Final Project Discussion
CS 598 DH
Today’s objectives

Survey possible topics for your project

Form a team

Start looking through the literature
**Setting**

- Semi-honest Security
- Malicious Security
- Zero Knowledge

**General-Purpose Tools**

- GMW Protocol
  - Multi-party
  - Multi-round
- Garbled Circuit
  - Constant Round
  - Two Party

**Primitives**

- Oblivious Transfer
- Pseudorandom functions/encryption
- Commitments
"Are there other security models?"

---

**Efficient Secure Multiparty Computation with Identifiable Abort**

Casper Bauml\(^1\), Emmanuela Orsini\(^2\), and Peter Seif\(^2\)

\(^1\) Department of Computer Science, Aarhus University
\(^2\) Department of Computer Science, University of Bristol
[Bauml@cs.au.dk, Peter.Seif@bristol.ac.uk]

**Abstract.** We study secure multiparty computation (MPC) in the dishonest majority setting providing security with identifiable abort, where if the protocol aborts, the honest parties can agree upon the identity of an aborting party. All known constructions that achieve this notion require expensive zero-knowledge techniques to obtain active security, so as are not practical.

In this work, we present the first efficient MPC protocol with identifiable abort. Our protocol has an information-theoretic online phase with message complexity $O(n^2)$ for each secure multiplication (where $n$ is the number of parties), similar to the RDOZ protocol (Bonodin et al., Eurocrypt 2011), and a factor in the security parameter lower than the identifiable abort protocol of Ishai et al. (Crypto 2014). A key component of our protocol is a linear homomorphic information-theoretic signature scheme, for which we provide the first definitions and constructions based on a previous non-homomorphic scheme. We then show how to implement the preprocessing for our protocol using somewhat homomorphic encryption, similar to the SPDZ protocol (Damgård et al., Crypto 2013) and other recent works with efficient homomorphic encryptions.

---

**Guaranteed Output Delivery Comes Free in Honest Majority MPC**

Vipul Goyal\(^3\), Yifan Song\(^4\), and Chunhai Zhu\(^2\)

\(^3\) Algoresy Founcton, New York, USA
\(^4\) University of Pennsylvania, Philadelphia, USA

**Abstract.** We study the output delivery over point-to-point channels in an honest majority setting. We show that a $2$-round protocol for output delivery is possible when each party has a unique identifier and is communication-efficient. Our construction is based on the Algorand blockchain and the DF-SC scheme of Doron and Ishai (Crypto 2017). We also construct a protocol for output delivery in a more general setting where not all parties are honest, assuming that a subset of the parties are honest and control the network. Our construction is based on standard assumptions such as the existence of a one-way function and a strong extractor.

---

**Complete Fairness in Secure Two-Party Computation**

S. Bow Gordon\(^5\), Carlini Hare\(^5\), Jonathan Katz\(^5\), Yuval Lindell\(^5\)

\(^5\) University of Maryland, College Park, USA

**Abstract.** In the setting of secure two-party computation, two mutually distrusting parties wish to compute some function of their inputs while protecting the privacy of their inputs. Two security guarantees that are important in this setting are fairness and perfect security. Fairness guarantees that if one party receives its output, then the other party also receives its output. In this work, we explore the relationship between fairness and perfect security in the setting of secure two-party computation. We show that, under certain assumptions, fairness implies perfect security. We also show that, in general, fairness does not imply perfect security.
“Are there other security properties/models?”

- Fairness
- Identifiable Abort
- Guaranteed Output Delivery
- Adaptive Security
- Covert Security
- Asynchronous Networks
- Publicly Verifiable Covert Security
- …
“combine GMW and GC?”

ABY – A Framework for Efficient Mixed-Protocol Secure Two-Party Computation

Daniel Demmler, Thomas Schneider, Michael Zohner
Engineering Cryptographic Protocols Group
Technische Universität Darmstadt, Germany
{daniel.demmler,thomas.schneider,michael.zohner}@tu-darmstadt.de

Abstract—Secure computation enables mutually distrustful parties to jointly evaluate a function on their private inputs without revealing anything but the function’s output. Generic secure computation protocols in the semi-honest model have been studied extensively and several best practices have evolved.

