possibility of further optimization. Some recent works by the PI and others have begun to explore ways to optimize RAM-based MPC by integrating cryptographic tools to oblivious algorithms and thereby obtaining great improvements [51, 68]. In this research thrust, we will bring recent advances in cryptographic protocols to accelerate MPC for big data to its full potential.

4.1 Interplay between Oblivious RAM and MPC

Reconfiguring ORAM for cryptographic optimizations. For two-party RAM-based MPC, almost all prior works use a garbled-circuit based protocol. This is because other alternatives like GMW also need $O(\kappa)$ bits of communication per gate and thus may not be of great advantage, where κ is the computational security parameter. As a result, most ORAM designs focus on minimizing the circuit size rather than the depth [62, 137, 143, 92, 53, 152]. Our preliminary work Ferret [148] (discussed in Section 2.2), implies an AND-triple generation protocol with less than 1 bit of communication per triple, which is $136\times$ better than the state-of-the-art triple generation protocol [49] in a broad range of network settings. The overall efficiency is also about $100\times$ better than garbled-circuit based protocols when the depth is low.

The proposed work will develop this new GMW protocol and explore its application for two-party RAM-based MPC. In particular, we will design a tree-based ORAM [131] where the client algorithm has a very small depth while keeping the circuit size reasonable. In our prior work [143], we designed a Path ORAM [133] variant where the depth is $O(\log \log n)$ and Circuit ORAM [137] that has a much smaller circuit size but high depth. In the proposed project, we will explore: 1) algorithmic improvements to combine Path-ORAM eviction and Circuit-ORAM eviction for a low-depth algorithm that also has a smaller circuit size. 2) improving parallelization via batch eviction. Note that if we cut few levels of the ORAM tree close to the root, we can perform multiple evictions on disjoint paths in parallel. Such change was proposed for oblivious parallel RAM [20], but here our goal is just to make evictions parallelizable, so that the above GMW protocol can be executed across different evictions efficiently. We will formally prove such modification with batch eviction has a similar probabilistic bound as before and explore various trade-offs in such a design. We will also pay attention to other ORAMs with worse circuit size but better circuit depth and explore their concrete efficiency with the new GMW.

Optimizing cryptographic protocols for recent ORAM frameworks. Recent work [51] has shown that we can improve the concrete efficiency of RAM-based MPC using a PIR scheme that can be easily integrated in RAM-based MPC [24, 25]. It has linear computational complexity and MPC circuit complexity of $O(\sqrt{n})$, where n is the database size. Our preliminary work [68] shows that the circuit computed in MPC can be reduced to polylogarithmic without increasing the server computation. This new framework further offers other benefits, including 1) support dynamic keyword-based access on any bounded range; 3) better performance if one or more parties know the index that is being accessed.

In the proposed work, we will study concrete instantiation of this framework and integrate it with the ORAM optimizations in the first part. It is also anticipated that the local computation from PIR will become a bottleneck after more optimizations on MPC. Although it is known that the complexity of PIR needs to be linear [14], it is possible to reduce the online complexity by a linear preprocessing. Recent work [40]

presented a protocol with a linear offline phase and a sublinear online phase. Unfortunately, such a PIR cannot be easily used for RAM computation, because the client algorithm is very complicated to compute in MPC. In the proposed project, we will 1) improve our protocol by exploring better trade-offs for PIR protocols with an emphasis on the suitability to use in RAM-based MPC and 2) study a three-party RAM-based MPC where such preprocessing-based PIR does not need to be computed in MPC.

4.2 Breaking the Performance Barrier of Oblivious Algorithms

Oblivious algorithms and structures are often very helpful in designing MPC for big data: they can be much more efficient while applicable to many other tasks. Here we will explore cryptographic protocols that can accomplish the same task without computing the whole circuit in MPC.

Oblivious random shuffle. Oblivious random shuffling is an important building block that can be further used to build oblivious sorting [72] and many other useful applications. Here, two parties hold secret sharing of a list of data and want to obtain the secret sharing of them permuted with some secret randomness. It is possible to reduce the use of MPC via local random shuffling: if one party holds the encrypted data block with the key secretly shared, we can ask this party to locally shuffle the data and send to the other party for another independent random shuffle. An idea similar to this was explored in prior work [109] in the context of hash-based ORAM but only as a feasibility result. The main bottleneck is the private encryption and decryption procedure to convert the ciphertext from/to secret shares. Note that this is generally useful as threshold cryptography, which we will explore independently in Section 6.2. Here we will explore the use of LowMC [7] and our improvements in GMW mentioned above to make such an approach practical. In particular, we only need an encryption scheme with a large block size and fairly low data complexity. When instantiated with block ciphers like LowMC, the circuit complexity is within a factor of 4 with respect to the input size. In addition, most lattice-based encryption schemes [86, 35] natively support encryption and decryption on shared keys and are practical when packing is enabled. However, moving blocks between ciphertexts is very expensive and moving blocks within a ciphertext introduce a fair computational overhead. For such lattice-based construction, we will also design 1) efficient perfectly random shuffling algorithms minimizing the computational overhead, and 2) very efficient shuffling with limited randomness, for example, we don't move blocks between ciphertexts, which is sufficient in many applications already.

A blazing fast oblivious stack and its extensions. Oblivious stack has been studied extensively [150, 145, 114] as it can be extended for oblivious queue and oblivious Turing machine, all of which have a $\Omega(\log n)$ lower bound with n as the capacity of the structure. When it comes to circuits, it means that every operation needs to compute $O(\ell \log n)$ gates, where ℓ is the block size. Here, we briefly describe an idea with communication $O(\ell + \kappa \log n)$ bits per operations.

We maintain a list L of encrypted blocks of the form $(\mathsf{payload}, \mathsf{dist})$, where $\mathsf{paylaod}$ is the actual data and dist is an integer, recording the distance between the current block and the last real block in L. We initiate L as empty, and a global integer $\mathsf{Dist} = 0$ and we will maintain the invariant that Dist always record the number of dummy blocks to the last real block in L. For a conditional push operations with data $\mathsf{payload}$, if the operation is real, we will append a block of the form $(\mathsf{payload}, \mathsf{Dist})$ to L and reset $\mathsf{Dist} = 0$; otherwise, we will push a dummy block and increment Dist by 1. For a conditional pop operations, if the operation is real, we will use function secret-sharing to read the element at location Dist. From the block, we will get the distance to the next real block. We can update Dist so that it points to the location of the next real block. We can see that the cost is to encrypt/decrypt a block in MPC (with cost $O(\ell)$ bits after using LowMC and the above GMW protocol) and in the case of conditional pop, to use a function-secret sharing to read a block (with cost $O(\kappa \log n)$ bits). The idea easily extends to a big set of other data structures with locality-based access patterns [145], which we will explore in this proposal.

4.3 Maliciously Secure Protocols for Big Data

Here we will study RAM-based MPC with malicious security. Prior work on malicious RAM-based MPC is rather limited and the overhead is far from being practical [92, 94], partially due to the significant slowdown caused by malicious security. By incorporating new malicious MPC proposed in Section 3, we already obtain much more efficient malicious RAM MPC. We will explore more techniques to further improve the scalability.

Efficiently verifiable oblivious algorithms. Many algorithms, or parts of the algorithms, have a more efficient verification procedure than computation and thus can use our optimization in Section 3.3 for further performance improvement. We can apply this idea in two levels: 1) we can identify components on oblivious algorithms that can be efficiently verified. Such example include oblivious sorting, oblivious compaction, and so on. 2) we could also apply the idea on the whole computation. As pointed by Hazay [76], such example include minimum spanning tree, computing maximum flow of a graph, both of which can be efficiently verified. We will further explore randomized verification procedure that can be more efficiently verified but only provides probabilistic correctness guarantee.

