Secure multi-party computation for supply chain collaboration

Author: Danila Romanov
Supervisor: Tianyu Li
Responsible Professor: Zekeriya Erkin

Cyber Security Group
Department of Intelligent Systems
Delft University of Technology

27th of June 2021

Abstract

Despite evidence that collaborating in the supply chain can reduce inefficiency and result in mutual gain, parties do not wish to collaborate if they have to share their private proprietary information. The main reason for their privacy concern is that the party does not want to lose their competitive advantage by giving away company secrets. Collaborative optimization algorithms can be applied to problems in the supply chain, and secure multiparty computation is incorporated as part of the algorithm to preserve the privacy of the parties. This paper aims to create an overview of privacy-preserving applications in the collaborative supply chain by conducting a literary study that focuses on secure collaborative optimization research and its limitations.

Research findings showed that secure multiparty computation can be applied to the following supply chain collaboration problems: capacity sharing, price-masking, distributed scheduling, collaborative production and transport, vehicle routing, and resource allocation. These algorithms use multiparty computation that is secure under the semi-honest adversarial model, because a malicious model is generally too inefficient for practical use. This choice comes with a cost in privacy, as the semi-honest model assumes parties collaborating will not break the protocol. This is a weak assumption that results in an impractical protocol, as real life applications of semi-honest multiparty computation would not be protected against a party that benefits from cheating. Furthermore, secure multiparty computation has a limitation that it cannot prevent a party from lying in its private input.

This paper recommends for future secure collaborative optimization research to combine multiparty computation with game theory. Achieving incentive compatibility in a protocol proves that it is in the best interest for a party to not cheat, as it either leads to a loss in benefits or they are caught. This allows for the MPC protocol to keep its efficiency by being secure under just a semi-honest adversarial model, as well as offer greater protection for honest parties from a rational malicious party.

1 Introduction

Collaboration in the supply chain is an increasingly important research topic that enables superior performance for the parties involved through increased outcomes and improved benefits [1]. One of the greatest issues that is in the way of successful collaboration is trust. As participating parties are not interested in collaborating if they feel that they cannot keep their secret proprietary data safe. Competitors learning this data can lead to a competitive advantage, which is why methods that are able to protect the data of the parties are very valuable [2].

For supply chain collaboration the use of a trusted third party has been tried practically. For example in 2003, Nistevo (a logistics service) gathered logistics information from 24 companies and optimized routing for trucks such that they could stop and be filled by a different company when they were empty [3]. However, a trusted third party is not always ideal and brings multiple problems. Firstly, it requires a high degree of trust from the collaborating parties, which can be difficult to acquire [4]. For example, it would be very difficult for any of the parties to know whether the trusted third party is biased towards a particular party or selling the data it is given. Secondly, if one of the parties decides that they do not trust the third party, the collaboration can get stuck, which results in a loss of efficiency. Lastly, a trusted third party usually charges a steep price for their service. For example Nistevo the company referred to previously, charged one of their participating parties 250,000 dollars per year [3]. Previous and ongoing research aims to address the limitations of trusted third parties by replacing them with secure multiparty computation (MPC), a cryptographic method that allows a function, whose inputs are held by different parties, to be solved in a distributed manner without revealing any of the inputs to each other [4].

This paper will conduct a literary study and summarize the currently existing work on MPC's theoretical applications to the collaborative supply chain. It will show a limitation in the way MPC is being implemented for practical solutions, and finally, show how combining MPC with game theory can result in a better preservation of privacy.

The remainder of this paper is organized as follows. Section 2 will go into more detail on supply chain collaboration, MPC and the importance of privacy. Section 3 will introduce the methodology of the paper and related works. Section 4 will summarize the research on collaborative optimization problems that were solved using MPC and how they could be applied to collaborative supply chain. Section 5 will introduce game theory and show how MPC can be improved with incentive compatibility. Section 6 will be a discussion on ethical implications and reproducibility. Section 7 will discuss results from Section 4 and 5. Finally Section 8 will conclude the paper and recommend ideas for future work.

2 Preliminaries

2.1 Supply Chain Collaboration

To demonstrate the possible benefits of supply chain collaboration, consider this real life scenario. General Mills and Land O' Lakes are two food producing companies who are collaborating through Nistevo's collaborative logistics service. General Mills is delivering goods from New York to Ohio and passing through several stops on the way. Normally, the truck is empty on its way back to New York from Ohio. However with Nistevo, General Mills is directed to stop at a Land O' Lakes warehouse to deliver their goods to New Jersey, which is in the direction the truck is heading. This form of collaboration reduced Land O' Lakes' annual freight costs by 15 percent [3]. This is a significant reduction which is also beneficial for carbon emissions. Figure 1 shows a detailed diagram illustrating the reduction in distance travelled with collaboration versus without collaboration.