This paper presents the first implementation of ABY, a network, called ABY, that efficiently combines secure computation schemes based on Arithmetic circuits, Boolean sharing, and Yao’s garbled circuits and that makes available best practice solutions to secure two-party computation. Our framework allows to pre-compute all cryptographic operations and provokes novel, high-efficiency communication between secure computation schemes based on pre-computed oblivious transfer extensions. ABY supports several standard operations and we perform benchmarks on a local network and in a public intercontinental cloud. From our benchmarks we deduce new insights on the efficient design of secure computation protocols, most prominently that oblivious transfer-based multiplications are much more efficient than multiplications based on homomorphic encryption.

We use ABY to construct mixed-protocols for three example applications – private set intersection, homomorphic matching, and modular representation – and show that they are more efficient than using a single protocol.

Keywords—secure two-party computation; mixed-protocol; efﬁcient protocol design

I. INTRODUCTION

Secure computation has come a long way from the first theoretical feasibility results in the eighties [34], [74]. Ever since, several secure computation schemes have been introduced and repeatedly optimized, yielding a large variety of different secure computation protocols and forms for several functions and deployment scenarios. This variety, however, has made the development of efficient secure computation protocols a challenging task for non-experts, who want to choose an efficient protocol for their specific functionality and available resources. Furthermore, since at this point it is unclear which protocol is advantageous in which situation, a developer would first need to prototype each scheme for his specific requirements before he can start implementing the chosen scheme. This task becomes even more tedious, time-consuming, and error-prone, since each secure computation protocol has its own representation in which a functionality has to be described, e.g., Arithmetic vs. Boolean circuits.

The development of efficient secure computation protocols for a particular function and deployment scenario has recently been addressed by IARPA in a request for information (RFI) [46]. Part of the vision that is given in this RFI is the automated generation of secure computation protocols that perform well for novel applications and that can be used by a non-expert in secure computation. Several tools, e.g., [8], [13], [14], [36], [48], [53], [66], [78], have started to bring this vision towards reality by introducing an abstract language that is compiled into a protocol representation, thereby enabling a developer from having to specify the functionality in the protocol’s (often complex) underlying representation. Those languages and compilers, however, are often tailored to one particular secure computation protocol and translate programs directly into the protocol’s representation. The efficiency of protocols that are generated by these compilers is hence bounded by the possibility to efficiently represent the function in the particular representation, e.g., multiplication of two 64-bit numbers has a very large Boolean circuit representation of size $O(64^2)$.

To overcome the dependence on an efficient function representation and to improve efficiency, several works proposed to mix secure computation protocols based on homomorphic encryption with Yao’s garbled circuits protocol, e.g., [3], [10], [34], [38], [39], [40], [46], [59], [68], [71]. The general idea behind such mix-protocols is to evaluate operations that have an efficient representation as an Arithmetic circuit (i.e., additions and multiplications) using homomorphic encryption algorithms, and operations that have an efficient representation as a Boolean circuit (e.g., comparators) using Yao’s garbled circuits. Those previous works show that using a mixed-protocol approach can result in better performance than using only a single protocol. Several tools have been developed for designing mixed-protocols, e.g., [11], [12], [23], [72], which allow the developer to specify the functionality and the assignment of operations to secure computation protocols. The assignment can even be done automatically as shown recently in [46]. However, since the conversion between homomorphic encryption and Yao’s garbled circuits protocol is relatively expensive and the performance of homomorphic encryption scales very poorly with increasing security parameter, these mixed-protocols achieve only relatively small run-time improvements over using
Round-Optimal Black-Box MPC
in the Plain Model

Yosay Ishai*  Dakshita Khurana†  Ankit Sahai‡  Abhishek Srinivasan§

Abstract

We give the first construction of a fully black-box round-optimal secure multiparty computation (MPC) protocol in the plain model. Our protocol makes black-box use of a sub-exponentially secure two-message statistical sender private oblivious transfer (S2O-OT), which in turn can be based on (sub-exponential variants of) almost all of the standard cryptographic assumptions known to imply public-key cryptography.