Taking advantage of local computation in malicious setting. In the semi-honest setting, utilizing local computation can often improve efficiency significantly, particularly when computing on large data. However, such improvement becomes unclear in the malicious setting, as an adversary can potentially do wrong local computation or use inconsistent data. In the proposed project, we will enable local computation when computing on large data from two directions.

1. *Algorithm-level checking.* In many algorithms that can benefit from local computation, there may exist a validity check algorithm: the output from local computation is valid if some equivalent input can be extracted efficiently from this output. If this is true, one just needs to run a zero-knowledge protocol to verify the validity of the output from local computation before using it in the MPC. In our preliminary work, we applied this idea to popular a data structure algorithm, Q-Digest [132], and designed such an oblivious validity check algorithm. Although the local computation is linear to the input, the validity check is only sublinear to the input. In the proposed work, we will apply this idea to many more big-data algorithms that can benefit from local computation.
2. *Protocol-level checking.* For reactive MPC protocols, where parties can perform multiple local computation while executing an MPC protocol, the above algorithm-level checking may not work well. In this case, we will apply ideas from Section 3.3, where we will design protocols that mix ZK and MPC together, to efficiently prove that local computation is done correctly.

5 RT3: Concretely Secure MPC

As MPC protocols start to be used in practical systems to protect real sensitive data, it is increasingly important to understand their exact security level. The recent NIST announcement on the planning of threshold cryptography standardization [27], where MPC plays an important role, further emphasizes the importance of considering the concrete security level of MPC. For a long time, MPC security remains asymptotical without concrete analysis. On the other hand, other widely deployed cryptographic tools like cryptographically secure hash functions, signatures, and encryptions, are all accompanied with concrete security analysis. The confusion in MPC concrete security can lead to devastating attacks (as we showed in our preliminary work), and thus hinders the practical deployment of MPC. We will systematically study the concrete security of MPC protocols, including attacks, formulations, constructions and proofs.

5.1 Concrete Security of Existing Protocols

As prior research of concrete security for MPC is very limited, we will first examine the concrete security level of popularly used protocols. In our preliminary work, we studied the widely implemented half-gate garbling [151] and developed an explicit attack with running time $O(k/\log C)$ for garbled keys of k-bits and C as the number of gates garbled by a party [70]. This means that the concrete security parameter of this scheme is roughly 30 to 40 bits less than k. Our first task here is to study the practice and theory of concrete security of existing MPC protocols.

Optimally secure garbled circuits in the standard model. We observe that all garbling protocols in the standard model suffer from a security loss like above. For example, the proof of standard PRF-based garbled circuits contains C hybrids, each of which invokes the security of the underlying PRF. As a result, the final security bound is roughly $C \cdot \epsilon_{\text{PRF}}$, where ϵ_{PRF} is the security bound of standard PRF game. Although we designed an optimally secure construction in our preliminary work [70], it is inherently based on the ideal-cipher model. In the proposed project, we will study this problem in two directions. 1) We plan to seek connections between this problem and the multi-key security degradation issue that has been studied

extensively in the symmetric-key community [15, 111]. This will provide us with insight on impossibility results and proper ways to get around with the possible lower bound. 2) In our preliminary work [70], we presented a hash-function definition suitable to make optimally secure garbled circuits. We will explore its instantiations in the standard model.

Concrete security of other building blocks. We believe the above observation also exists in other building blocks like the GGM-PRF construction [64], which is increasingly important given recent works on function secret sharing [24, 25] and Pseudorandom Correlation Generators (PCG) [21, 23, 22, 148, 146]. However, the security implication can be different depending on the use case. In addition to symmetric-key primitives, we will also study more complicated protocols starting from basic ones like oblivious transfer (OT). For example, we will conduct a systematic study on the tightness of security reduction of various OT protocols [123, 38, 10], including those that the PI recently designed [32, 33]. However, such an approach cannot scale well to more complicated protocols due to the lack of composition, which we will study in the next task.

5.2 Foundations of Concrete Security: Definition and Composition

With sufficient analysis of existing protocols, we will seek to extend popular security models with the support of concrete security analysis. Note that it might be possible to directly use existing composition results for concrete security [99, 29, 30], they are cumbersome to use as they are designed mainly for asymptotic analysis. We will explore ways to provide more flexible notions that are better for analyzing concrete security.

1. *Property-based definition.* Protocols not defined in the ideal-real paradigm may not be composed easily, but sufficient in certain cases. For example, we could prove that the protocol is private with some concrete security level using a game-based definition, but only a lower concrete security level can be claimed for other properties like correctness or input independency.
2. *Semi-honest security.* Semi-honest security can be easier to define and prove because the simulator does not need to extract adversary's input, which could potentially increase the running time of the simulator and thus reduce the concrete security. However, we still need to consider composition carefully: for example, sequential and parallel composition may affect concrete security in different ways.
3. *Malicious security.* In the malicious case, the simulator needs to extract adversary's effective input, which may involve rewinding for one or more times. Rewinding can harm the concrete security and make the analysis challenging. Therefore, we will start with malicious MPC with a straight-line simulation. This will also make concrete security proofs in the UC model simple [31].

We will also study frameworks that combine different levels. For example, a protocol can have one level of concrete security with respect to semi-honest adversaries while having a lower concrete security level with respect to malicious adversaries.

5.3 New Building Blocks for Concrete Security

Our final goal is to propose a set of new building blocks that can be used for obtaining high concrete security easily. Note that even defining an appropriate tool is a highly non-trivial task: if the definition is too powerful, there may not be an efficient protocol with high concrete security; if the definition is too weak, it may not be very useful. In our prior work [70], we proposed a special variant of correlation robust hash function [83] that can be instantiated with high concrete security while at the same time only need one call to the ideal cipher. This is widely useful to many garbling schemes, OT protocols, and more.

In our proposed work, we will use the above hash function to build other MPC building blocks like OT and VOLE with high efficiency. They will be particularly useful to obtain high concrete security for our RT1 protocols. We will also explore other important primitives that can be widely used, including the GGM-PRF construction [64]. For practical purposes, we will study two instantiations, one based on ideal cipher (to be instantiated using AES), and one based on large-domain random permutation (to be instantiated using the SHA-3 permutation function). Although the second is slower, it can potentially be more useful since the output size is much larger.

6 System Building, Evaluation and Application

6.1 System Evaluation based on Real-World Deployment

We will implement protocols from our research thrusts and evaluate their practicality based on real-world deployments with our collaborators in academia and industry (see letters of collaboration). Below we describe three deployment scenarios that we will focus on.

Efficient MPC for Anonymous Payment Channels. We are working closely with [redacted], a company that builds privacy-preserving off-chain payment systems between individuals. It uses malicious two-party computation as a key building block. Their current system is based on our open-source implementation of authenticated garbling [140]. Our research outcomes in RT1 and RT3 will lead to a concretely secure malicious two-party computation protocol with essentially optimal efficiency. This speedup will revolutionize the user experience of their product. We will work with researchers and engineers from [redacted] to incorporate our protocol implementation to their product and evaluate its real-world performance. [redacted]

Federated relational databases. I have been collaborating with Jennie Rogers, assistant professor at Northwestern University, in developing efficient federated relational databases that use MPC to execute SQL queries over multiple data owners securely. In our current collaboration, we use our EMP-toolkit [138] as the primary MPC backend. Professor Rogers is deploying our system to the Chicago Area Patient-Centered Outcomes Research Network (CAPriCORN). CAPriCORN is a data-sharing platform for the greater Chicago area covering 14 hospitals with data from more than 10 million patients. Our ongoing collaboration that combines MPC, sampling, and differential privacy just got accepted to VLDB 2020 [12], and we will continue pushing forward our collaboration to improve federated databases using the outcome of this proposal. In particular, the research results in RT2 can significantly improve the efficiency of important data structures when running SQL in MPC. We will implement, integrate, and evaluate our RT2 protocols in this context. See letter of collaboration. Implementation and evaluation within the scope of this proposed project do not involve any human subject research or personal information.