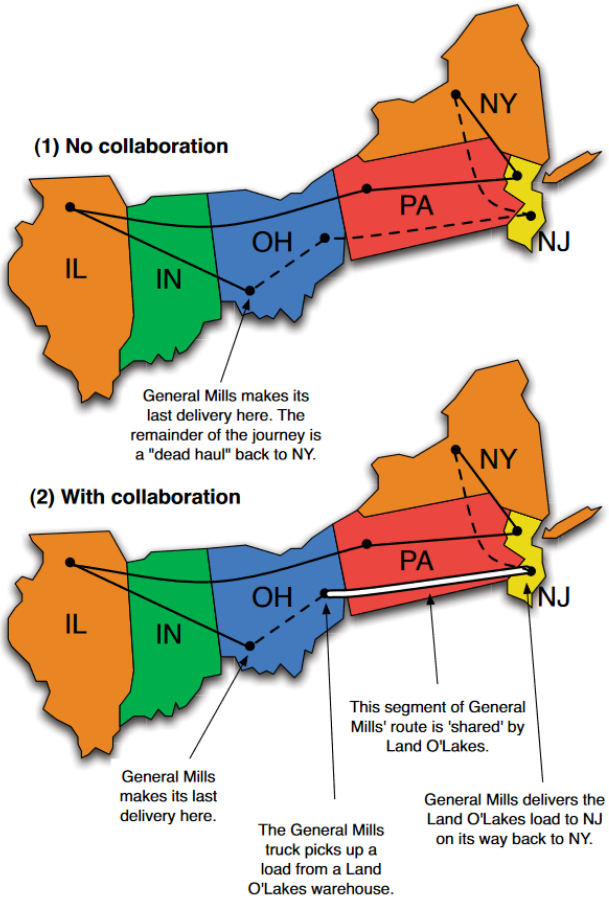

(1) No collaboration
NY
PA
OH
IL
IN
NJ
General Mills makes its last delivery here. The remainder of the journey is a "dead haul" back to NY.
(2) With collaboration
NY
PA
OH
IL
IN
NJ
This segment of General Mills' route is 'shared' by Land O'Lakes.
General Mills makes its last delivery here.
The General Mills truck picks up a load from a Land O'Lakes warehouse.
General Mills delivers the Land O'Lakes load to NJ on its way back to NY.

Figure 1: Diagram showing how collaboration was used to benefit General Mills and Land O' Lakes [2].

Two typical categories exist for supply chain collaboration: horizontal and vertical collaboration. In horizontal collaboration, the parties have identical roles in the supply chain, which means that they are, for example, all suppliers/manufacturers. In vertical collaboration, parties may have different roles, one party being the supplier/manufacturer and the other being a distributor [4]. The above example of collaboration between General Mills and Land O' Lakes, is horizontal collaboration. A similar example can be made for vertical collaboration. If instead of General Mills delivering their own goods, they hired a distributor to perform the task.

Logistics is not the only area where collaboration in the supply chain may occur. For example, two factories producing mugs and bowls both needs to utilize the same limited resource: clay. Both factories also have a limited number of laboring hours and their own unit revenues. The resources required to make a bowl or a mug are given in Table 1. The first factory has 10 hours of labor

Table 1: Example of collaborative production showing unit demands and revenues of two factories producing bowls and mugs.

	Labor (Hours/Unit)	Clay (Pounds/Unit)	Revenue ($/Unit)
Bowl (Factory 1)	1	4	40
Mug (Factory 2)	2	3	50

and 60 pounds of clay to use every day, while the second factory has 30 hours of labor and 60 pounds of clay to use. The revenue maximizing solution for both factories are $400 and $750 dollars respectively. However, if the two factories were to collaborate and optimize their shared resources then their maximum revenue is now $1,360 dollars, which is greater than the sum of the factories' local optimum revenue [4]. This is another example of how collaboration can result in mutual gain for the participating parties.

2.2 Secure multiparty computation

In MPC, two or more parties aim to calculate the output of a function based on their own private inputs without disclosing them. MPC computes the solution to a distributed problem such that in the end only the output is revealed to the parties [5]. In the context of collaborative supply chain, we aim to replace a trusted third party with an MPC protocol to increase privacy protection for the collaborating parties.

The origin of MPC was for the solution to a two-party comparison problem called 'Yao's Millionaire Problem'. Here, two millionaires would like to know which one of them is richer, but not reveal their actual wealth [6]. Goldreich et al. expanded the solution to work for a multi-party scenario and proved that a secure solution exists for any multi-party computation problem that is in polynomial-time [7]. This was achieved by representing the function that is solved, as a Boolean circuit consisting of a series of gates. Each gate is then evaluated securely. Unfortunately, the size of the circuit grows very quickly with the complexity of the function and the number of inputs, which makes it too inefficient for collaborative supply chain problems [4]. Since a secure solution exists, large amount of work goes into creating efficient algorithms instead.

