Summary

Secure Multiparty Computation: How is it doing in reality?

220031985

The University of St. Andrews

Abstract. The basis for this summary is the evolution of Secure Multipart Computation **(MPC)** from a theoretically based study to a cryptographic technology after being studied for more than three decades and proving a solid theoretical foundation. MPC has been utilised in commercial products because it proved to be effective enough for usage in practice over the last few years. Supplying security and privacy, particularly for data dispersed across vast, distant networks, is difficult. Secure Multipart Computation is one method that offers a solution to the issue of data security and privacy. This essay will summarise MPC's functions, problems, and position in the technological chain.

Keywords. Secure multiparty computation (with the acronym MPC, and sometimes SMPC), Linear Secret Sharing Schemes (LSSSs).

1 Introduction

Utilizing personal data without exposing is a significant difficulty today. Particularly in Distributed Systems, Secure Multiparty Computation might help handle the enormous challenge of sensitive data such as our location, health, and social data. The term "distributed systems" or "distributed computing" describes a situation in which several distinct, connected computer devices are motivated to do a joint computation of any function. These devices may have a distributed database system that performs database updates. Secure Multiparty Computation aims to make it possible for these parties to process distributed computing tasks securely.

Due to the risk of machine failures and unintentional defects, distributed computing is more specifically addressed and explained in this protocol. Secure Multiparty Computation is concerned with the potential for malicious behaviour that is planned out by an adversary entity, also known as **Byzantine Faults**: a protocol execution that might be "attacked" by a third party or even only a part of the parties involved. Such an attack aims to discover privileged information or the outcome of an inaccurate calculation. Any Secure Multiparty Computation protocol must satisfy the two key requirements of privacy and correctness.

- **Privacy Requirements:** Parties should only learn what is essential and not more.
- **Correctness Requirements:** Each party should receive its correct output.

The adversary cannot influence the computation's outcome to vary from the function to which the parties intended to compete.

2 How it Fits!

Numerous issues can be resolved via MPC, allowing data to be used while keeping privacy. Given a scenario, a person's DNA information is very private. Comparing a person's DNA to a database to find whether that person has an elevated risk of cancer could have significant health and societal benefits and should not be shown to a private organisation or parties other than those to whom it is intended. MPC finds a solution to this conundrum by showing the type of cancer the person is most likely to have, or not. The condition for correctness supplies the assurance that the outcome cannot be altered by a malicious party.

Another scenario involves a trading platform where parties make offers and bids, which are matched whenever an offer is higher than a bid. MPC demonstrates its value by keeping opponents' offers and bids a secret, ensuring that just the buyer and seller and the resulting price are shown and that the price revealed is the right one. In any case, MPC guarantees both properties (privacy & correctness) and more.

3 Security of MPC

There are cases when an adversary has some limited power over the parties and looks to undermine the protocol's execution. Corrupted parties are those who have fallen under the sway of the adversary and adhere to their directives. Any secure protocols should withstand such attacks, to formally claim and prove that a protocol is safe, a precise definition of security for MPC is needed.

The following criteria, which are discussed in the paper in the form of an auction scenario, are summarised to set up requirements that ensure many important security protocols that are general enough to encompass most (if not all) multiparty computation tasks:

- **Privacy:** No party should gain knowledge beyond what is required of it. Emphasis should be placed on the knowledge that can be gained from the input of the other parties and can be extracted from the result itself.
- **Correctness:** Each Party is certain that every outcome it receives is correct and truthful.
- **Independence of Inputs:** The inputs chosen by corrupted parties must be distinct from those chosen by honest parties.
- **Guaranteed Output Delivery:** The ability of honest parties to obtain their products without interference from corrupt parties.
- **Fairness:** If only honest parties receive their output, the corrupted parties should also receive theirs.

Note: Guaranteed output delivery implies fairness, the converse is not necessarily true.

3.1 The Definition Paradigm

The criteria should apply to any secure protocol, even though they do not necessarily form a definition of security. While it is possible to define security by simply coming up with a list of distinct needs and declaring that a protocol is safe if all these requirements are met, it may not be possible due to the following conditions.

- Given that different apps have varied requirements, it's possible that a crucial criterion was overlooked.
- Defining should be simple enough so that it is trivial to see all possible adversarial attacks are prevented by the proposed definition.

The **ideal/real simulation paradigm**, which compares the functionality of security measures in an ideal environment to that in the real world, is a special case of the scenario caused by these circumstances.

3.2 Ideal / Real Simulation Paradigm

Imagine an "**ideal world**" where a third party that you can trust and incorruptible is willing to aid the parties in doing their computation chores. In such a world, the parties can easily communicate their inputs to the trusted party, who will then compute the desired function and provide each party with their specified result. The only action a party takes is giving its input to the trustworthy party, and the only flexibility an adversary has is to select the information from the corrupted parties. This ideal computation holds all the described security properties (and more).