Very large-scale MPC. I have been collaborating with [redacted], a research scientist at [redacted], in architecting our EMP-toolkit to fit their internal applications that need MPC with massive parallelization. Her team has successfully deployed the semi-honest protocol from EMP between two parties, where each party consists of hundreds of machines. The whole MPC execution, even after massive parallelization, takes a few hours to finish. Existing maliciously secure protocols are not efficient enough for such deployment, but our new RT1 protocol could be optimized for such large-scale computation. We will work with Dr. [redacted] team to ensure seamless deployment of our protocol to their scalable systems and evaluate how it performs in very large-scale computation.

Evaluation metrics. Our benchmark and the above deployment cases will focus on the following metrics.

- *Concrete security of the implementation.* We will use our RT3 results to evaluate the concrete security of our implementation. With this effort, we aim to provide the first MPC toolset with accurate and formal concrete security claims.
- *End-to-end execution time.* We will evaluate the execution time of our protocols using standard computational instances from Amazon EC2 (e.g., `m5.2xlarge`). In this way, other works can benchmark their system on the same hardware and compare it with our system easily. The evaluation will also be performed on configurations similar to real deployment settings. We will record the performance of the whole protocol as well as each of the subcomponents.
- *Hardware-independent evaluation.* We will report other hardware-independent metrics of the protocol, including network communication, round-trip complexity, the number of symmetric-key operations, and so on. This helps to identify unexpected bottlenecks in our protocol design and system implementation.

6.2 Submitting to NIST Standardization of Threshold Cryptography

Standardization is a crucial step in pushing MPC to practice as it can increase the public confidence in MPC technologies. The PI is currently in a team submitting to the NIST Standardization of Post-Quantum Cryptography (PQC) [153], partially based on the PI's prior work on zero-knowledge proofs [87]. Our

signature scheme, Picnic, is currently among the round-3 alternate candidates. NIST is currently planning on the standardization of threshold cryptography, where MPC plays a key role in executing public-key and symmetric-key operations over multiple parties [27]. The PI will actively participate in this standardization and use our experience from the PQC standardization to maximize our chance of selection. Results from RT1 and RT3 will enable us to make a strong submission with outstanding efficiency and security. Based on the most recent report on the roadmap of this standardization [27], we plan to make a submission to the multi-party track for computing symmetric-key primitives and explore potential extensions to support public-key primitives. To comply with the typical no-patent requirement from NIST, we are committed not to file patents on technologies related to this standardization.

Extra requirements in system building and evaluation. The NIST standardization process involves high scrutiny in both protocol design and system implementation. We will make extra effort to develop reference and optimized implementations to meet the requirements. In particular, we will make our implementation available across multiple architectures, including x86 and ARM architectures like Cortex-M. In addition to the generic metrics mentioned above, we will also evaluate other metrics that are often ignored in many cases. These metrics include the executable size, the number of lines of code, the exact memory consumption, and so on. On the ARM architecture, we will further measure the energy consumption of our system.

6.3 Use MPC to Identify Frequent Offenders from NDAs

I am collaborating with professor Alyson Carrel from Northwestern Pritzker School of Law using MPC to solve dilemmas in confidential settlements like non-disclosure agreements (NDA). See letter of collaboration. Professor Carrel is an expert in negotiation and mediation and has extensive experience in the judicial system.

As a bit of background, mediation and settlement outside of the court date back to 1976 [127]. Since then, the debate has never stopped on its impact (See [113] and references therein). On the one hand, it provides justice to the victims: it allows individual autonomy over the outcome of the case and provides a voice for those who may otherwise not be able to use the court systems due to limited financial resources. On the other hand, it has also received criticism that it can potentially harm society as criminals can get away undetected. The recent revelations of sexual harassment from celebrities have demonstrated this debate concretely. We have witnessed multiple occurrences where victims take the risk of breaking NDAs and speak publicly about repeated sex offenders. These victims are often restricted from speaking publicly due to the NDAs they signed. From this perspective, NDAs prevent sex offenders from being identified in the early stages. However, NDAs indeed provide the victims with some justice and interests, especially when there is a huge imbalance in social status between the offenders and the victims.

We plan to use MPC to solve such a dilemma. In particular, we will use an MPC protocol executed among a set of law firms, where they submit the NDAs that they executed. The MPC protocol can output, e.g., a list of offenders that appears in more than a certain number of cases. Although the use of MPC technology seems simplistic, a lot of practical factors must be considered, which in turn calls for new requirements for MPC. Callisto [125] and SATE [9] are also cryptography-enabled systems for detecting serial sexual predators. However, our emphasis here is not about individual reporting but identification from signed NDAs. The goal of our system is complementary to prior works, as NDAs might prevent the victim from using their systems. Because of this, our preliminary study discovered new problems that we will study and overcome in this proposal.

1. *Motivation.* Law firms representing the interest of the defendants (that is, those who are accused of sexual harassment) may not be willing to participate in an MPC execution. Based on our preliminary discussion with attorneys in the local area, we will add more features to motivate participation better. For example, our preliminary exploration shows that we could also identify frequent plaintiffs who appear in many NDAs, which could potentially be linked to fraudulent accuses. This is something useful to the defendants, particularly those with high social status. Other long-term solutions include making the validity of NDA dependent on the frequency of the committed sexual harassment by the defendants. We will explore their legal implications in this proposal.
2. *Trust in MPC.* As a new technology, it may take some time for lawyers to trust that MPC indeed holds the claimed property. Our NIST standardization effort described above can partially help with building the

trust. The PI plans to organize workshops and crash courses, targeting people without a background in computer science, to educate law practitioners with MPC. This will be conducted as part of our educational activities (Section 7).

3. *MPC with enhanced properties.* Due to the sensitivity of the execution, we may consider extra properties, including guaranteed output delivery, public auditability, computation on committed inputs, and so on. We will develop these add-ons when needed.

The PI is advising a graduate student, Peter Chan, who is currently in the PhD+JD program at Northwestern University (details see Section 7). With knowledge in both cryptography and law, Peter will be partially supported by this proposal to bridge the effort made by the two fields.

Extra requirements in system building and evaluation. Unlike other collaborations in this proposal, the MPC system here will be used by people in law firms without any background in computer science. Therefore, it is crucial to develop a front-end that is easy to use. We plan to develop a web-based front-end that can help with entering data. We will evaluate our system focusing on wide-area networks and limited hardware configuration to simulate a practical deployment setting. We will also evaluate the usability of our system with lawyers. System design and building in this project do not involve any human subject or sensitive information.

7 Integrated Education Plan

New curriculum development. In the past year, I developed two new courses at Northwestern: introduction to cryptography (undergraduate) and advanced cryptographic protocols (graduate). In the undergraduate curriculum, I allocated one lecture for teaching basic MPC concepts using the classical five-card trick to compute an AND gate [48] as an example. Poker cards were distributed to the students so that they could try it in class. In the next offering of this course, I plan to design a course project where students can use MPC, specifically the open-source framework from this proposal (See Section 8.1 for details), to build applications that otherwise need a trusted third party. The goal is twofold. 1) By allowing students with only basic background on cryptography to use our system, we will be able to evaluate the usability of our framework and understand how to improve our systems for a broader deployment. 2) By letting our undergraduate students develop "toy" applications using MPC, we foster a better environment for these future entrepreneurs to think about real start-up ideas based on MPC.

As part of the proposed project, the PI will also develop a new graduate-level course that focuses on the interplay between cryptography and system design, covering protocol design methodologies that can benefit from modern computer system architecture and tricks to implement cryptographic protocols efficiently on various CPU architectures. All curriculum materials, including course projects and lecture notes, will be made publicly available. During course design, the PI will take it into consideration that the lecture should be easy to follow even when taught remotely.