1 Introduction

The exact round complexity of secure computation has been a focus of research in cryptography over the past two decades. This has been especially well-studied in the synchronous setting in the plain model, with up to all-but-one static malicious corruptions. It is known that general-purpose secure multiparty computation (MPC) protocols in this setting admitting a black-box simulator require at least 3 rounds of simultaneous exchange [KFR06] [KOS06] [CMPP10]. In this work we focus on MPC with black-box simulation. On the positive side, there has been a long sequence of works [GMPP16, HHP+17, PS17, BEH+17, BEH+18, CCS19] improving the round complexity, culminating in a round-optimal construction that relies on the minimal assumption that a 4-round malicious-secure OT protocol exists [CSS19].

Black-Box Use of Cryptography. Notably, all MPC protocols discussed above make non-black-box use of cryptography, which is typically associated with significant overheads in efficiency. It is interesting, from both a theoretical and a practical perspective, to realize fully black-box protocols [RTV04] where not only does the simulator make black-box use of an adversary, but also the construction itself can be fully specified given just oracle access to the input-output relation of the underlying cryptographic primitives, and without being given any explicit representation of those primitives. In the following, we refer to this standard notion of fully black-box protocols as simply black-box protocols. The focus of this work is on the following natural question:

What is the round complexity of black-box MPC in the plain model?

*Technion. Email: yosi@cs.technion.ac.il
†UIUC. Email: dakshita@illinois.edu
‡UCLA. Email: ankit@cs.ucla.edu
§IIT, Institute of Fundamental Research. Email: abhishek.sinivasan@iitr.ernet.in

The disclosure soundness exchange means that once the round, every party can read a message on a broadcast, and hence the only problem is that all parties can read a message on a broadcast.

“how many rounds are needed?”
"malicious garbled circuits?"
Multiparty Computation from Somewhat Homomorphic Encryption

Ivan Damgård1, Vincenzo Peri1, Niki Sharma2, and Sarah Zakyrić1

1 Department of Computer Science, Aarhus University
2 Department of Computer Science, Bristol University

Abstract. We propose a general multiparty computation protocol secure against an active adversary corrupting up to 2/3 of the n players. The protocol may be used to compute securely arithmetic circuits over any finite field \( \mathbb{F}_p \). Our protocol consists of a preprocessing phase that is both independent of the function to be computed and of the inputs, and a much more efficient online phase where the actual computation takes place. The online phase is unconditionally secure and has total computational (and communication) complexity linear in n, the number of players, where earlier work was quadratic in n. Moreover, the work done by each player is only a small constant factor larger than what one would need to compute the circuit in the clear. We show this is optimal for computation in large fields. In practice, for 3 players, a secure 64-bit multiplication can be done in 4.16 ms. Our preprocessing is based on a somewhat homomorphic cryptosystem. We extend a scheme by Hazay et al. so that we can perform distributed decryption and handle many values in parallel in one oblivious transfer. The computational complexity of our preprocessing phase is determined by the public-key operations, where \( 2(n-1) \) operations per secure multiplication, where \( n \) is a parameter that increases with the security parameter of the cryptosystem. Earlier work in this model needed \( 2(\ell n^2) \) operations. In practice, the preprocessing requires a secure 64-bit multiplication for 3 players in about 13 ms.

1 Introduction

A central problem in theoretical cryptography is that of secure multiparty computation (MPC). In this problem, n parties, holding private inputs \( x_1, \ldots, x_n \), wish to compute a given function \( f(x_1, \ldots, x_n) \). A protocol for doing this securely should be such that honest players get the correct result and this result is the only new information released, even if some subset of the players is corrupted by an adversary. In the case of dishonest majority, where more than half the players are corrupted, unconditionally secure protocols cannot exist. Under computational assumptions, it was shown in \[8\] how to construct UC-secure MPC protocols that handle the case where all but one of the parties are actively corrupted. The public-key machinery one uses for this is typically expensive so efficiency issues are hard to design for dishonest majority. Recently, however, a new approach has been proposed making such protocols more practical. This approach works as follows one

\[ \text{A Full Proof of the BGW Protocol for Perfectly-Secure Multiparty Computation} \]

