Efficient Multi-Party Computation Algorithm Design For Real-World Applications

Zengxiang Li[a], Chutima Kitcharoenpaisan[a,b], Phond Phunchongharn[b], Yechao Yang[a], *Rick Siow Mong Goh*[a], and *Yusen* Li[c]

[a]Institute of High Performance Computing, A * STAR, Singapore
[b]King Mongkut's University of Technology Thonburi, Thailand
[c]Nankai – Baidu Joint Lab, Nankai University

***Abstract*— Secure Multi-Party Computation (MPC) is a promising privacy-preserving technology to enable multiple trustless parties to compute a function jointly without revealing private inputs to each other. With the fast development of MPC protocols, software implementation, and underlying computation infrastructure, MPC has developed from purely theoretical interest to tangible platform implementations for real-world applications. In this paper, we investigate multiple mechanisms to design efficient MPC algorithms by avoiding costly MPC operations and leveraging parallel operations. In order to speed up database table searching, a machine learning-based approach is proposed to completely avoid equality-check operations, playing the trade-off between efficiency and accuracy. According to our experimental results, a well-designed MPC algorithm could improve performance and scalability significantly, and thus make MPC technology practicable.**

Keywords—Privacy-preserving, Multi-Party Computation, Machine Learning, Performance, Scalability, Cargo Consolidation

I. INTRODUCTION

In the big data era, data becomes more and more valuable to find out deep insights and thus enable better recommendation and decision making. Moreover, sharing data across multiple organizations can be beneficial in terms of transparency and global optimizations. Take maritime shipping as an example, optimum cargo consolidation is achievable, if cargo owners, vessel owners, and intermediate brokers could share the supply and demand in a centralized marketplace. As a result, all participants could obtain benefits from the global resource optimizations and potential new business.

However, there are several barriers that should be removed to encourage people and companies to share personal and confidential data to a trusted third party or even their competitors. Sharing privacy-sensitive data might violate regulations (e.g., European General Data Protection Regulation (GDPR) entered into force on May 2018), causing unaffordable punishment. In addition, a company might lose its business competitive advantages over other competitors, if confidential data is leaking. For example, a vessel owner might be inferior in price negotiation, if its un-occupied capacity and operation cost is disclosed to competitors. Therefore, privacy-preserving technologies are desirable to enable collaborations across multiple organizations without compromising their data privacy.

Secure Multi-Party Computation (MPC) is a class of cryptographic techniques that enables multiple parties to compute a function jointly over privacy-sensitive data. No party learns anything beyond the final output of the function. The output is sent to authenticated parties only and the whole confidential computation procedure never reveals the private data inputs of individual parties.

In the last few decades, several active MPC projects (e.g., Scale-Mamba [1], EMP-Tool [2] and Enigma [3]) address research topics such as security level and scalability. Velgushev et al [4] have proposed an MPC-based query compiler that makes MPC on "big data" accessible and efficient. According to recent survey papers [5] [6], MPC performance has been improved tremendously, due to the fast development of MPC protocols, software implementation, and underlying computation infrastructure. Some commercial companies have been founded using MPC to solve real problems that otherwise cannot be solved. Sharemind uses MPC to investigate the correlations between education and financial data [7]. It is also used to prevent satellites from colliding without disclosing trajectories [8]. Moreover, Sharemind employed cryptographically secure multiparty computation to tackle privacy-preserving Principal Component Analysis (PCA), which is essential for reducing the dimensionality of the problem for genome-wide association studies [10].

MPC computation speed is heavily dependent on the choice of MPC protocols, the security levels, software implementation, programming languages, the problem scale (e.g., number of parties), and the power of underlying computation infrastructure. Different from traditional operations on original open data, the computation and communication cost of different kinds of MPC operations and data types might be dramatically different. In addition, performing private operations in parallel could save a great deal of network communication overhead, and thus reduces the time cost involved [9]. Hence, implementing an efficient secure MPC-algorithm remains a challenging task. It requires domain-specific expertise from both overlying applications and underlying MPC implementations.

In this paper, we investigate multiple mechanisms to design efficient MPC algorithms by avoiding costly MPC operations and leveraging parallel operations, according to the application requirements. For example, global cargo consolidation requires a large number of queries of travel distance (or time) between source and destination ports, which must always be kept in privacy without endangering business competitive advantages. Implementing such an algorithm in a traditional manner requires massive expensive MPC equality-check operations [9], crossing the entire database table. To solve the problem, we propose to use a machine learning model trained by the database table to calculate approximate queries results, while completely avoiding equality-check operations. According to our

978-1-7281-2583-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICPADS47876.2019.00154

experimental results, a well-designed MPC algorithm could improve performance and scalability dramatically, and thus make MPC practicable in real-world applications.