Educational materials for interdisciplinary research. To extend MPC to more applications like the one we are exploring between confidential settlements and MPC, we will design short videos and examples to explain the concepts and pitfalls of MPC to people without a background in computer science. This will also help our study in Section 6.3, where we need to explain the concept and the mode of execution to lawyers and potentially judges. We will also make a set of tutorials on using our open-source framework to enable fast development of MPC-enabled prototypes by programmers without any cryptography background. The development of these tutorials will go hand-in-hand with the undergraduate course project mentioned above.

Student advising. I believe that every student can succeed in their Ph.D. if we can find their unique strength. I have been sticking to this philosophy to my current Ph.D. students. Chenkai Weng has extensive experience in system building and has been working on practical MPC protocols and their implementations. As a first-year student, he has already published two papers from top-tier conferences and is taking the lead on some preliminary works pertaining to this proposal [70, 148, 146]. Peter Chan is a PhD-JD student from a joint program between Northwestern Pritzker School of Law and the graduate school. Peter's expertise on judiciary and cryptography provides him a unique angle on how to push MPC to practical applications in the legal domain, especially the collaboration described in Section 6.3.

Organize undergraduate research seminar. I have been organizing an undergraduate research seminar series in my first year at Northwestern. The seminar targets undergraduate CS students who are interested in doing research. More than one hundred students showed up in the first introductory session. In the second quarter, I invited speakers at different stages of their research career, including a senior undergraduate student with extensive research experience, a graduating Ph.D. student, a junior assistant professor, and a tenured professor. Every speaker gave a talk on their research, followed by an open session where students ask questions. The PI will continue improving this seminar series and find solutions to host the talk series during remote teaching quarters. When hosted online, speakers from other institutions will be invited too.

8 Broader Impacts

In addition to the immediate impacts from our efforts mentioned in system building and application (Section 6) and education activities (Section 7), we discuss two more aspects in broader impacts.

8.1 Open-source systems

The main objective of this proposal is to push MPC to practice, and thus system building is indispensable. The PI has maintained an excellent record on system building and open sourcing [143, 116, 108, 137, 144, 152, 142, 139, 140, 141, 79, 97, 71, 70]. Many of these systems have been actively maintained as part of EMP-toolkit [138], an open-source framework for efficient MPC in existence since 2016. It has been used widely in recent years as discussed in Section 2.2. The practice of open sourcing has become a default in PI's research. Indeed, most collaborations discussed in Section 6.1 are made possible thanks to our framework.

The PI will continue the open-source effort to streamline the process from cryptography research, to system implementation and integration, and then to real products. Our open-source plan will be coordinated with other parts of this proposal, including system building and benchmarking (Section 6), a playground for our undergraduate-level MPC course project and graduate-student advising (Section 7), and a tool for our interdisciplinary collaboration (Section 6.3). More details on how we manage our open-source projects can be found in the Data Management Plan supplementary document. For the proposed project, we plan to accomplish the following open-source milestones.

Supporting protocol-agnostic development. In practical deployment, we usually start with a simpler semi-honest protocol and later switch to a protocol with malicious security. This will become even more common as the proposed research will minimize the overhead of malicious security for a large set of protocols. We will open-source a programming framework so that MPC application development can be done in a protocol-agnostic way. As a result, switching protocols only need to modify a few lines of code.

Flexible and secure data structures. We will build a library of advanced secure data structures from Section 4. To make our library easy to use, we will follow the conventions from the C++ Standard Template Library (STL) so that programmers' prior experience can be brought to MPC programming seamlessly.

Multi-architecture support. Almost all state-of-the-art MPC implementations, including those by the PI, assume an x86 platform. Given the dominance of ARM on the mobile platform, we will extend all systems from this proposal to support other platforms like ARM.

8.2 Fostering Diversity and Outreach Activities

A solution to address sexual harassment. As described in Section 6.3, the proposed research collaboration will develop a system that uses MPC to enhance private settlement, which has a major application in protecting victims in sexual harassment. By deploying our MPC-based solution, it could protect individual victims' interests and rights that come with NDA while at the same time protecting social justice by identifying frequent sex offenders, who can do huge harm to our society. With the help from local courts, the PI and his collaborators will make trial executions using data from local law firms to demonstrate its potential impact and continue promoting the system to law practitioners. We believe our tool has the potential to significantly change how sexual harassment settlements are handled and thus can make a long-term impact on our society.

Hosting a Chicago-area cryptography workshop. The PI is committed to promoting multi-institutional collaboration and plans to initiate a Chicago-area cryptography workshop with researchers from all institu-

		Year 1	Year 2	Year 3	Year 4	Year 5
RT 1	Task 3.1					
	Task 3.2					
	Task 3.3					
RT 2	Task 4.1					
	Task 4.2					
	Task 4.3					
RT 3	Task 5.1					
	Task 5.2					
	Task 5.3					

Table 2: **Project Timeline.** Purple blocks ( ) denote research efforts; red blocks ( ) denote implementation and evaluation.

tions in the Chicago metropolitan area. Due to the influence of pandemic, the reading group will be hosted online in the short term and move to an in-person format depending on the situation. The budget has been appropriately reserved to cover the cost of hosting the reading group when it rotates to Northwestern.

Supporting female and minority participation. The PI has been very active in promoting the participation of female and minority students in his first year at Northwestern. The PI just finished hosting a long-term female visitor Xiao Lan and is advising a female master student Zixuan Zhu, who also intends to continue pursuing a Ph.D. degree. The PI serves on the graduate recruitment committee of the CS department and is actively promoting the hiring of female and minority students. Indeed, one of the PI's incoming students is from the gender and sexual minority communities. With the help of these students, the PI will continue promoting participation from female and minority communities, including hosting in-person as well as online gathering with minority students. When organizing the undergraduate research seminar described in Section 7, preferences will also be given to female and minority speakers.

9 Project Timeline

In Table 2, we show the timeline of our main research activities (denoted as *RT1*, *RT2*, and *RT3*). Purple blocks ( ) are research efforts, and red blocks ( ) are prototyping and evaluation. We list other activities below, including system implementation, evaluation, and application (*SYS*), broader impacts (*BI*), and educational activities (*EDU*). For tasks that require our effort over the span of the whole project, we will not list them below. Such tasks include, for example, general open-source effort, supporting female and minority participation, student advising, and so on.

Year 1. *SYS:* Study the details of settlements, explore ways to motivate participation, and build a prototype. *BI:* Initiate a Chicago-area cryptography workshop. *EDU:* Organize an undergraduate research seminar.

Year 2. *SYS:* Deploy our malicious 2PC with industry collaborators. *BI:* Participate in NIST threshold cryptography standardization. *EDU:* Design and teach an MPC-based undergraduate course project.

Year 3. *SYS:* Implement optimized oblivious algorithms and data structures and integrate them to federated databases. *BI:* Extend the NIST submission with multi-architecture support and provable concrete security. *EDU:* Design educational materials for interdisciplinary research.

Year 4. *SYS, BI:* Deploy our systems for private settlement in the local area; find support from the judicial system. *BI, EDU:* Make crash courses based on year-3 material and share them with local law firms.

Year 5. *SYS, BI:* Obtain broader support to use our system in the context of sexual harassment and other applicable settings. *EDU:* Design a course on the system perspective of applied cryptography based on experience from prior years.

10 Results from Prior NSF Support

The PI does not have prior NSF grant support.

References

[1] Cybernetica, Accessed in July, 2020. https://cyber.ee/.

[2] Mpc alliance, Accessed in July, 2020. https://www.mpcalliance.org/.

[3] Partisia, Accessed in July, 2020. https://partisia.com/.

[4] Sepior, Accessed in July, 2020. https://sepior.com/.

[5] Unbound tech, Accessed in July, 2020. https://www.unboundtech.com/.

[6] Arash Afshar, Payman Mohassel, Benny Pinkas, and Ben Riva. Non-interactive secure computation based on cut-and-choose. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 387–404. Springer, Heidelberg, May 2014.