One of MPC's greatest advantages is that a protocol that uses it can be shown to be provably secure [4]. 'Provably secure' systems have the benefit that their security is backed by mathematical proofs as opposed to heuristics [8]. This level of security is dependant on assumptions about the adversary, a malicious party, and which computational resources they have [9]. In MPC, the exchange of messages can be protected under perfect security or computational security. In computational security MPC, the security of the messages exchanged relies on the existence of one-way trapdoor functions [10]. These are functions whose inverse is very difficult to compute without the secret key [9]. In theory, it is possible to discover the key through brute force, but computational security is based on the assumption that the adversary is bounded computationally and cannot discover it in a reasonable amount of time. In perfect security MPC, an adversary is bounded by information. For example, the method of secret sharing is where a number x is broken down such that x cannot be reconstructed unless a certain number of shares is obtained by a party [9]. This is commonly half of the number of available shares. One major limitation of this security model is that it cannot be used for MPC protocols consisting of two parties, as the malicious party will begin with at least half of the shares.

So far we have addressed the security assumptions on the messages being exchanged in MPC, however it is almost important to define what it means for a computation to be secure. The definition that is most commonly used nowadays, relates to the ideal/real simulation paradigm [11]. The ideal model is one where parties give their inputs to a trusted third party, who then computes the function and sends the output to everyone. The real model aims to have equivalent outcomes as the ideal one, but without the use of a trusted third party. Some of the most important characteristics of an MPC protocol is that it ensures correctness and privacy [2]. Correctness is ensured when every party receives the correct output or none at all. Privacy is said to be achieved when no party learns anything more than the output they receive. Finally these characteristics must apply and the protocol must remain secure, even if a few parties 'cheat'.

In MPC there are two main adversarial models, 'semi-honest' and 'malicious'. A semi-honest adversary will follow the protocol correctly, but will attempt to extract any information they can from other parties from the messages exchanged. A malicious adversary may deviate from the protocol, trying to create an incorrect final output or extracting private information [2]. Protocols secure under the malicious model are often quite inefficient for practical use [4].

The first large scale application of MPC occurred in Denmark in 2008, for an online double auction of sugar beet licenses [12]. Farmers were competing for contracts that gave them production rights for a certain amount of beets per year and to deliver them to Danisco. The farmers did not trust Danisco to act as an auctioneer, as that would reveal their economic position and productivity. Danisco also did not trust the farmer's union DKS to be auctioneer either and a trusted third party was found to be an expensive solution. For this reason, it was decided to replace the auctioneer with a MPC protocol. 1200 bidders successfully participated in the auction and the computation lasted 30 minutes, resulting in 25,000 tons of production rights being exchanged. In a survey after the auction, 80% of respondents stated that it was important to them that their bids were kept confidential, and were pleased with the confidentiality that the system offered [12].

2.3 Collaborative Optimization

The computation problems in supply chain collaboration have a common theme, that is they are optimizing a variable, such as maximizing revenue or minimizing costs. For this reason, collaborative optimization research has great applications in supply chain collaboration [4].

Collaborative optimization research is used for distributed problems that have a variable, that need to be optimized and aims to achieve a global objective using only local information [4]. Generally work was mostly focused on increasing efficiency and reducing the cost of communication, rather than privacy, however recently there has been a lot more interest in the area of secure collaborative optimization. To achieve secure protocols, MPC has been one of the methods used to replace the third party that is commonly present in these protocols.

2.4 Importance of privacy for supply chain collaboration

While privacy usually refers to protecting the data of consumers and anonymity, in collaborative supply chain, it means to keep others from learning company secrets. For example, to find the optimal vehicle routing for multiple parties, requires knowing:

- The location of factories and distribution centers
- The amount of product that is delivered to a location

- The total supply of a party that is being distributed

A lot of information can be inferred from this data. For example, the demand for a party's good in a city [4]. This data can be tracked over time, and could be used by a competitor to learn a lot of information about the party. This is more than enough of a reason for many companies to not want to collaborate, as the risk of competitors exploiting their company information is too high [2]. This is why the importance for a privacy-preserving method for supply chain collaboration is substantial.

3 Methodology and Related Work

3.1 Methodology

The research question that this paper will aim to answer is: "How can secure multiparty computation be used to preserve privacy in supply chain collaboration?". The method that will be used is to perform a literary study. Google Scholar will be the literary work search engine that is used. Works from collaboration optimization research where MPC was used will be looked at. There is already lots of existing work on collaboration problems solved using MPC, so it is not necessary to recreate anything. The analysis done by the authors on the privacy of their algorithms will be looked at, as well as other works that cited it that aimed to improve it or show issues. Furthermore, papers illustrating the limitations of practical uses of MPC will also be looked at. The goal is to create an overview of the applications of MPC that could theoretically be used in the supply chain, and to gather what issues there are that could potentially hinder it from being used in the real world.

