SECURE MULTI-PARTY COMPUTATION - A SURVEY

Arpesh Singh, Vishal Passricha
Computer Engineering Department, National Institute of Technology, Kurukshetra, Haryana, India

ABSTRACT - This is particularly true if the data from several sources are combined for some common task may be beneficial and speed up the task. For example, by combining data from several of its institutions, a state can discover trends or pin-point problematic issues. However, this is often avoided due to privacy concerns, as the combined data set becomes an attractive target for both inside and outside attackers. In this internet era, privacy and security of data are highly demanded and it is a sensitive issue. Secure Multi-Party Computation (SMC) is a technology which permits data to be processed with privacy so that the computation servers could not see any actual data values. First practical implementations of this emerging technology is started in the 2000s. This technology is now mature enough to be used for privacy-preserving data analysis on real data. SMC ease the problem of data sharing between the parties as computation is securely done over the secure inputs (encrypted inputs) provided by different parties. It generates the corresponding output in a secure form which is correspondingly delivered to each party. In this paper, a detailed study of SMC is discussed which benefits the readers to understand it at a single point.

KEYWORDS - Cryptography, Data Sharing, Encryption Technique, Privacy, Secure Multiparty Computation.

INTRODUCTION

The present era is an era of the internet. In this, all world is connected through the internet and share their data and information over the internet. The major issue with sharing is some vulnerable activities i.e. modification, alteration, etc. may happen with this data. This is a sensitive issue. For example, a set of parties who wish to correctly run some common function on their local inputs. Neither they trust the channels by which they communicate nor each other. Therefore, they try to keep their local data as private as possible and want to perform computation on others data. This arises the problem of trust among multiparty computation. Figure 1 shows the block diagram of multiparty computation model. Various forms are taken by this problem depending on the underlying network, on the function to be computed, and on the amount of distrust the parties having in each other and in the network. This issue is resolved by Secure Multi-Party Computation (SMC). Its block diagram is shown in figure 2. SMC protocols allow two or more mutually non-trusting parties to compute a shared result while keeping their respective inputs hidden from the other participating parties. SMC is the fundamental application of cryptography, as well as relevant to practical cryptography applications. Cryptographic community has developed the field of SMC to allow applications to perform computation over encrypted data. In SMC, the required data are shared in encrypted form by the computing parties.

After collecting the encrypted input, the server applies function directly to the encrypted values and returns users the generated encrypted output. In SMC, the input is encrypted in such a way that it remains secure and can also be computed without decryption by maintaining the correctness of output.

![Fig 1: The Block Diagram of Multiparty Computation Model](image1)

![Fig 2: The Block Diagram of Secure Multiparty Computation Model](image2)
According to the literature available, mostly SMC research are theoretical studies and few implementing problems have been studied. Several examples that are using SMC for secure input computation are privacy-preserving statistical database, information retrieval problem, and privacy-preserving data mining.

RELATED WORK
Secure Multi-Party Computation was introduced in 1982 by A. Yao[1] and was further extended by O. Goldreich et al.[2]. The first representation of the computational problem is a combinatorial circuit, then for every gate in the circuit, a short protocol is run by the parties. The size of the protocol depends on the size of the input domain, and on the complexity of expressing such a computation.

According to literature, many researchers have worked to convert traditional computation into secure multi-party computation by applying any secure mechanism to preserve the privacy of data. Privacy is a fundamental need of present computations. There are several problems resolved using SMC as a solution like a database query[3], geometric computation[4], scientific computation[5], data sorting and selection[6], etc.

A) Database Query
In SMC, the major problem is the matching of the query in the database, where one party needs to check his input in another database by maintaining the privacy of both the parties. In this problem, matching result are either given as perfect ‘yes’ or ‘no’ in case of binary function or the output may be produced in the form of correlation score which denotes the percentage of matching with the like items in the database.

Example: Let’s say Alice wants to search for some entry to be present in Bob’s database, but she doesn’t want to share details with Bob because of personal information in the input. Bob also does not want to give his whole database access to Alice to search her query.

B) Geometric Computation
In geometric computation, points are provided as input then different geometric functions are applied to them to obtain the corresponding result. The problem in co-operative geometric computation is the privacy of the coordinates provided as input.

Example: Two persons standing at the points $P_1$ and $P_2$ want to know their distance from each other but don’t want to reveal their original location for a security purpose.

C) Scientific Computation
Linear equation problems are solved mutually by using several linear equations provided by different parties without revealing the actual equation to another party. Such problems occur between two competitors working on the same project proving individual linear equations.

Example: Alice has some private linear equations say $M_1x = a$ and Bob also has private equations say $M_2x = b$. They mutually want to compute the value of $x$ that satisfy both.

D) Data Sorting and Selection
The private dataset is provided by two or more parties to sort their data and server return the indices of their respective inputs from the combined sorted dataset. Further, this will help in finding $k^{th}$ element or median of the dataset obtained. Comparison of the dataset is needed using SMC in this problem.