[7] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454. Springer, Heidelberg, April 2015.

[8] Toshinori Araki, Assi Barak, Jun Furukawa, Tamar Lichter, Yehuda Lindell, Ariel Nof, Kazuma Ohara, Adi Watzman, and Or Weinstein. Optimized honest-majority MPC for malicious adversaries - breaking the 1 billion-gate per second barrier. In *2017 IEEE Symposium on Security and Privacy*, pages 843–862. IEEE Computer Society Press, May 2017.

[9] Venkat Arun, Aniket Kate, Deepak Garg, Peter Druschel, and Bobby Bhattacharjee. Finding safety in numbers with secure allegation escrows. In *ISOC Network and Distributed System Security Symposium – NDSS 2020*. The Internet Society, 2020.

[10] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer and extensions for faster secure computation. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013: 20th Conference on Computer and Communications Security*, pages 535–548. ACM Press, November 2013.

[11] Johes Bater, Xi He, William Ehrich, Ashwin Machanavajjhala, and Jennie Rogers. Shrinkwrap: Efficient sql query processing in differentially private data federations. *Proc. VLDB Endow.*, 12(3):307–320, November 2018.

[12] Johes Bater, Yongjoo Park, Xi He, Xiao Wang, and Jennie Rogers. Saqe: Practical privacy-preserving approximate query processing for data federations. *Proc. VLDB Endow.*, November 2020.

[13] Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols (extended abstract). In *22nd Annual ACM Symposium on Theory of Computing*, pages 503–513. ACM Press, May 1990.

[14] Amos Beimel, Yuval Ishai, and Tal Malkin. Reducing the servers' computation in private information retrieval: PIR with preprocessing. *Journal of Cryptology*, 17(2):125–151, March 2004.

[15] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 259–274. Springer, Heidelberg, May 2000.

[16] Aner Ben-Efraim, Yehuda Lindell, and Eran Omri. Efficient scalable constant-round MPC via garbled circuits. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 471–498. Springer, Heidelberg, December 2017.

[17] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *20th Annual ACM Symposium on Theory of Computing*, pages 1–10. ACM Press, May 1988.

[18] F. Benhamouda, S. Halevi, and T. Halevi. Supporting private data on hyperledger fabric with secure multiparty computation. *IBM Journal of Research and Development*, 63(2/3):3:1–3:8, 2019.

[19] Marina Blanton, Aaron Steele, and Mehrdad Aliasgari. Data-oblivious graph algorithms for secure computation and outsourcing. In Kefei Chen, Qi Xie, Weidong Qiu, Ninghui Li, and Wen-Guey Tzeng, editors, *ASIACCS 13: 8th ACM Symposium on Information, Computer and Communications Security*, pages 207–218. ACM Press, May 2013.

[20] Elette Boyle, Kai-Min Chung, and Rafael Pass. Oblivious parallel RAM and applications. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 175–204. Springer, Heidelberg, January 2016.

[21] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. Compressing vector OLE. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018: 25th Conference on Computer and Communications Security*, pages 896–912. ACM Press, October 2018.

[22] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Peter Rindal, and Peter Scholl. Efficient two-round OT extension and silent non-interactive secure computation. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 291–308. ACM Press, November 2019.

[23] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 489–518. Springer, Heidelberg, August 2019.

[24] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 337–367. Springer, Heidelberg, April 2015.

[25] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing: Improvements and extensions. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pages 1292–1303. ACM Press, October 2016.

[26] Luís T. A. N. Brandão. Secure two-party computation with reusable bit-commitments, via a cut-and-choose with forge-and-lose technique - (extended abstract). In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 441–463. Springer, Heidelberg, December 2013.

[27] Luís T. A. N. Brandão, Michael Davidson, and Apostol Vassilev. Nist roadmap toward criteria for thresholdschemes for cryptographic primitives, June 2020. https://doi.org/10.6028/NIST.IR.8214A.

[28] Niklas Büscher, Daniel Demmler, Nikolaos P Karvelas, Stefan Katzenbeisser, Juliane Krämer, Deevashwer Rathee, Thomas Schneider, and Patrick Struck. Secure two-party computationin a quantum world. In *ACNS 20: 18th International Conference on Applied Cryptography and Network Security*, Lecture Notes in Computer Science. Springer, Heidelberg, 2020.

[29] Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, January 2000.

[30] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science*, pages 136–145. IEEE Computer Society Press, October 2001.

[31] Ran Canetti, Asaf Cohen, and Yehuda Lindell. A simpler variant of universally composable security for standard multiparty computation. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 3–22. Springer, Heidelberg, August 2015.

[32] Ran Canetti, Pratik Sarkar, and Xiao Wang. Blazing fast OT for three-round UC OT extension. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 12111 of *Lecture Notes in Computer Science*, pages 299–327. Springer, Heidelberg, May 2020.

[33] Ran Canetti, Pratik Sarkar, and Xiao Wang. Efficient and round-optimal oblivious transfer and commitment with adaptive security. Cryptology ePrint Archive, Report 2020/545, 2020. https://eprint.iacr.org/2020/545.

[34] Melissa Chase, Esha Ghosh, and Oxana Poburinnaya. Secret shared shuffle. Cryptology ePrint Archive, Report 2019/1340, 2019. https://eprint.iacr.org/2019/1340.

[35] Hao Chen, Ilaria Chillotti, and Ling Ren. Onion ring ORAM: Efficient constant bandwidth oblivious RAM from (leveled) TFHE. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 345–360. ACM Press, November 2019.

[36] Koji Chida, Daniel Genkin, Koki Hamada, Dai Ikarashi, Ryo Kikuchi, Yehuda Lindell, and Ariel Nof. Fast large-scale honest-majority MPC for malicious adversaries. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 34–64. Springer, Heidelberg, August 2018.

[37] Seung Geol Choi, Jonathan Katz, Alex J. Malozemoff, and Vassilis Zikas. Efficient three-party computation from cut-and-choose. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 513–530. Springer, Heidelberg, August 2014.

[38] Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. In Kristin E. Lauter and Francisco Rodríguez-Henríquez, editors, *Progress in Cryptology - LATINCRYPT 2015: 4th International Conference on Cryptology and Information Security in Latin America*, volume 9230 of *Lecture Notes in Computer Science*, pages 40–58. Springer, Heidelberg, August 2015.

[39] S. A. Cook and R. A. Reckhow. Time-bounded random access machines. In *Proc. 4th ACM Symposium on Theory of Computing (STOC)*, pages 73–80, 1972.

[40] Henry Corrigan-Gibbs and Dmitry Kogan. Private information retrieval with sublinear online time. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 44–75. Springer, Heidelberg, May 2020.

[41] Ivan Damgård, Daniel Escudero, Tore Kasper Frederiksen, Marcel Keller, Peter Scholl, and Nikolaj Volgushev. New primitives for actively-secure MPC over rings with applications to private machine learning. In *2019 IEEE Symposium on Security and Privacy*, pages 1102–1120. IEEE Computer Society Press, May 2019.

[42] Ivan Damgård and Yuval Ishai. Constant-round multiparty computation using a black-box pseudorandom generator. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 378–394. Springer, Heidelberg, August 2005.

[43] Ivan Damgård, Yuval Ishai, and Mikkel Krøigaard. Perfectly secure multiparty computation and the computational overhead of cryptography. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 445–465. Springer, Heidelberg, May / June 2010.

[44] Ivan Damgård, Yuval Ishai, Mikkel Krøigaard, Jesper Buus Nielsen, and Adam Smith. Scalable multiparty computation with nearly optimal work and resilience. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 241–261. Springer, Heidelberg, August 2008.

[45] Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P. Smart. Practical covertly secure MPC for dishonest majority - or: Breaking the SPDZ limits. In Jason Crampton, Sushil Jajodia, and Keith Mayes, editors, *ESORICS 2013: 18th European Symposium on Research in Computer Security*, volume 8134 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Heidelberg, September 2013.