3.2 Related Work

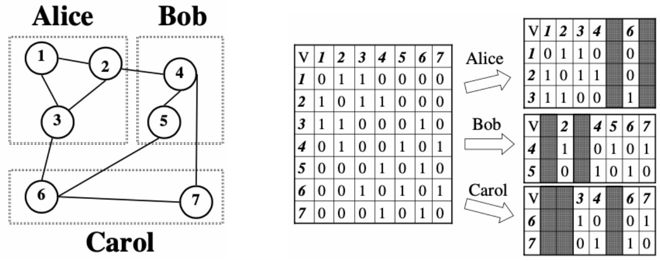
The number of literary studies on the applications of MPC in collaborative supply chain is limited. The biggest and most recent is from 2013, where Hong et al. performed a survey on privacy-preserving methods in collaborative supply chain. As well as MPC, they also looked at secure transformation, which is a different method for secure computation. One of the disadvantages with the method is that it is not provably secure and its security is based on heuristic reasoning [4]. The literary study was performed by gathering methods from collaborative optimization research, and identifying whether privacy protection was integrated. This is a similar approach as to this paper, however a lot of these privacy-preserving methods are very old or do not have any privacy. It is the aim of this paper to find the newer developments in this area of research, and to focus on what is possible with methods that do protect privacy and determine their limitations.

4 Applications of Secure Collaborative Optimization in Supply Chain Collaboration

Previously, it was stated how secure collaborative optimization can be used to solve distributed problems in the collaborative supply chain. This section will introduce their different types and show how they can be applied to various collaborative supply chain problems. At the end a table will presented, summarizing this research.

Travelling Salesman Problem (TSP) is a fundamental optimization problem that can be applied to logistics, planning and production. Hong et al. have created an efficient secure communication protocol for this problem under the semi-honest adversarial model [13]. To give an example of how it can be applied, they give a scenario where a client would like to choose between Alice or Bob to ship his goods between certain cities. The problem is that the client would not like to share his list of cities to Alice or Bob before signing the contract, meanwhile, Alice or Bob would not like to share their delivery costs either. The collaborative TSP solution can be applied to this scenario by calculating which distributor is a more optimal choice for the client, without having any of the parties reveal their proprietary data.

Graph Coloring is an NP-complete decision problem that finds a way to color the vertices of a graph under the constraint that adjacent vertices cannot have the same color [14]. Hong et al. have created a solution to the distributed graph coloring problem with the use of privacy-preserving tabu search that is secure under the semi-honest model [14]. Exhaustive search is not practical for graph coloring, due to it being NP-hard, which is why tabu search is used. Tabu search will achieve near optimal solutions with significantly higher performance, allowing it to be used in more practical applications. Distributed graph coloring can be used in distributed scheduling and network allocation. As an example of an application of distributed scheduling: Alice, Bob and Carol have a set of jobs at a location that contain conflicts (some jobs cannot be performed at the same time). To optimize this problem, rather than every party doing their job after each other, the parties can perform some jobs at the same time, as long as the jobs the parties are doing are not conflicting. By gathering the set of all jobs that parties would like to do, an optimal order could be calculated, however collaborating parties would not like to share their sets of jobs. The distributed graph coloring problem can solve this by formulating job collisions as edges and then finding the optimum coloring solution which gives the order in which a party should perform their job. This solution had some privacy leakage, as a party can learn which one of its jobs is conflicting with a party. However this is due to the nature of the problem and not the protocol, as this is something that can be derived from the output. Figure 2 illustrates the problem being formulated as a graph and Figure 3 illustrates the output given to each party following the algorithm.



V	*1*	*2*	*3*	*4*	*5*	*6*	*7*
1	0	1	1	0	0	0	0
2	1	0	1	1	0	0	0
3	1	1	0	0	0	1	0
4	0	1	0	0	1	0	1
5	0	0	0	1	0	1	0
6	0	0	1	0	1	0	1
7	0	0	0	1	0	1	0

V	*1*	*2*	*3*	*4*	▨	*6*	▨
1	0	1	1	0	▨	0	▨
2	1	0	1	1	▨	0	▨
3	1	1	0	0	▨	1	▨

V	▨	*2*	▨	*4*	*5*	*6*	*7*
4	▨	1	▨	0	1	0	1
5	▨	0	▨	1	0	1	0

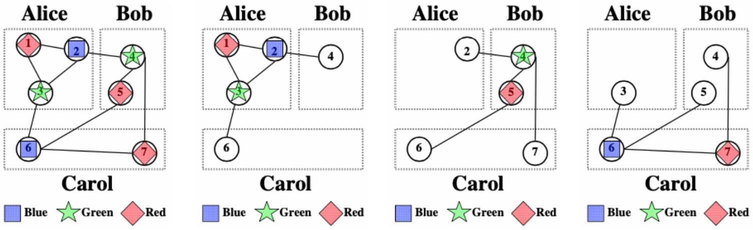
V	▨	▨	*3*	*4*	▨	*6*	*7*
6	▨	▨	1	0	▨	0	1
7	▨	▨	0	1	▨	1	0

(a) Partitioned Graph

(b) Partitioned Adjacency Matrix

Figure 2: A distributed scheduling problem formulated as a graph coloring problem, including the matrix representation of the graph [14].