II. BACKGROUND

In this section, we introduce some background of MPC technology, especially the Sharemind framework [9]. In addition, the problem statement of cargo consolidation in maritime shipping is also discussed in detail.

A. MPC Procotol and Implementation: Sharemind

There are two dominated MPC protocols: Garbled circuits and secret sharing. The latter has been more commonly used in production systems, e.g., JIFF and Enigma [3] are using Shamir secret sharing based on polynomial interpolation, while Sharemind [9] is using lightweight additive secrete sharing. Sharemind maintained by a commercial company is selected for performance comparison in this paper, due to its leadership in applying MPC technologies to real-world applications [7, 8, 10].

The architecture of the Sharemind framework composes of three basic roles: input parties, computing parties, and result parties. Input parties split each sensitive data into "secret shares" by adding random numbers. Then, the secret shares are delivered to corresponding computing parties for confidential computation. Computing parties jointly computing the confidential computation composed of several pre-agreed functions. Result parties receive results or partial results of the functions from the confidential computation. The number of input parties (data source providers) and result parties (analytics users) is not restricted. There is no restriction that an input party cannot be the result party or computing party. For example, privacy-preserving for genomic data MPC application, the data provider (hospital, gene bank) might also want to query the analytics results.

Sharemind supports hundreds of instructions on different data types and bit widths. Secure instructions in Sharemind protection domains are designed in sequential and parallel manners. Sequential private operations are individually slow requiring a great deal of network communication overhead. In contrast, parallel private operations reduce the time cost involved, and the network communication cost is reduced too, as it's more efficient to send data in bulk. Due to the complexity of MPC protocols, the cost of secure instructions can be significantly different [9]. For example, equality-check MPC operation which requires looping individual bits at the protection domain could be several times slower than multiplication MPC operation. Sharemind already supports a comprehensive set of floating-point operations. However, secure floating-point addition and multiplication are much more expensive than those for integers. Hence, it is very common to use integer and fixed-point numbers to replace floating-point numbers to play the trade-off between efficiency and accuracy [10]

B. Problem Statement: Cargo Consolidation in Maritime Shipping

In maritime shipping, cargo consolidation is very important to reduce the operational cost and increase vessel utilization. Traditionally, cargo and vessel owners, as well as the intermediate brokers, have to share the supply and demand in a centralized marketplace. As a result, all participants could obtain benefits from the global resource optimizations and potential new business. However, they are cautious to share personal and confidential data due to regulatory constraints and business competition. For example, the vessel owner should not disclose its unoccupied capacity and operation cost to business partners and competitors, while the cargo owner keeps its shipping demand in secret to protect its business activity and confidentiality. Therefore, a privacy-preserving marketplace is desirable to enable collaborations across organizations without compromising their data privacy.

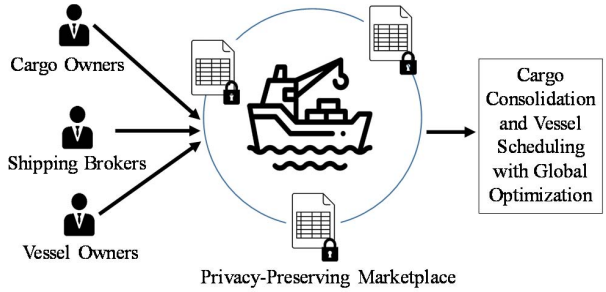

Figure 1. Privacy-Preserving Marketplace for Cargo Consolidation

To achieve global optimization, the cargo consolidation and vessel scheduling algorithms require a large number of queries of travel distance (or time) between source and destination ports by searching the entire distance table. The supply and demand, i.e., the source and destination ports of vessel and cargo respectively must always be kept in privacy to ensure the cargo consolidation would not endanger business competitive advantages. The supply and demand matching results are informed to the relevant and authenticated parties only for privacy protection.

Maritime port distance table is an open dataset including the distance between every pair of ports. Currently, the Sharemind framework does not support string comparison. Thus, data pre-processing must be done before the computation process. First, the port name needs to be changed from string to integer. Second, the distance table should be stored in a two-dimensional array. To find the distance between two specific ports, a loop is usually used to iterate to each row in the dataset and get the result as the distance. However, with MPC computation, the input data (source port and destination port) is privacy-sensitive and should be saved as secret shares distributed among computing parties.

Due to the complexity of MPC protocols, the cost of operations (e.g., addition, multiplication and equality-check) could be dramatically different, in terms of the number of computation rounds and communication messages. The equality-check operations are much more expensive, compared with addition/subtraction and multiplication operations. This is

because the MPC protocols for equality-check operations have to be done by individual bits, and the comparison results must be kept in secrete shares to ensure the privacy protection in the entire confidential computation. The significant difference in MPC operation cost has been observed in most MPC projects including JIFF and Sharemind [9].