Glad Asharov1, Yehuda Lindell1

June 12, 2022

Abstract

In the setting of secure multiparty computation, a set of parties wish to compute some function of their inputs. One of the most fundamental results in this computation was proved by Ben-Or, Goldreich and Waksman (BGW) in 1988. Their demonstration that any polynomial functionality can be computed with perfect security in the public channel model. When the adversary is semi-honest this holds as long as \( \lambda \geq \log n \) parties are corrupted. Unfortunately, a full proof of these results was never published. In this paper, we present a full proof of a full proof of the BGW protocol. This includes a full description of the protocol for the malicious setting, including the construction of a new multiplicative protocol for the perfect multiplicative protocol. This requires new for the case of \( \lambda \leq \log n \) parties.

\[ \text{1. Introduction} \]

A central problem in theoretical cryptography is that of secure multiparty computation (MPC). In this problem, n parties, holding private inputs \( x_1, \ldots, x_n \), wish to compute a given function \( f(x_1, \ldots, x_n) \). A protocol for doing this securely should be such that honest players get the correct result and this result is the only new information released, even if some subset of the players is corrupted by an adversary. In the case of dishonest majority, where more than half the players are corrupted, unconditionally secure protocols cannot exist. Under computational assumptions, it was shown in \[8\] how to construct UC-secure MPC protocols that handle the case where all but one of the parties are actively corrupted. The public-key machinery one uses for this is typically expensive so efficiency issues are hard to design for dishonest majority. Recently, however, a new approach has been proposed making such protocols more practical. This approach works as follows one
"Constant-round multiparty protocols?"
Are there other paradigms for secure computation?
Are there special-case problems with interesting solutions?

Efficient Batched Oblivious PRF with Applications to Private Set Intersection

HENNY CHOR
Technion, Haifa, Israel

ODED GOLDBRECHT
Weizmann Institute of Science, Rehovot, Israel

EYAL KUSHILEVITZ
Technion, Haifa, Israel

AND

MADHU SUDAN
Massachusetts Institute of Technology, Cambridge, Massachusetts

Abstract: Publicly accessible databases are an indispensable resource for retrieving up-to-date information. They also pose a significant risk to the privacy of the user, whose personal data is stored in the database. We describe a protocol that enables a user to access data in the database without revealing any information to the database owner.

Efficient Pseudorandom Correlation Generators: Silent OT Extension and More*

Etete Buyi, Godfrey Corront, Xio Gilinav, Yuval Ishai, Lea Hof, and Peter Schind

SoftSpokenOT: Communication-Computation Tradeoffs in OT Extension

Lawrence Boy
February 17, 2023

Extending Oblivious Transfers Efficiently

Yonvil Ishai1, Joe Kilian2, Kobbi Nissim2, and Erez Petrank3

SecureML: A System for Scalable Privacy-Preserving Machine Learning

PITMEM MIHAI1 YUEFEN ZHANG1

Abstract

Machine learning is widely used in practice to produce predictive models for applications such as image processing, text analysis, and recommendation. Recent advances have enabled models that are trained on large amounts of data collected from different sources. However, the massive data collection raises privacy concerns.

In this work, we present a novel and efficient protocol for privacy-preserving machine learning for large input spaces that uses exponential communication networks using the quantum folded network.

Our protocol allows the bounded resources to achieve the privacy goals of the underlying system, which allows data owners to maintain their privacy while still allowing machine learning to proceed.

We implement our system in C++ and our implementation shows that our protocol can be used to achieve secure machine learning in a practical setting.

1 Introduction

Machine learning techniques are widely used in practice to produce predictive models for use in medicine, banking, recommendation systems, sensor networks, and authentication. Large amounts of data collected over time have enabled new solutions to arising problems, and solutions to these problems have led to breakthroughs in speech, image, and text recognition.

Large neural networks suffer from slow training of individual components that predict their future outcomes. Health data from different hospitals, and government organizations can be used to produce new diagnostic models, while financial companies and payment networks can enable transaction history, merchant data, and account holder information to train new accurate models in the future.

While the recent development of efficient computational neural machine learning frameworks and compilers can produce user outcomes, these outcomes are often delivered as black boxes. Comparative advantage, privacy concerns, and regulations, and loosen surrounding data availability and include the protection of many organizations from meaningful data leaks. Privacy-preserving machine learning tools.

*This work is supported by Google postdoctoral fellowship.