Linear programming is an optimization problem that consists of 3 elements: an objective function that is optimized, a set of variables and a set of constraints, with the important characteristic that all of these elements are linear [15]. In a collaboration scenario, there are a variety of ways that the data can be partitioned between parties. These can be categorized into: horizontal, vertical or arbitrary



(a) 3-Colored (b) Alice (c) Bob (d) Carol

Figure 3: Output of the distributed scheduling problem after graph coloring, including the output that every party receives. [14]

partitioning.

- Horizontal Partitioning: Every party holds their own set of constraints with no requirements on the objective function, which may be held by one party or all of them. This can be applied to distributed scheduling, which graph coloring was also able to solve.
- Vertical Partitioning: The constraints are shared between a subset of all of the parties with the same objective function rules as horizontal partitioning.
- Arbitrary Partitioning: This describes partitioning that is a combination of horizontal and vertical. This is a more general case and if solved, it can be used to solve both vertical and horizontal partitioned problems. There could also be requirements for the way that an objective function is partitioned, for example, if only one party has the objective function.

There have been many solutions for collaborative linear programming, targeting different kinds of partitioning. Hong et al. gave a privacy preserving solution to the arbitrary case where one party owns the objective function and the other party owns the constraints [15]. A real application of this can be a manufacturer trying to find the best transportation option between their factories and suppliers. In this scenario, the constraints and objective function are separate from the collaborating parties. In the next scenario, they have also used a partitioning where two or more parties are partitioned arbitrarily, and that both the objective function and constraints are the sum of the partitioned data. This has a more substantial application in collaborative supply chain, for example a winemaking producer grows grapes in different locations and sends them to different wineries. The growing locations may contain different types and amounts of grapes, of which a winery may demand one or multiple in varying quantities. Ideal scenario would be to supply a winery with grapes that they want from the closest vineyard, however they may not have all of the grapes they need. This reveals the complexity of the scenario and it can be formed into a linear optimization problem where the objective function is cost of transport and the constraints are the supply of the vineyards and the demand of the wineries. [15] The multiparty collaboration extension to this problem would be to solve the same problem, but with several wine-making companies. Hong et al's solution to the collaborative linear programming problem used MPC, secure under the semi-honest adversarial model. In the analysis of the solution, they were not sure on whether it was secure from inference attacks, and a party could potentially learn some information such as the hardness of the problem or the type of constraints.

Another linear programming solution by Hong et al. addressed the collaborative linear programming problem where constraints are arbitrarily partitioned and each party has their own set of variables [16]. This can be applied to the collaborative supply chain in two ways. Firstly, in collaborative transportation where k companies share delivery trucks and seek to optimize on the minimum cost. The constraints would be the transportation demands for each location per company. Secondly, it can apply to a different kind of collaborative production, where k companies share raw materials for production and are optimizing more profit. The constraints in this case, are the amount of raw materials needed per company. In both of these cases, the solution of the problem should only reveal a company's own transportation routes or the amount of raw materials allocated to it. The solution to this partition of the collaborative linear problem is secure under both semi-honest and malicious model.

Distributed Constraint Satisfaction Problem (DCSP) is solved by finding a combination of variables controlled by multiple agents, such that all of the constraints are satisfied [17]. This problem can be applied to resource allocation, where a set of resources has to be allocated among different agents. To form this into DCSP, the resources that every agents can supply and demand is formed as a constraint. Leaute et al. developed a solution to DCSP using MPC under the semi-honest adversarial model, and in addition, added privacy on top of it to keep the decisions of agents private as well [17].

Finally, there are some privacy-preserving applications of MPC to be used in supply chain collaboration that are solved using methods not mentioned so far. Clifton et al. developed a solution for collaborative swapping, which can be applied to independent distributors to reduce inefficiency by swapping tasks (for example empty trucks with a full truck from a different party) [18]. Deshpande et al. developed a solution for an electronics manufacturing service and an original equipment manufacturer to negotiate on component parts without revealing information [19]. This shows that there is a substantial number of supply chain collaboration problems that can be solved with collaborative optimization combined with MPC.