In the "**real world,**" there couldn't possibly be external parties that could be trusted by all parties. The parties run some protocols among themselves without help, and some are corrupted and colluded. While a secure protocol should simulate the so-called "ideal world," which presupposes an enemy launching a successful attack in the real world, an adversary launching a successful assault with the same impact does not exist in the ideal world. However, successful adversarial assaults are impossible to execute in an ideal setting, leading to the conclusion that any adversarial attacks on protocol executions in the real world must also be unsuccessful.

For any adversary attacking a real protocol execution, there exists an adversary attacking an ideal execution such that the input/output distributions of the adversary and the participating parties in the real and ideal executions are essentially the same. Concluding that an actual protocol execution "emulates" the ideal world. This security setup is known as **the ideal/real simulation** paradigm, and the following conditions are reached:

- **Privacy:** The adversary's output is the same in real and ideal executions.
- **Correctness:** Honest parties' outputs are the same in the real and ideal executions, and from the fact that in an ideal execution, the honest parties all receive correct inputs as computed by the trusted party.
- **Independence of outputs:** In an ideal execution, all inputs are set to the trusted party before any output is received.
- **Guaranteed Output Delivery and Fairness:** In an ideal world, the trusted party always returns all outputs, and the same holds in the real world.

There is an added definitional parameter, albeit it is not extensively said as in the requirements mentioned above. **Adversarial power** sets up an adversary's capability to attack a protocol's execution. The review highlights two key characteristics that characterise the adversary:

- **Allowed Adversarial behaviour:** Actions that corrupted parties are allowed to take.
    - **Semi-honest adversaries:** Corrupted parties correctly follow protocol specifications.
    - **Malicious adversaries (Active):** Corrupted parties can arbitrarily deviate from the protocol specification according to the adversary's instructions.
    - **Covert Adversaries:** Adversaries that behave maliciously to break the protocol. If undetected, the adversary might successfully cheat.
- **Corruption Strategy:** Deals with the question of when and how parties are corrupted.
    - **Static corruption model:** The set of parties controlled by the adversary is fixed before the protocol begins.
    - **Adaptive corruption model:** Adaptive adversaries are given the power to corrupt parties during computation with a fixed set of corrupted parties. A party that has been corrupted still is corrupt moving forward.
    - **Proactive Security model:** Possibly, parties are corrupted for a certain period only.

4 Modular Sequential and Concurrent Composition

It has been proven that when used in a more extensive system, an MPC protocol runs like that of an incorruptible trusted party performing computation for parties. In actuality, an MPC protocol is not executed in isolation but rather as a system part. *Modular composition* is a powerful theorem that allows for the modular construction of larger protocols using secure sub-protocols and the analysis of larger systems that employ MPC for calculations. In terms of *sequential compositions*, the MPC protocol can run as a subprotocol of another protocol, sending arbitrary pre- and post-MPC messages. This is a stand-alone setting and sets up the basic definition of the security of Canetti (Ran Canetti, 2000).
No matter what other protocols are running simultaneously with it, every protocol that has been proven to be secure by this definition is guaranteed to behave like an ideal execution. It is said to be the MPC definition and regarded as the gold standard, though it has a cost in terms of assumptions and efficiency.

5 Feasibility of MPC

While definitions state that no adversarial success is accepted and that the protocol should act as if a trustworthy party is performing the computation, it is up to the individual to consider if security protocols are even conceivable and, if so, for which distributing tasks. We summarise the centre of these results.

Let n be the total number of parties involved and t represent the upper bound on the potential number of corrupted parties.

- **For t < n/3** (when less than a third of parties can be corrupted), secure multiparty protocols with fairness and guaranteed output delivery can be achieved for any function. This is assuming a point-to-point network with authenticated channels and information-theoretic security.
- **For t < n/2** (in the case of a guaranteed honest majority), secure multi-party cooperation is needed.
- **For t ≥ n/2** Secure multiparty protocols are possible (without fairness or output delivery guarantees; that is when the number of corrupted parties is not constrained).

To sum up, Secure Multiparty protocols are available for all distributed computing tasks. Although the feasibility above results is theoretical, they do not consider actual expenses associated with efficiency.

6 Techniques

The few examples given in the paper are just a handful of the many strategies that have been developed over the past three decades for building MPC protocols with various features. Secret Sharing Scheme is associated with an Access Structure. This is the set of parties that can recover the secret. Access structures are monotonically increasing, i.e., if you add a party to a set that can access the secret, the more extensive set can also access it. A subset of the parties reveals their secret shares for everyone to learn secrets.

Since they enable parties to locally compute linear functions of their secrets for free, so-called Linear Secret Sharing Schemes (LSSSs) are typical of interest in multi-party computation. In Shamir Secret Sharing, the number of parties which can be adversarial t is a threshold of the total number of parties n. In multi-party computation protocols, one usually has either $t < n/2$ or $t < n/3$. Shamir, a type of LSSS, is typically used in basic classes to introduce topics [1].