[46] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 643–662. Springer, Heidelberg, August 2012.

[47] Leo de Castro, Chiraag Juvekar, and Vinod Vaikuntanathan. Fast vector oblivious linear evaluation from ring learning with errors. Cryptology ePrint Archive, Report 2020/685, 2020. https://eprint.iacr.org/2020/685.

[48] Bert den Boer. More efficient match-making and satisfiability: The five card trick. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology – EUROCRYPT'89*, volume 434 of *Lecture Notes in Computer Science*, pages 208–217. Springer, Heidelberg, April 1990.

[49] Ghada Dessouky, Farinaz Koushanfar, Ahmad-Reza Sadeghi, Thomas Schneider, Shaza Zeitouni, and Michael Zohner. Pushing the communication barrier in secure computation using lookup tables. In *ISOC Network and Distributed System Security Symposium – NDSS 2017*. The Internet Society, February / March 2017.

[50] Jack Doerner, David Evans, and abhi shelat. Secure stable matching at scale. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pages 1602–1613. ACM Press, October 2016.

[51] Jack Doerner and abhi shelat. Scaling ORAM for secure computation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 523–535. ACM Press, October / November 2017.

[52] Nico Döttling, Satrajit Ghosh, Jesper Buus Nielsen, Tobias Nilges, and Roberto Trifiletti. TinyOLE: Efficient actively secure two-party computation from oblivious linear function evaluation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 2263–2276. ACM Press, October / November 2017.

[53] Sky Faber, Stanislaw Jarecki, Sotirios Kentros, and Boyang Wei. Three-party ORAM for secure computation. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part I*, volume 9452 of *Lecture Notes in Computer Science*, pages 360–385. Springer, Heidelberg, November / December 2015.

[54] Matthew K. Franklin and Moti Yung. Communication complexity of secure computation (extended abstract). In *24th Annual ACM Symposium on Theory of Computing*, pages 699–710. ACM Press, May 1992.

[55] Tore Kasper Frederiksen, Thomas P. Jakobsen, Jesper Buus Nielsen, and Roberto Trifiletti. TinyLEGO: An interactive garbling scheme for maliciously secure two-party computation. Cryptology ePrint Archive, Report 2015/309, 2015. http://eprint.iacr.org/2015/309.

[56] Tore Kasper Frederiksen, Thomas Pelle Jakobsen, Jesper Buus Nielsen, Peter Sebastian Nordholt, and Claudio Orlandi. MiniLEGO: Efficient secure two-party computation from general assumptions. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 537–556. Springer, Heidelberg, May 2013.

[57] Jun Furukawa and Yehuda Lindell. Two-thirds honest-majority MPC for malicious adversaries at almost the cost of semi-honest. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 1557–1571. ACM Press, November 2019.

[58] Jun Furukawa, Yehuda Lindell, Ariel Nof, and Or Weinstein. High-throughput secure three-party computation for malicious adversaries and an honest majority. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 225–255. Springer, Heidelberg, April / May 2017.

[59] Daniel Genkin, Yuval Ishai, and Antigoni Polychroniadou. Efficient multi-party computation: From passive to active security via secure SIMD circuits. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 721–741. Springer, Heidelberg, August 2015.

[60] Daniel Genkin, Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and Eran Tromer. Circuits resilient to additive attacks with applications to secure computation. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 495–504. ACM Press, May / June 2014.

[61] Daniel Genkin, Yuval Ishai, and Mor Weiss. Binary AMD circuits from secure multiparty computation. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B: 14th Theory of Cryptography Conference, Part I*, volume 9985 of *Lecture Notes in Computer Science*, pages 336–366. Springer, Heidelberg, October / November 2016.

[62] Craig Gentry, Kenny A. Goldman, Shai Halevi, Charanjit S. Jutla, Mariana Raykova, and Daniel Wichs. Optimizing ORAM and using it efficiently for secure computation. In Emiliano De Cristofaro and Matthew K. Wright, editors, *PETS 2013: 13th International Symposium on Privacy Enhancing Technologies*, volume 7981 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Heidelberg, July 2013.

[63] Satrajit Ghosh, Jesper Buus Nielsen, and Tobias Nilges. Maliciously secure oblivious linear function evaluation with constant overhead. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 629–659. Springer, Heidelberg, December 2017.

[64] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th Annual Symposium on Foundations of Computer Science*, pages 464–479. IEEE Computer Society Press, October 1984.

[65] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th Annual ACM Symposium on Theory of Computing*, pages 218–229. ACM Press, May 1987.

[66] Oded Goldreich and Rafail Ostrovsky. Software protection and simulation on oblivious rams. *J. ACM*, 43(3):431–473, May 1996.

[67] S. Dov Gordon, Jonathan Katz, Vladimir Kolesnikov, Fernando Krell, Tal Malkin, Mariana Raykova, and Yevgeniy Vahlis. Secure two-party computation in sublinear (amortized) time. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *ACM CCS 2012: 19th Conference on Computer and Communications Security*, pages 513–524. ACM Press, October 2012.

[68] S. Dov Gordon, Jonathan Katz, and Xiao Wang. Simple and efficient two-server ORAM. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 141–157. Springer, Heidelberg, December 2018.

[69] S. Dov Gordon, Samuel Ranellucci, and Xiao Wang. Secure computation with low communication from cross-checking. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 59–85. Springer, Heidelberg, December 2018.

[70] Chun Guo, Jonathan Katz, Xiao Wang, Chenkai Weng, and Yu Yu. Better concrete security for half-gates garbling (in the multi-instance setting). In *Advances in Cryptology – CRYPTO 2020, Part I*, Lecture Notes in Computer Science. Springer, Heidelberg, August 2020.

[71] Chun Guo, Jonathan Katz, Xiao Wang, and Yu Yu. Efficient and secure multiparty computation from fixed-key block ciphers. In *2020 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 2020.

[72] Koki Hamada, Dai Ikarashi, Koji Chida, and Katsumi Takahashi. Oblivious radix sort: An efficient sorting algorithm for practical secure multi-party computation. Cryptology ePrint Archive, Report 2014/121, 2014. http://eprint.iacr.org/2014/121.

[73] Carmit Hazay, Yuval Ishai, Antonio Marcedone, and Muthuramakrishnan Venkitasubramaniam. LevioSA: Lightweight secure arithmetic computation. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 327–344. ACM Press, November 2019.

[74] Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Actively secure garbled circuits with constant communication overhead in the plain model. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 3–39. Springer, Heidelberg, November 2017.

[75] Carmit Hazay, Peter Scholl, and Eduardo Soria-Vazquez. Low cost constant round MPC combining BMR and oblivious transfer. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 598–628. Springer, Heidelberg, December 2017.

[76] Carmit Hazay, abhi shelat, and Muthuramakrishnan Venkitasubramaniam. Going beyond dual execution: MPC for functions with efficient verification. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 12111 of *Lecture Notes in Computer Science*, pages 328–356. Springer, Heidelberg, May 2020.

[77] Carmit Hazay, Muthuramakrishnan Venkitasubramaniam, and Mor Weiss. The price of active security in cryptographic protocols. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 184–215. Springer, Heidelberg, May 2020.

[78] David Heath and Vladimir Kolesnikov. Stacked garbling for disjunctive zero-knowledge proofs. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 569–598. Springer, Heidelberg, May 2020.

[79] Cheng Hong, Jonathan Katz, Vladimir Kolesnikov, Wen-jie Lu, and Xiao Wang. Covert security with public verifiability: Faster, leaner, and simpler. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 97–121. Springer, Heidelberg, May 2019.

[80] Yan Huang, Jonathan Katz, and David Evans. Efficient secure two-party computation using symmetric cut-and-choose. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 18–35. Springer, Heidelberg, August 2013.