To summarize the research on collaborative optimization, Table 2 is presented to show the possible applications in the supply chain along with the method used and notes on the privacy issues from the paper.

5 Combining MPC with Game Theory.

5.1 The limitations of semi-honest MPC protocols

A large number of privacy preserving methods using MPC to solve SCC problems have been developed over the last 15 years. Some of the methods that were discussed in this paper have been: capacity sharing, price-masking, collaborative production, distributed scheduling, network allocation, vehicle routing and resource allocation. Each method discussed was proven secure under a semi-honest adversarial model, and only one was also secure under a malicious model.

The malicious model has a stronger privacy guarantee than the semi-honest model, as it provides security from parties that stray away from the protocol. Its main disadvantage is that it requires significantly more computational power [4]. This results in far fewer practical uses for the model in supply chain collaboration, where it would take too long to compute for the large number of inputs and variables. Furthermore, many of the creators of the methods discussed in the paper have reasoned that security under the malicious model is not required for practical applications, as a deviation from the protocol could lead to producing a less efficient solution, which goes against the

Table 2: Summary of secure collaborative optimization research using MPC

Supply chain problem	**Optimization problem used**	**Privacy level**
Capacity Sharing (2008) [18]	Other	No listed issues
Price-masking (2011) [19]	Other	Uses a semi-trusted third party in its protocol
Distributed Scheduling and Network Allocation (2018) [14]	Graph coloring	Limited privacy leakage
Collaborative Production (2018) [15]	Linear Programming	Secure, but might be vulnerable to certain inference attacks
Collaborative Transport and Production (2012) [16]	Linear Programming	No listed issues
Vehicle Routing (2014) [13]	Traveling Salesman Problem	Could be susceptible to inference attacks
Resource Allocation (2009) [17]	Constraint Satisfaction	Modified MPC that preserves both constraints and decisions

Table 3: Table showing the possible combination of prison sentences. First number is the number of years sentence for the first prisoner, and the second number for the second prisoner.

	P2 stays silent	P2 betrays
P1 stays silent	1,1	4,0
P2 betrays	0,4	2,2

purpose of the collaboration [19]. This reasoning is quite heuristic, but a stronger alternative would be to prove that a protocol is incentive compatible.

5.2 Using game theory to achieve incentive compatibility in MPC protocols

Game theory comes from mathematics and it is the study of multiparty decision problems [20]. The model studies the interactions and possible outcomes of parties, with the assumption that they are rational decision makers. It was initially developed for economics, to study the behaviors of firms cooperating. However it has since developed to also study non-cooperative games, where individual parties compete in self-interest [21]. A popular example of such game is the "Prisoner's Dilemma". Here, two prisoners are facing light prison sentences, and are given an offer. If they betray the other prisoner, they will receive no prison sentence, but only if the other prisoner also doesn't betray them. For clearer understanding of possible combinations, Table 3 shows all of the possible combinations of prison sentences. The solution to this game is that a prisoner should always betray no matter what the other prisoner does, as that will always give himself a lower sentence. However, the outcome of both betraying is worse than both staying silent. This is an example of how it can be difficult to make rational parties acting in self interest to pursue the common good [22]. In the context of supply chain collaboration, game theory can be used to show that the benefits of collaborations can be quickly lost with parties that benefit from breaking protocol.

Shoham and Tennenholtz added on to non-cooperatives games by introducing non-cooperative computation (NCC) [23]. It is a "joint computation of a function by self-motivated agents, where each of the agents possesses one of the inputs to the function" [23]. While somewhat similar to MPC, NCC is still a game theoretic concept that analyzes what is in the best interest of each party, rather than assuming cases with a 'good' or 'bad' party. A function is in the class of NCC, if the parties can be incented to give their correct private input to the function.

Incentive compatibility is used for mechanism design, that proves that if every party has truthfulness as a dominant strategy, then a mechanism is 'cheat-proof' [24]. A malicious party wouldn't try to cheat if it was not his best strategy. There are two basic methods for having incentive compatibility, either truth telling is the dominant strategy, or a party will not cheat because they will be caught. Bhargava and Clifton state that a privacy preserving secure protocol cannot exist, unless there is incentive compatibility [24]. Cryptographic protocols assume a party either follows the protocol or deviates from it, while game theory assumes a party acts to their best interest. The combination of both can best model a real life scenario.

