The Multi-Facets of MPC

(secure multi-party computation)

Benny Pinkas Aptos Labs Bar-Ilan University NIST WPEC,September 2024



The Early History

WPEC

Mental Poker

by Adi Shamir, Ronald L. Rivest, and Leonard M. Adleman MIT Cambridge, Massachusetts 02139 November 29, 1978

Abstract

Can two potentially dishonest players play a fair game of poker without using any cards (e.g. over the phone)?

This paper provides the following answers:

(1) No. (Rigorous mathematical proof supplied.)

(1) Yes. (Correct & complete protocol given.)



"How can one allow only authorized actors to have access to certain information while not using a trusted arbiter?"

Yao's Millionaires Problem (1982)

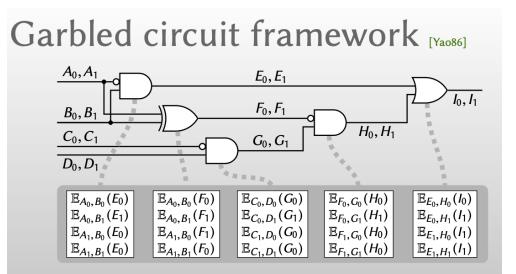






Yao's Millionaire's Problem

- A. Yao, Protocols for Secure Computation. 23th FOCS, 1982.
- A. Yao, How to Generate and Exchange Secrets. 27th FOCS, 1986.
- Garbled (or, encrypted) circuits



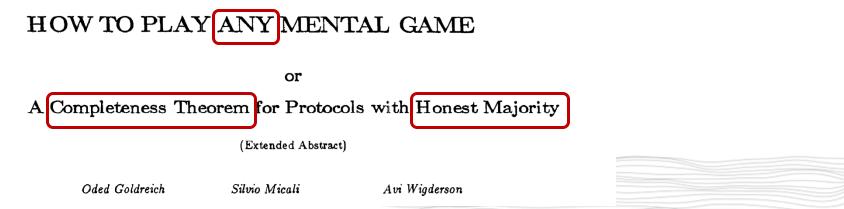
Drawing by Mike Rosulek

Yao gave a solution for Generic MPC

Translate **any** program to a logic circuit, apply MPC transformation to circuit

Two major approaches

- Garbled circuits [Yao, BMR]: low latency, high communication per gate
- Shared inputs [GMW, BGW, CCD]: higher latency, less communication



Multi-party computation

 X_5

 X_4

Wish to compute $F(X_1, X_2, X_3, X_4, X_5)$

 X_2

 $F(X_1, X_2, X_3, X_4, X_5)$

 X_3

 X_1

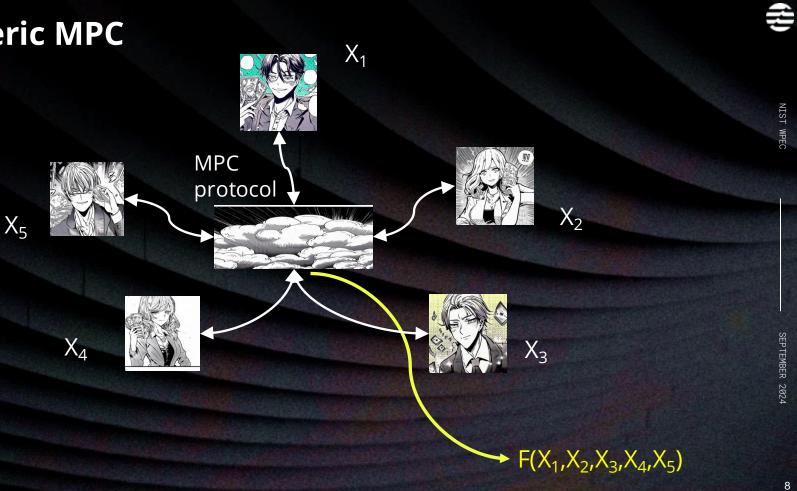
Trusted party

(})

NIST WPEC

7

Generic MPC



Threshold cryptography

How to Share a Function Securely

Alfredo De Santis" Yvo Desmedt[†]

YAIR FRANKEL[‡] MOTI YUNG[§]

(extended summary)

Abstract

1 Introduction

We define the primitive of *function sharing*, a functional analog of secret sharing, and employ it to construct novel cryptosystems.

The basic idea of function sharing is to split a hard to compute (trapdoor) function into shadow functions (or share-functions). The intractable function becomes easy to compute at a given point value when given any threshold (at least t out of l) of shadow functions evaluations at that point. Otherwise, the function remains hard. Furthermore, the function must remain intractable even after exposing up to t-1 shadow functions and exposing values of all shadow functions at polynomially many inputs.

The primitive enables the distribution of the power to perform cryptography (signature, decryption, etc.) to agents. This enables the design of various novel cryptosystems with improved integrity, availability and security properties. As cryptographic techniques mature and security applications grow, the fundamental need to distribute "cryptographic capabilities" amongst a multitude of agents (rather than relying on a single agent) has been recognized. A multi-agent operation, as typically required, should be based simply on outputs of a quorum (threshold) of agents which can be easily combined to the desired output. The combination should anot require security constraints (such as private or oblivioustransfer channels) beyond what is required from a single agent, and should be efficient. It should produce a cryptographic result (i.e., a publicly available output such as a signature, or a private output such as a decryption value of a publicly available ciphertext). Various heuristics have been employed to assure distribution of power, but no systematic secure methods have been developed for this set of tasks.