[81] Yan Huang, Jonathan Katz, Vladimir Kolesnikov, Ranjit Kumaresan, and Alex J. Malozemoff. Amortizing garbled circuits. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 458–475. Springer, Heidelberg, August 2014.

[82] Siam Umar Hussain, M. Sadegh Riazi, and Farinaz Koushanfar. Shaip: Secure hamming distance for authentication of intrinsic pufs. *ACM Trans. Des. Autom. Electron. Syst.*, 23(6), December 2018.

[83] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 145–161. Springer, Heidelberg, August 2003.

[84] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 572–591. Springer, Heidelberg, August 2008.

[85] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Secure arithmetic computation with no honest majority. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 294–314. Springer, Heidelberg, March 2009.

[86] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In William Enck and Adrienne Porter Felt, editors, *USENIX Security 2018: 27th USENIX Security Symposium*, pages 1651–1669. USENIX Association, August 2018.

[87] Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018: 25th Conference on Computer and Communications Security*, pages 525–537. ACM Press, October 2018.

[88] Jonathan Katz, Samuel Ranellucci, Mike Rosulek, and Xiao Wang. Optimizing authenticated garbling for faster secure two-party computation. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 365–391. Springer, Heidelberg, August 2018.

[89] Marcel Keller, Emmanuela Orsini, Dragos Rotaru, Peter Scholl, Eduardo Soria-Vazquez, and Srinivas Vivek. Faster secure multi-party computation of AES and DES using lookup tables. In Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi, editors, *ACNS 17: 15th International Conference on Applied Cryptography and Network Security*, volume 10355 of *Lecture Notes in Computer Science*, pages 229–249. Springer, Heidelberg, July 2017.

[90] Marcel Keller, Emmanuela Orsini, and Peter Scholl. MASCOT: Faster malicious arithmetic secure computation with oblivious transfer. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pages 830–842. ACM Press, October 2016.

[91] Marcel Keller, Valerio Pastro, and Dragos Rotaru. Overdrive: Making SPDZ great again. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 158–189. Springer, Heidelberg, April / May 2018.

[92] Marcel Keller and Peter Scholl. Efficient, oblivious data structures for MPC. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 506–525. Springer, Heidelberg, December 2014.

[93] Marcel Keller, Peter Scholl, and Nigel P. Smart. An architecture for practical actively secure MPC with dishonest majority. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013: 20th Conference on Computer and Communications Security*, pages 549–560. ACM Press, November 2013.

[94] Marcel Keller and Avishay Yanai. Efficient maliciously secure multiparty computation for RAM. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 91–124. Springer, Heidelberg, April / May 2018.

[95] Vladimir Kolesnikov, Jesper Buus Nielsen, Mike Rosulek, Ni Trieu, and Roberto Trifiletti. DUPLO: Unifying cut-and-choose for garbled circuits. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 3–20. ACM Press, October / November 2017.

[96] Vladimir Kolesnikov, Mike Rosulek, and Ni Trieu. Swim: Secure wildcard pattern matching from ot extension. In *International Conference on Financial Cryptography and Data Security*, pages 222–240. Springer, 2018.

[97] Vladimir Kolesnikov, Mike Rosulek, Ni Trieu, and Xiao Wang. Scalable private set union from symmetric-key techniques. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part II*, volume 11922 of *Lecture Notes in Computer Science*, pages 636–666. Springer, Heidelberg, December 2019.

[98] Benjamin Kreuter, abhi shelat, and Chih-Hao Shen. Billion-gate secure computation with malicious adversaries. In Tadayoshi Kohno, editor, *USENIX Security 2012: 21st USENIX Security Symposium*, pages 285–300. USENIX Association, August 2012.

[99] Yehuda Lindell. General composition and universal composability in secure multiparty computation. *Journal of Cryptology*, 22(3):395–428, July 2009.

[100] Yehuda Lindell. Fast cut-and-choose based protocols for malicious and covert adversaries. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 1–17. Springer, Heidelberg, August 2013.

[101] Yehuda Lindell and Ariel Nof. A framework for constructing fast MPC over arithmetic circuits with malicious adversaries and an honest-majority. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 259–276. ACM Press, October / November 2017.

[102] Yehuda Lindell and Benny Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 52–78. Springer, Heidelberg, May 2007.

[103] Yehuda Lindell and Benny Pinkas. Secure two-party computation via cut-and-choose oblivious transfer. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 329–346. Springer, Heidelberg, March 2011.

[104] Yehuda Lindell, Benny Pinkas, Nigel P. Smart, and Avishay Yanai. Efficient constant round multiparty computation combining BMR and SPDZ. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 319–338. Springer, Heidelberg, August 2015.

[105] Yehuda Lindell and Ben Riva. Cut-and-choose Yao-based secure computation in the online/offline and batch settings. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 476–494. Springer, Heidelberg, August 2014.

[106] Yehuda Lindell and Ben Riva. Blazing fast 2PC in the offline/online setting with security for malicious adversaries. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015: 22nd Conference on Computer and Communications Security*, pages 579–590. ACM Press, October 2015.

[107] Yehuda Lindell, Nigel P. Smart, and Eduardo Soria-Vazquez. More efficient constant-round multi-party computation from BMR and SHE. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B: 14th Theory of Cryptography Conference, Part I*, volume 9985 of *Lecture Notes in Computer Science*, pages 554–581. Springer, Heidelberg, October / November 2016.

[108] Chang Liu, Xiao Shaun Wang, Kartik Nayak, Yan Huang, and Elaine Shi. ObliVM: A programming framework for secure computation. In *2015 IEEE Symposium on Security and Privacy*, pages 359–376. IEEE Computer Society Press, May 2015.

[109] Steve Lu and Rafail Ostrovsky. Distributed oblivious RAM for secure two-party computation. In Amit Sahai, editor, *TCC 2013: 10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 377–396. Springer, Heidelberg, March 2013.

[110] Wen-jie Lu and Jun Sakuma. More practical privacy-preserving machine learning as a service via efficient secure matrix multiplication. In *Proceedings of the 6th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pages 25–36, 2018.

[111] Atul Luykx, Bart Mennink, and Kenneth G. Paterson. Analyzing multi-key security degradation. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 575–605. Springer, Heidelberg, December 2017.

[112] Eleftheria Makri and Tim Wood. Full-threshold actively-secure multiparty arithmetic circuit garbling. Cryptology ePrint Archive, Report 2019/1098, 2019. https://eprint.iacr.org/2019/1098.

[113] Carrie Menkel-Meadow. Whose dispute is it anyway: A philosophical and democratic defense of settlement (in some cases). *Geo LJ*, 83:2663, 1994.

[114] John C Mitchell and Joe Zimmerman. Data-oblivious data structures. In *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[115] Payman Mohassel and Yupeng Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy*, pages 19–38. IEEE Computer Society Press, May 2017.

[116] Kartik Nayak, Xiao Shaun Wang, Stratis Ioannidis, Udi Weinsberg, Nina Taft, and Elaine Shi. GraphSC: Parallel secure computation made easy. In *2015 IEEE Symposium on Security and Privacy*, pages 377–394. IEEE Computer Society Press, May 2015.

[117] Jesper Buus Nielsen, Peter Sebastian Nordholt, Claudio Orlandi, and Sai Sheshank Burra. A new approach to practical active-secure two-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 681–700. Springer, Heidelberg, August 2012.

[118] Jesper Buus Nielsen and Claudio Orlandi. LEGO for two-party secure computation. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 368–386. Springer, Heidelberg, March 2009.

[119] Jesper Buus Nielsen and Claudio Orlandi. Cross and clean: Amortized garbled circuits with constant overhead. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B: 14th Theory of Cryptography Conference, Part I*, volume 9985 of *Lecture Notes in Computer Science*, pages 582–603. Springer, Heidelberg, October / November 2016.