A simple example of why incentive compatibility is necessary for secure protocols occurs in this case of revenue sharing contracts [24]. Imagine if a retailer pays a supplier a price for each unit purchased, as well as a percentage of the revenue that they generate. The contract would consist of two parts, in the first part the quantity and the price at which good is sold is decided through profit maximizing, then in the second part, the retailer shares profits. A privacy preserving method to this problem would allow the negotiation of this contract without revealing the retailer's costs, sales and revenue, which would prove to be quite useful. However, if this information were to be hidden from the supplier, then the retailer's best strategy would be to lie about their revenue. A lower revenue that is declared results in less money needed to be paid to the supplier. Input modification is not protected by MPC, even by the malicious model. However as it turns out, in a multiparty scenario (more than 1 retailer), the privacy preserving protocol is incentive compatible. This is because now there is competition, so a retailer that lies risks their supplier switching to an honest retailer, that now offers the best outcome for the supplier. This shows how some privacy preserving protocols may be incentive compatible and some not, and that a simple assumption of semi-honest parties is not sufficient.

From the collaborative optimization problems that were analyzed, only Clifton et al.'s collaborative swapping and Hong et al.'s collaborative transportation and production solution proved or considered incentive compatibility. Clifton et al. considered cases where a lying party may hide some of their points, however that results in a final suboptimal solution and longer travel time for the party. Alternatively, if a party inputs false points to swap, then the lying party will be quickly discovered as soon as an honest party tries to pick up the nonexistent load [18]. Meanwhile, Hong et al considered potential benefits of a dishonest party, as well as collusion of parties and improved their linear programming solution so that it was incentive compatible [16]. In both cases, Clifton et al. and Hong et al. showed that cheating either increases the cost for a malicious party or they are caught by an honest party. The remaining privacy preserving MPC methods assumed that the parties are semi-honest, so do not have protection from malicious parties breaking protocol or lying about their private input, even if it might be in that party's best interest.

6 Responsible Research

The literary study conducted in this research paper used academic studies indiscriminately, locating them from Google Scholar. To reproduce the results, the same papers can be found which will give the same results.

Ethically, if the use of the privacy methods being investigated in this paper were to become more popular, there would be a great ethical benefit. For example, protecting the privacy of a company's supply chain, would prevent other companies from using that information to their competitive advantage. In healthcare, where people's private data may be part of a collaboration, privacy preserving methods can stop their data from leaking through the collaboration. Applications of supply chain collaboration in logistics, can result in reduced transportation, which reduces greenhouse gas emissions that contribute to the climate change crisis.

7 Discussion

From the literary study it was found that there are many collaborative optimization problems that can be used to solve collaborative supply chain problems. The collaborative optimization problems were secure using MPC under the semi-honest adversarial model. The study also demonstrated how a semi-honest assumption may not be sufficient for real life applications, where malicious parties may exist. However MPC under a malicious model is too inefficient. To address the issue, the literary study found that incentive compatibility combined with a semi-honest assumption can achieve better practical security without becoming inefficient.

These results build on the existing evidence that a high degree of trust is integral for parties to collaborate in the supply chain. The addition of incentive compatibility to MPC under the semi-honest assumption improves the practicality of performing collaborative optimization in supply chain collaboration. These findings match the developments from other areas of research of combining cryptography with game theory. Balanjaneyudu et al. and Kantarcioglu et al. created incentive compatible privacy-preserving methods for data analysis, addressing the issue that MPC could not verify the validity of a party's private input data [25] [26]. Xu et al. combined MPC with game theoretical approach to modify the interactions of users in distributed data mining [27].

A protocol that is incentive compatible assumes that the participating parties are rational. Therefore an obvious limitation of the model is that in some instances, an irrational party could still hinder the collaboration process even if they would get caught or suffer economic loss. Another limitation to the research is that some of the methods researched may have serious underlying privacy issues that were not discovered by their authors or by others. Since this paper consisted of just a literary study, there was no security analysis of the methods.

8 Conclusion and Future Work

Before parties can begin to realize the benefits from supply chain collaboration, it is of great importance that their private inputs are kept private. This paper conducted a literary study that showed that MPC can be used to preserve privacy in solutions to various supply chain collaboration problems. MPC under the malicious adversarial model is rarely used in the solutions due to its inefficiency that make it impractical. The semi-honest model is used more frequently, however its security lacks protection against parties that attempt to break the protocol. Furthermore, a limitation of MPC that affects both models is that neither can verify that a party has not lied in the private input that they gave.

As a solution to this problem, this paper recommends combining MPC with game theory to achieve incentive compatible secure protocols for supply chain collaboration. In an incentive compatible semi-honest adversarial model, a party will not break protocol or lie if it is not in its best interest. This emulates a real world scenario better, as collaborating parties in the supply chain aim to increase efficiency for monetary gain, and if a protocol is incentive compatible, then not lying will result in the greatest monetary gain for the party. Furthermore, incentive compatible protocols are also able to maintain efficiency, as they can remain secure under just a semi-honest model.

References

[1] C. A. Soosay and P. Hyland, "A decade of supply chain collaboration and directions for future research," *Supply Chain Management: An International Journal*, 2015.

[2] A. Bednarz, *Methods for two-party privacy-preserving linear programming.* PhD thesis, 2012.