Hence, it is non-trivial to design an efficient MPC-based algorithm for the cargo consolidation application. If programming MPC-based algorithms in a traditional way, the performance may degrade dramatically due to the intensive use of high-cost operations. We will introduce empirical optimization methods for privacy-preserving table searching, which are frequently used in transportation for distance and travel time queries between two confidential locations.

III. PRIVACY-PRESERVING TABLE SEARCH ALGORITHMS

In this section, several MPC-based algorithms are introduced to improve the performance of privacy-preserving table searching, by reducing or completely avoiding the use of expensive MPC operations.

A. Traditional Algorithm

Algorithm 1: Traditional Algorithm

```
function getDistance (port1, port2);
Input: port1= source port (integer),
       port2 = destination port (integer)
Output: The distance between port
result = 0
for i := 0 to length of port distances do
    for j := 0 to length of port distances do
        port1Eq = (port1 equal to i?) 1:0
        port2Eq = (port2 equal to j?) 1:0
        distanceBetweenIJ = portsDistances[i][j]
        cellRes = (((port1Eq + port2Eq) equal to 2?) 1:0)
                  * distanceBetweenIJ
        if result != 0 then
            result = result + cellRes
        else
            result = cellRes
        end
    end for
end for
return result
```

Figure 2. Traditional algorithm

The pseudocode of the traditional algorithm for privacy-preserving table searching is shown in Figure 2. Please be noted that usually, it requires traditional "AND" operation (i.e. a combination of addition and checking for equality). We have already used multiplication operation to implement "AND", otherwise, this traditional algorithm would be even more time-consuming. In addition, all intermedia variables (e.g., port1Eq, port2Eq) could not be opened for escaping different levels of looping control. Among the operations used in the traditional algorithm, equality-check operation (48 in total if the number of ports equal to 4) is the most expensive. To speed up the MPC-based algorithm, we should reduce and avoid the number of these operations.

B. Machine Learning-based Algorithm

Our proposed machine learning-based (ML-based) algorithm is composed of two steps. In the first step, a machine learning model is trained off-line using the database table (e.g., the distance between maritime ports). Since the database table is usually an open dataset, the ML training step is on plain data without using MPC. In the second step, the trained ML model is used to calculate the approximate distance between source and destination maritime ports. By replacing the table searching, expensive equality-check MPC operations could be avoided completely. The ML-based algorithm is shown in figure 3.

Algorithm 2: Machine learning
(One hot encoding with polynomial regression degree 2):

```
Input: x = values from every column except for distance
between port
       y = values from column distance
Output: Polynomial equation to compute the distance
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(degree = 2)
x_poly = poly.fit_transform(x)
lin = LinearRegression()
lin.fit(x_poly, y)
```

Figure 3. Machine Learning ML-based algorithm

If we simply use source port and destination port as the features, the polynomial regression machine learning model has very low training accuracy to fit the results saved in the entire distance table. With the increasing number of ports, the model training accuracy becomes worse and worse. To improve, more features need to be added to train the polynomial regression machine learning model. One-hot encoding which has been proved to be helpful for increasing ML model accuracy is applied in our algorithm.

Be noted that the distance table is an open data set and the ML model is trained offline using polynomial equation degree two, without secure MPC privacy protection. To support the privacy-preserving table search for cargo consolidation, the trained machine learning model (i.e., the long polynomial equation with trained co-efficiency) must be computed in Sharemind using confidential source and destination ports as inputs. Due to one-hot encoding features, the equation becomes much longer with many variables (refer to Table 1 for more details). It is still expensive because of a large number of multiplication and addition operations in secure MPC confidential computation.

Parallel computation is provided by Sharemind [9]. Instead of calculating the long polynomial equation sequentially, we use addition and multiplication operations on arrays, enabling computation and communication in bulks. After applying parallel MPC computation, an ML-based algorithm could improve scalability significantly. When the number of ports is increasing, parallel computation helps to reduce the computation time dramatically compared with the sequential version. More details are illustrated in Section IV.

IV. PERFORMANCE EVALUATION

The experiment is evaluated using machine-learning-based algorithms with sequential and parallel versions on the Sharemind emulator. The parameters and accuracy of machine learning models trained without secure MPC privacy protection are shown in Table 1. Due to one-hot encoding, the number of features and co-efficiency are increased dramatically with the increasing number of maritime ports is increased from 4 to128.