[120] Jesper Buus Nielsen, Thomas Schneider, and Roberto Trifiletti. Constant round maliciously secure 2PC with function-independent preprocessing using LEGO. In *ISOC Network and Distributed System Security Symposium – NDSS 2017*. The Internet Society, February / March 2017.

[121] Valeria Nikolaenko, Stratis Ioannidis, Udi Weinsberg, Marc Joye, Nina Taft, and Dan Boneh. Privacy-preserving matrix factorization. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013: 20th Conference on Computer and Communications Security*, pages 801–812. ACM Press, November 2013.

[122] Rafail Ostrovsky and Victor Shoup. Private information storage (extended abstract). In *29th Annual ACM Symposium on Theory of Computing*, pages 294–303. ACM Press, May 1997.

[123] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, Heidelberg, August 2008.

[124] Nicholas Pippenger and Michael J Fischer. Relations among complexity measures. *Journal of the ACM (JACM)*, 26(2):361–381, 1979.

[125] Anjana Rajan. Callisto: A cryptographic approach to #metoo. Burlingame, CA, January 2019. USENIX Association. https://mycallisto.org/assets/docs/callisto-cryptographic-approach.pdf.

[126] Peter Rindal and Mike Rosulek. Faster malicious 2-party secure computation with online/offline dual execution. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016: 25th USENIX Security Symposium*, pages 297–314. USENIX Association, August 2016.

[127] F.E.A. Sander. *Varieties of Dispute Processing*. West Publishing Company, 1976.

[128] Phillipp Schoppmann, Lennart Vogelsang, Adrià Gascón, and Borja Balle. Secure and scalable document similarity on distributed databases: Differential privacy to the rescue. *Proceedings on Privacy Enhancing Technologies*, 2020(2):209 – 229, 2020.

[129] abhi shelat and Chih-Hao Shen. Two-output secure computation with malicious adversaries. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 386–405. Springer, Heidelberg, May 2011.

[130] abhi shelat and Chih-Hao Shen. Fast two-party secure computation with minimal assumptions. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013: 20th Conference on Computer and Communications Security*, pages 523–534. ACM Press, November 2013.

[131] Elaine Shi, T.-H. Hubert Chan, Emil Stefanov, and Mingfei Li. Oblivious RAM with $O((\log N)^3)$ worst-case cost. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 197–214. Springer, Heidelberg, December 2011.

[132] Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. Medians and beyond: new aggregation techniques for sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 239–249, 2004.

[133] Emil Stefanov, Marten van Dijk, Elaine Shi, Christopher W. Fletcher, Ling Ren, Xiangyao Yu, and Srinivas Devadas. Path ORAM: an extremely simple oblivious RAM protocol. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013: 20th Conference on Computer and Communications Security*, pages 299–310. ACM Press, November 2013.

[134] Justin Wagner, Joseph N Paulson, Xiao Wang, Bobby Bhattacharjee, and Hector Corrada Bravo. Privacy-Preserving Microbiome Analysis Using Secure Computation. In *Bioinformatics*, 2016.

[135] Ryan Wails, Aaron Johnson, Daniel Starin, Arkady Yerukhimovich, and S. Dov Gordon. Stormy: Statistics in tor by measuring securely. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 615–632. ACM Press, November 2019.

[136] Liang Wang, Gilad Asharov, Rafael Pass, Thomas Ristenpart, and abhi shelat. Blind certificate authorities. In *2019 IEEE Symposium on Security and Privacy*, pages 1015–1032. IEEE Computer Society Press, May 2019.

[137] Xiao Wang, T.-H. Hubert Chan, and Elaine Shi. Circuit ORAM: On tightness of the Goldreich-Ostrovsky lower bound. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015: 22nd Conference on Computer and Communications Security*, pages 850–861. ACM Press, October 2015.

[138] Xiao Wang, Alex J. Malozemoff, and Jonathan Katz. EMP-toolkit: Efficient MultiParty computation toolkit. https://github.com/emp-toolkit, 2016.

[139] Xiao Wang, Alex J. Malozemoff, and Jonathan Katz. Faster secure two-party computation in the single-execution setting. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 399–424. Springer, Heidelberg, April / May 2017.

[140] Xiao Wang, Samuel Ranellucci, and Jonathan Katz. Authenticated garbling and efficient maliciously secure two-party computation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 21–37. ACM Press, October / November 2017.

[141] Xiao Wang, Samuel Ranellucci, and Jonathan Katz. Global-scale secure multiparty computation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 39–56. ACM Press, October / November 2017.

[142] Xiao Shaun Wang, S. Dov Gordon, Allen McIntosh, and Jonathan Katz. Secure computation of MIPS machine code. In Ioannis G. Askoxylakis, Sotiris Ioannidis, Sokratis K. Katsikas, and Catherine A. Meadows, editors, *ESORICS 2016: 21st European Symposium on Research in Computer Security, Part II*, volume 9879 of *Lecture Notes in Computer Science*, pages 99–117. Springer, Heidelberg, September 2016.

[143] Xiao Shaun Wang, Yan Huang, T.-H. Hubert Chan, abhi shelat, and Elaine Shi. SCORAM: Oblivious RAM for secure computation. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 2014: 21st Conference on Computer and Communications Security*, pages 191–202. ACM Press, November 2014.

[144] Xiao Shaun Wang, Yan Huang, Yongan Zhao, Haixu Tang, XiaoFeng Wang, and Diyue Bu. Efficient genome-wide, privacy-preserving similar patient query based on private edit distance. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015: 22nd Conference on Computer and Communications Security*, pages 492–503. ACM Press, October 2015.

[145] Xiao Shaun Wang, Kartik Nayak, Chang Liu, T.-H. Hubert Chan, Elaine Shi, Emil Stefanov, and Yan Huang. Oblivious data structures. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 2014: 21st Conference on Computer and Communications Security*, pages 215–226. ACM Press, November 2014.

[146] Chenkai Weng, Kang Yang, Jonathan Katz, and Xiao Wang. Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. Cryptology ePrint Archive, Report 2020/925, 2020. https://eprint.iacr.org/2020/925.

[147] Kang Yang, Xiao Wang, and Jiang Zhang. More efficient MPC from improved triple generation and authenticated garbling. In *ACM CCS 20: 27th Conference on Computer and Communications Security*. ACM Press, 2020.

[148] Kang Yang, Chenkai Weng, Xiao Lan, Jiang Zhang, and Xiao Wang. Ferret: Fast Extension for coRRElated oT with small communication. In *ACM CCS 20: 27th Conference on Computer and Communications Security*. ACM Press, 2020.

[149] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science*, pages 162–167. IEEE Computer Society Press, October 1986.

[150] Samee Zahur and David Evans. Circuit structures for improving efficiency of security and privacy tools. In *2013 IEEE Symposium on Security and Privacy*, pages 493–507. IEEE Computer Society Press, May 2013.

[151] Samee Zahur, Mike Rosulek, and David Evans. Two halves make a whole - reducing data transfer in garbled circuits using half gates. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 220–250. Springer, Heidelberg, April 2015.

[152] Samee Zahur, Xiao Shaun Wang, Mariana Raykova, Adria Gascón, Jack Doerner, David Evans, and Jonathan Katz. Revisiting square-root ORAM: Efficient random access in multi-party computation. In *2016 IEEE Symposium on Security and Privacy*, pages 218–234. IEEE Computer Society Press, May 2016.

[153] Greg Zaverucha, Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Jonathan Katz, Xiao Wang, and Vladmir Kolesnikov. Picnic. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[154] Fan Zhang, Sai Krishna Deepak Maram, Harjasleen Malvai, Steven Goldfeder, and Ari Juels. Deco: Liberating web data using decentralized oracles for tls. 2020.

[155] Ruiyu Zhu, Darion Cassel, Amr Sabry, and Yan Huang. NANOPI: Extreme-scale actively-secure multi-party computation. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018: 25th Conference on Computer and Communications Security*, pages 862–879. ACM Press, October 2018.