[3] D. Buss, "Case study: Land o'lakes and collaborative logistics," 2003.

[4] Y. Hong, J. Vaidya, and S. Wang, "A survey of privacy-aware supply chain collaboration: From theory to applications," *Journal of Information Systems*, vol. 28, no. 1, p. 243â268, 2013.

[5] S. Chakraborty, "Secure multi-party computation: how to solve the conflict between security and business intelligence," tech. rep., Citeseer, 2015.

[6] A. C.-C. Yao, "How to generate and exchange secrets," in *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pp. 162–167, 1986.

[7] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game, or a completeness theorem for protocols with honest majority," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pp. 307–328, 2019.

[8] H. Delfs and H. Knebl, *Provably Secure Encryption*, pp. 191–219. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[9] S.-C. S. Cheung and T. Nguyen, "Secure multiparty computation between distrusted networks terminals," *EURASIP Journal on Information Security*, vol. 2007, pp. 1–10, 2007.

[10] R. Cramer and I. Damgård, "Multiparty computation, an introduction," in *Contemporary cryptology*, pp. 41–87, Springer, 2005.

[11] R. Canetti, "Security and composition of multiparty cryptographic protocols," *Journal of CRYPTOLOGY*, vol. 13, no. 1, pp. 143–202, 2000.

[12] P. Bogetoft, D. L. Christensen, I. Damgård, M. Geisler, T. P. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, *et al.*, "Multiparty computation goes live.," *IACR Cryptol. ePrint Arch.*, vol. 2008, p. 68, 2008.

[13] Y. Hong, J. Vaidya, H. Lu, and L. Wang, "Collaboratively solving the traveling salesman problem with limited disclosure," in *Data and Applications Security and Privacy XXVIII* (V. Atluri and G. Pernul, eds.), (Berlin, Heidelberg), pp. 179–194, Springer Berlin Heidelberg, 2014.

[14] Y. Hong, J. Vaidya, and H. Lu, "Securely solving the distributed graph coloring problem," 2018.

[15] Y. Hong, J. Vaidya, N. Rizzo, and Q. Liu, "Privacy-preserving linear programming," in *WORLD SCIENTIFIC REFERENCE ON INNOVATION: Volume 4: Innovation in Information Security*, pp. 71–93, World Scientific, 2018.

[16] Y. Hong, J. Vaidya, and H. Lu, "Secure and efficient distributed linear programming," *Journal of Computer Security*, vol. 20, no. 5, pp. 583–634, 2012.

[17] T. Leaute and B. Faltings, "Privacy-preserving multi-agent constraint satisfaction," in *2009 International Conference on Computational Science and Engineering*, vol. 3, pp. 17–25, 2009.

[18] C. Clifton, A. Iyer, R. Cho, W. Jiang, M. Kantarcıoğlu, and J. Vaidya, "An approach to securely identifying beneficial collaboration in decentralized logistics systems," *Manufacturing & Service Operations Management*, vol. 10, no. 1, pp. 108–125, 2008.

[19] V. Deshpande, L. B. Schwarz, M. J. Atallah, M. Blanton, and K. B. Frikken, "Outsourcing manufacturing: Secure price-masking mechanisms for purchasing component parts," *Production and Operations Management*, vol. 20, no. 2, pp. 165–180, 2011.

[20] R. Gibbons *et al.*, *A primer in game theory*. Harvester Wheatsheaf New York, 1992.

[21] M. A. Khan and Y. Sun, "Non-cooperative games with many players," *Handbook of game theory with economic applications*, vol. 3, pp. 1761–1808, 2002.

[22] S. Kuhn, "Prisonerâs dilemma." `https://plato-stanford-edu.tudelft.idm.oclc.org/entries/prisoner-dilemma`, 1997.

[23] Y. Shoham and M. Tennenholtz, "Non-cooperative computation: Boolean functions with correctness and exclusivity," *Theoretical Computer Science*, vol. 343, no. 1-2, pp. 97–113, 2005.

[24] R. Bhargava and C. Clifton, "When is a semi-honest secure multiparty computation valuable?," in *Decision and Game Theory for Security* (T. Alpcan, Y. Vorobeychik, J. S. Baras, and G. Dán, eds.), (Cham), pp. 45–64, Springer International Publishing, 2019.

[25] T. Balanjaneyudu and K. G. Reddy, "Incentive compatible privacy-preserving data analysis," *International Journal of Scientific Engineering and Technology Research*, 2015.

[26] M. Kantarcioglu and W. Jiang, "Incentive compatible privacy-preserving data analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1323–1335, 2012.

[27] L. Xu, C. Jiang, Y. Qian, and Y. Ren, "Privacy-accuracy trade-off in distributed data mining," in *Data Privacy Games*, pp. 151–177, Springer, 2018.