Table 1. Parameters and accuracy of machine learning ML-based Algorithms

No. of ports	**4**	**16**	**64**	**128**
No. of Features	8	32	128	256
No. of Coefficients	45	561	8,385	33,153
Calculation Error (RMSE)	3.426e -13	3.653e -12	5.367e -11	1.456e -11

Table 2. Computation time (ms) of traditional and ML-based algorithms

No. of ports	**4**	**16**	**64**	**128**
Traditional	113. 043	1,802. 642	28,837. 049	115,344. 807
ML-based Sequential Version	352. 333	5,253. 235	79,575. 487	-
ML-based Parallel Version	36. 408	121. 337	625. 379	-

Root-mean-square error (RMSE) and R-squared (R2) measures are used to evaluate the accuracy of the trained ML model. RMSE is the standard deviation of the residuals (prediction errors) which is the measure of how large your residuals are spread out. The result of our regression model is almost zero which indicates that it is a good regression model and the polynomial equation is very accurate. R-squared (R2) measures the strength of the relationship between your model and the dependent variable. R-squared values range from 0 to 1. The ML polynomial regression model is trained by open distance data tables with a different number of ports, and the R2 is very close to 1.0, indicating that the model could fit the data point very well. Due to the high accuracy of well-trained ML models, an ML-based algorithm could obtain the approximate distance between the confidential source and destination ports, which is almost the same as the table searching results.

The computation time of traditional and ML-based algorithm on Sharemind emulator are listed in Table 2. The ML-based algorithm generates a long polynomial equation for computing highly accurate port distance, due to one-hot encoding features and the dramatically increased number of variables. For the 128 ports case, ML model MPC computation encounters an exception due to the very long polynomial equation, Sharemind emulator limitation and computation power constraints on a PC.

As shown in Table 2, the sequential ML-based algorithms could be even worse than the traditional algorithm, due to the complex polynomial equation and large number of coefficients. Fortunately, the ML-based parallel version achieves 10 to 127 times of speed up compared with the sequential version. Compared with the traditional algorithm, an ML-based parallel version could achieve 3 to 46 times of speedups. ML-based parallel version achieves the lowest computation time with the best performance and scalability among other algorithms.

V. CONCLUSION AND FUTURE WORK

In this paper, multiple mechanisms are investigated to design efficient MPC algorithms by avoiding costly MPC operations and leveraging parallel operations. Different algorithms are proposed and implemented to address a typical privacy-preserving table search problem, which is frequently used in transportation for distance and travel time queries between two confidential locations. After employ one-hot encoding features, ML-based algorithms could handle a large table with close to 100% accuracy. The parallel computation version of the ML-based algorithm could be 10 to 127 times faster than the sequential one.

In the future, our work will be extended to complex algorithms and workflows to make MPC practicable for large-scale real-world applications with acceptable performance and scalability. In the meantime, we will develop an adaptor/converter to make the developed optimized algorithms applicable to various MPC protocols or platforms deployed in a distributed environment. Besides the execution time, the computation resource cost will also be considered for developing the most suitable MPC-based algorithms.

REFERENCES

[1] Aly et al. SCALE-MAMBA, (2019), GitHub Repository, https://github.com/KULeuven-COSIC/SCALE-MAMBA

[2] Xiao Wang and Alex J. Malozemoff and Jonathan Katz, (2016), GitHub Repository, https://github.com/emp-toolkit

[3] Guy Zyskind, Oz Nathan, and Alex Pentland, "Enigma: Decentralized Computation Platform with Guaranteed Privacy" https://arxiv.org/abs/1506.03471

[4] Volgushev, Nikolaj and Schwarzkopf, Malte and Getchell, Ben and Varia, Mayank and Lapets, Andrei and Bestavros, Azer. "Conclave: Secure Multi-Party Computation on Big Data." Proceedings of the Fourteenth EuroSys Conference, 2019

[5] W Archer David, Bogdanov Dan, Lindell Yehuda, Kamm Liina, Nielsen Kurt, Pagter Jakob, P Smart Nigel, N Wright Rebecca. (2018). From Keys to Databases—Real-World Applications of Secure Multi-Party Computation. Computer Journal. 61. 1749-1771.

[6] David Evans, Mike Rosulek, Vladimir Kolesnikov. (2018) A Pragmatic Introduction to Secure Multi-Party Computation, NOW Publishers

[7] Chillotti, I., N. Gama, M. Georgieva, and M. Izabachène (2017). "Faster Packed Homomorphic Operations and Efficient Circuit Bootstrapping for TFHE". In: Advances in Cryptology – ASIACRYPT

[8] Dan Bogdanov, Liina Kamm, Baldur Kubo, Reimo Rebane, Ville Sokk, and Riivo Talviste. Students and taxes: A privacy-preserving social study using secure computation. In Privacy Enhancing Technologies Symposium (PETS), 2016.

[9] Sharemind logo Developer Zone, (2019) https://docs.sharemind.cyber.ee/2019.03/development/secrec-reference#performance-of-shared3p-protocols

[10] D. Bogdanov, L. Kamm, S. Laur and V. Sokk, "Implementation and Evaluation of an Algorithm for Cryptographically Private Principal Component Analysis on Genomic Data," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 15, no. 5, pp. 1427-1432, 1 Sept.-Oct. 2018.