1 University of Maryland, College Park, MD 20742, USA.

2 This work was produced while the author was a Visiting Scholar at UC Berkeley.


SecureML: A System for Scalable Privacy-Preserving Machine Learning

PITMEM MIHAI1 YUEFEN ZHANG1

Abstract

Machine learning is widely used in practice to produce predictive models for applications such as image processing, text analysis, and recommendation. Recent advances have enabled models that are trained on large amounts of data collected from different sources. However, the massive data collection raises privacy concerns.

In this work, we present a novel and efficient protocol for privacy-preserving machine learning for large input spaces that uses exponential communication networks using the quantum folded network.

Our protocol allows the bounded resources to achieve the privacy goals of the underlying system, which allows data owners to maintain their privacy while still allowing machine learning to proceed.

We implement our system in C++ and our implementation shows that our protocol can be used to achieve secure machine learning in a practical setting.

1 Introduction

Machine learning techniques are widely used in practice to produce predictive models for use in medicine, banking, recommendation systems, sensor networks, and authentication. Large amounts of data collected over time have enabled new solutions to arising problems, and solutions to these problems have led to breakthroughs in speech, image, and text recognition.

Large neural networks suffer from slow training of individual components that predict their future outcomes. Health data from different hospitals, and government organizations can be used to produce new diagnostic models, while financial companies and payment networks can enable transaction history, merchant data, and account holder information to train new accurate models in the future.

While the recent development of efficient computational neural machine learning frameworks and compilers can produce user outcomes, these outcomes are often delivered as black boxes. Comparative advantage, privacy concerns, and regulations, and loosen surrounding data availability and include the protection of many organizations from meaningful data leaks. Privacy-preserving machine learning tools.
"MPC that goes beyond circuits?"
“More efficient approaches to Zero Knowledge?”

1 INTRODUCTION

Zero knowledge proofs (ZKP) are important for privacy and security in various applications. These proofs enable a prover to convince a verifier that a statement is true without revealing any additional information. In this paper, we explore more efficient approaches to Zero Knowledge, focusing on post-quantum signatures and improved Non-Interactive Zero Knowledge with applications to post-quantum signatures.

Ligero: Lightweight Sublinear Arguments Without a Trusted Setup

We introduce Ligero, a new framework for constructing sublinear zero-knowledge arguments without relying on a trusted setup. Ligero is designed to be efficient and scalable, making it suitable for applications in post-quantum cryptography.

Further Research...

QuickSilver: Efficient and Affordable Zero-Knowledge Proofs for Circuits and Polynomials over Any Field

QuickSilver is a new protocol that enables efficient zero-knowledge proofs for circuits and polynomials over arbitrary fields. This protocol is designed to be both fast and scalable, making it suitable for a wide range of applications.

Mac'n'Cheese: Zero-Knowledge Proofs for Boolean and Arithmetic Circuits with Nested Disjunctions

Mac'n'Cheese is a novel approach to constructing zero-knowledge proofs for Boolean and arithmetic circuits with nested disjunctions. This protocol offers improved efficiency and privacy guarantees compared to previous methods.

Conclusion

In summary, we have presented several new approaches to constructing more efficient zero-knowledge proofs. These techniques open up new possibilities for privacy-preserving applications in the post-quantum era.
“What is the state-of-the-art in garbled circuits?”
“What tools exist for using MPC?”

SoK: General Purpose Compilers for Secure Multi-Party Computation

Marcello Heremann, Brett Henderson, Daniel Nahr, and Steve Zdonik
University of Pennsylvania
{mbhd, bhe, dbnh, sze} @penn.edu

Abstract—Secure multi-party computation (MPC) allows a group of mutually distrustful parties to compute a joint function on their inputs without revealing any information beyond the result of the computation. This type of computation is extremely powerful and has applications in academia, industry, and government. However, secure computation is slow, and when compared to non-secure computation, it is significantly slower. In recent years, significant progress has been made in creating MPC algorithms that are faster and more efficient. However, despite the advances, MPC remains a complex and challenging area. In this survey, we present a comprehensive overview of general purpose compilers for secure multi-party computation. We then provide high-level observations to describe the current state of practice and discuss several major limitations. We conclude with a discussion of the challenges and potential solutions for the future.
Today’s objectives

Survey possible topics for your project

Form a team

Start looking through the literature