While most general MOC protocols can be used to compute any function, the function being computed is often represented as a Boolean or arithmetic circuit. The MPC protocol participants are all included in this circuit, and it is expected that they can all safely communicate with one another. The following summarises the paper's argument for the protocol for semi-honest adversaries:

- **Input sharing** – Using Shamir's secret sharing, each party communicates its ideas to the other parties. The secret sharing used is (t+1)-out-of-n, with $t = (n-1)/2$. Supplying security against any minority of corrupted parties.
- **Circuit Evaluation** – In this phase, parties evaluate the circuit one gate at a time from the input gates to the output gates.
- **Output Reconstruction** – Once the parties have obtained shares on the output wires, they can obtain the outputs by simply sending their shares to each other and reconstructing the outputs via interpolation.

This protocol is valid and secure for semi-honest adversaries if fewer than $n/2$ parties are compromised.

While every function can be computed using general secure computing, it often proves to be the most effective method. But occasionally, the unique structure of the problem-solving function allows us to arrive at more expedient, specialised answers. The following examples are given in the paper:

- **Private set intersection** - In a private set intersection protocol, two parties with separate sets of values seek to show their intersection while keeping just the components of their sets secret. Today's most effective private set intersection protocols employ innovative hashing methods and can quickly process millions of items.
- **Threshold cryptography -** With threshold cryptography, several parties can perform cryptographic operations without a single person having access to the secret key.
- **Dishonest-majority MPC -** Different strategies are needed when there is a dishonest majority, including the crucial case of two parties (with one corrupted). According to the study, there has been so much effort done in this area that any attempt to explain it would be extremely unfair.

7 Efficient and Practical Use of MPC

MPC research focuses on what cryptography and setup assumptions are necessary to enable MPC as well as how to define and prove security for various adversarial and network models. This approach's first stages were entirely algorithmic and concentrated on lowering the overhead associated with accommodating the cryptographic primitives. Since then, a significant amount of research has concentrated on increasing MPC's effectiveness. Unlike and/or gates, which have a cost, XOR gates can be computed virtually for nothing. The culmination of these developments has made MPC efficient enough to be applied in practice to a wide range of issues. For instance, several protocols calculate XOR gates virtually for free instead of expensive AND/OR gates.

8 MPC Usage in Reality

Although theoretical research discusses MPC's possible advantages, today's reality is significantly different. MPC is currently utilised in many real-world scenarios, and utilisation is rapidly expanding. The following MPC programmes are active right now.

Boston Wage Gap - The MPC for the Greater Boston area calculated information on the compensation of about 16% of the workforce using data from 166,705 individuals and 114 enterprises. Due to privacy concerns, businesses would not offer raw data; hence the use of MPC was essential. The gender difference in the Boston area is much broad than what the U.S. Bureau of Labor Statistics had previously predicted, according to the findings.

Advertising Conversion - Google calculates the amount of the intersection between a commercial and the list of consumers who have made the promoted purchases. While doing so, only the size of the set intersection is revealed. To compute the intersection, Google and the business paying for the advertisement must supply their respective lists.

MPC for cryptographic key protection – Threshold cryptography enables one to perform cryptographic operations without keeping the private key in an individual location. For the protection of cryptocurrency keys, several businesses are turning to threshold cryptography as an alternative to outdated hardware. A single organisation can create keys and perform cryptographic calculations using MPC without the key being kept in a location where it can be stolen.

Government Collaboration - Information on residents is held by government agencies; by pooling this information, enormous benefits can be realised. The government used MPC to ensure that all tax secrecy and data protection laws were adhered to without sacrificing the usefulness of the data. For instance, Canada cancelled an initiative to aggregate citizen data in 2000 after hearing accusations that they were creating a "big brother database." To figure out if students who work while pursuing their degrees are more likely to fail than those who concentrate only on their studies, Estonia collected encrypted data from higher education and income tax records using MPC.

Privacy-preserving analytics - Machine learning models (growing exponentially) can be applied to data via MPC without showing the identity of the data owner. MPC can also be used for risk score computations, anti-money laundering, and other things.

9 Conclusion

According to the report, Secure multiparty computation is a phenomenal illustration of a research triumph. In the first 20 years of its existence, no applications were anticipated, and the future of MPC was uncertain. However, it has undergone a significant metamorphosis, and MPC has not only improved in speed to the point where it can be utilised in practice but has also gained industry acceptance and evolved into a technology that is used in that setting.

The development of the last few years and the substantial amount of applied research that is currently produced indicate that MPC has a promising future in practice.

References

[1] Shamir Secret Sharing, https://wiki.mpcalliance.org/Shamir%20Secret%20Sharing.html, Accessed 20-10-22