In this work we suggest a basic primitive which enables rigorous sharing of cryptographic power. We call

Information Technology Laboratory

COMPUTER SECURITY RESOURCE CENTER



Multi-Party Threshold Cryptography MPTC

Signatures, Verifiable PRF, ...



Types of security

- Semi-honest behavior (passive adversaries)
- Malicious behavior (active adversaries)

• Zero-knowledge proofs for a general transformation

Proofs that Yield Nothing But their Validity and a Methodology of Cryptographic Protocol Design



SUMMARY OF OUR RESULTS

Under the assumption that encryption functions exist, we show that all languages in NP have zero-knowledge proofs. That is, it is possible to demonstrate that a CNF formula is satisfiable without revealing any other property of the formula. In particular, without

State of the art in the early 90s:

Many results about feasibility

"No more open questions in cryptography"

My Experience with MPC

Yao's protocol

When discussing auctions, I learned about Yao's protocol from my advisor, Moni Naor (1997)

Wanted to run a sealed-bid auction while hiding the bids even from the auctioneer

<u>Main takeaways:</u>

- Generic MPC is **practical** for some functions (even in python)
- Unrealistic to expect bidders to be online together
- Instead, they share their inputs between two servers that run the MPC

SEPTEMBER 2024

Privacy Preserving Auctions and Mechanism Design*

Moni Naor





Benny Pinkas

Reuben Sumner

Yao's protocol – Interesting discussions

 Summer school on auctions and market design at the Hebrew University (preventing cheating vs. giving incentives)

• "What to compute" vs. "How to compute"

Takeaway

- Ask "customers" to think how they could use a trusted party solution
- MPC can implement the computation without a trusted party (or trusted hardware)
- (but this is the only problem MPC can solve)

Yao's protocol – Proof

• Security proof (with Yehuda Lindell, 2006)

A Proof of Security of Yao's Protocol for Two-Party Computation

Yehuda Lindell*

Benny Pinkas[†]

June 26, 2006

Both works found under-specified issues in the protocols

A Full Proof of the BGW Protocol for Perfectly-Secure Multiparty Computation*

Gilad Asharov[†]

Yehuda Lindell[†]

SEPTEMBER 2024

Fairplay – a system for generic MPC (2003)

Fairplay — A Secure Two-Party Computation System

Dahlia Malkhi¹, Noam Nisan¹, Benny Pinkas², and Yaron Sella¹

- Why implement a system?
- How? Compiler ?!?! Waste of time??
- New issues (cache, memory, O(1) improvements)
- Impact (even though performance was not good)



Table 28. Nine-month Salaries, 11 Responses of 12 US CS Computer Science Departments Ranked 1-12

Faculty Rank	Number of Faculty	Reported Salary Minimum			Overall	Overall	Reported Salary Maximum		
		Minimum	Mean	Maximum	Mean	Median	Minimum	Mean	Maximum
Non-Tenure Teaching Faculty	75	\$37,296	\$53,956	\$76,136	\$71,672	\$71,719	\$66,150	\$88,114	\$110,000
Assistant Professor	118	\$50,000	\$74,711	\$82,000	\$80,891	\$81,357	\$83,200	\$86,483	\$96,000
Associate Professor	86	\$62,995	\$84,148	\$103,000	\$91,412	\$90,847	\$79,300	\$97,949	\$120,000
Full Professor	218	\$51,600	\$88,632	\$109,800	\$122,732	\$116,825	\$139,518	\$168,860	\$198,646

- The first "customer"
- Separation between *input providers* and *computation servers*
- Sorting as the main tool
- Real world requirements: UI, data not arriving, data sanity checks
- Usage...

Secure Computation of Surveys *

Joan Feigenbaum [†]	Benny Pinkas	Raphael S. Ryger [‡]		
Yale University	HP Labs	Yale University		
feigenbaum@cs.yale.edu	benny.pinkas@hp.com	ryger@cs.yale.edu		
	0			

Felipe Saint Jean [§] Yale University felipe.saint-jean@yale.edu

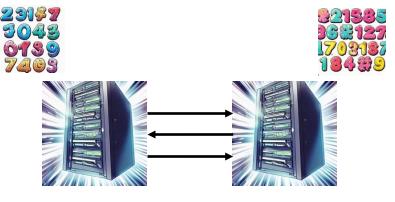
2024

Takeaway

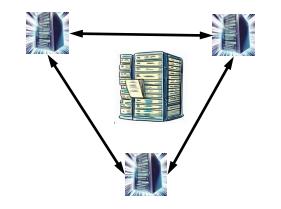
- If something can be efficiently computed using a circuit, it can be efficiently and securely computed using MPC.
- Designing a circuit is *almost* like writing a program
- But functions that do not have an efficient circuit representation might need specific solutions

MPC for Data Analysis

Private Set Intersection (PSI)



Encrypted DB operations (via secure sorting)

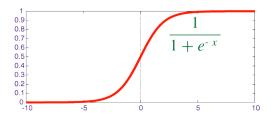


Useful applications, generic solutions are inefficient for these tasks, cool techniques

MPC for machine learning

Applications:

- Train an ML model based on secret data (of many parties)
- Federated Learning
- Google/Meta: Target ads to users without seeing users' data
- Alice has private data, Bob has a secret ML model. Run Bob's model on Alice's data.
- Problem: ML uses non-discrete math
 - Replace with "MPC friendly" quantized models



Conclusions



- The importance of curious theoretical research
- Once researchers realized that MPC can be implemented, many interesting questions came up (same for ZK)
- Threshold cryptography is a major application of MPC

S

Thank you!