Example: Let Alice has some private dataset $D_1$ and Bob has a private dataset $D_2$. They mutually want to sort the dataset and need to know the $k^{th}$ element belongs to which party.

There may be some more secure multi-party computations which are not touched, as the present era is getting completely depended upon internet creates some more important computation which can be only solved by SMC.

SECURE MULTI-PARTY COMPUTATION: A DETAILED STUDY
SMC protocols allow two or more bilaterally non-trusting parties to compute a shared output while keeping their respective inputs hidden from the other participating parties. A number of different techniques have been used to achieve this impressive security guarantee.

This is a technology that allows joint computation on data from the different parties in such a way that the data remains private while only the intended result becomes known. More precisely, MPC allows a number of parties $P_1, P_2, P_3, ..., P_n$ with input $x_1, x_2, x_3, ..., x_n$ to compute function $(y_1, y_2, y_3, ..., y_n) = f(x_1, x_2, x_3, ..., x_n)$ while guaranteeing interesting properties such as input and output privacy (that the inputs and outputs remain private to each party $P_i$) and correctness (that only $f$ is evaluated and not some other function $f'$ is evaluated), even if some of the parties may be acting maliciously.

A. Yao[1] developed the first protocol for SMC in 1982. Currently, research in secure multiparty computation has become mature and applied into a broad and diverse field of technology. Encompassing constructions such as oblivious transfer[7] and private information retrieval[8][9] are latest growing field of SMC. SMC generally describes any technique in a privacy-preserving way for evaluating some functionality. The security provided by these schemes is mostly commonly demonstrated using area/ideal world paradigm, which proves that in a real-world execution of the SMC protocol, participants output a computationally indistinguishable set of values from an ideal world where the function is evaluated by a trusted third party, who then distributes some output to all participants[10]. While many variations on this paradigm are used to prove the security of different constructions, this basic concept instinctively demonstrates that the SMC protocol provides equivalent security to the best achievable solution of a
A) **Secret-Sharing**

Secret-sharing SMC techniques encompass a variety of protocols that allow to split private input into encrypted parts and share between the parties[11]. These shares are then processed in a privacy-preserving interactive protocol and combined at the end to recover the resulting output. These protocols commonly require some random data (e.g., multiplication triples) to be encrypted in a pre-processing phase, which is then combined with the secret shares during the online computation to make it feasible for non-linear operations such as multiplications in the arithmetic setting or bitwise AND in the boolean setting.

O. Goldreich et al.[2] developed one of the earliest secret-sharing techniques (GMW Protocol), which allows the for computation over boolean circuits, and required an oblivious transfer protocol to be executed for computing AND gates. Recent developments have shown that the GMW protocol can be quite efficient for evaluating low-depth boolean circuits[12][13]. In addition, many arithmetic secret-sharing protocols and optimizations have been developed using a wide range of underlying secret sharing protocols[14][15][16]. These optimizations have been augmented with further research into constructing optimal arithmetic circuit representations for common functions[17][18]. However, secret-sharing protocols are generally found optimal for large numbers of participants and tend to be less practical in the two-party setting.

B) **Homomorphic Encryption**

Homomorphic encryption uses several encryption schemes that perform several homomorphic operations to the whole cipher value. Especially, given an encryption technique $(\text{Gen}(), \text{Enc}_{\text{key}}(), \text{Dec}_{\text{key}}())$, two messages $M_a, M_b$ holds some operation $\star$ over normal text and operation $\star$ over encrypted value that is given as:

$$\text{Decrypt}_{\text{key}}(\text{Encrypt}_{\text{key}}(M_a) \star \text{Encrypt}_{\text{key}}(M_b)) = M_a \star M_b$$

(1)

For instance, the homomorphic additive scheme turns it to an additive operation, allowing the plaintext messages to get added still remaining as encrypted. This process allows for a single homomorphic operation which is comparatively at par to the standard RSA construction[19]. When a system is partly encrypted on the homomorphic basis, it results in a variety of special-purpose protocols[20][21][22][23][24][25][26][27][28][29] to sort out a particular function in a suitable feasible way. Although, generic computation of encrypted type data is not possible if an operation is not supporting any homomorphic operation sets.

In order to examine the arbitrary functions from encrypted data, Rivest et al.[24] suggested a basic fully-homomorphic encryption (FHE) form. The scheme must come with three basic properties to justify FHE. Firstly, a universal set of homomorphic operations should be allowed by it, operations viz. multiplication, additions required to allow. Second, it should allow examining an arbitrary-depth circuit. Thirdly, ciphertext size should be independent of function size that resulted it. C. Gentry[25] developed the novel scheme for achieving those desired properties, it used the concept of bootstrappable encryption plan. The basic idea is dependent on the use of basic Somewhat Homomorphic Encryption (SHE), this scheme allows a universal set of homomorphic operations, although it is only limited to evaluation of limited depth circuits. When SHE scheme’s depth-limit happens to behaving the greater depth that of a decryption circuit, decryption of ciphertext can be done homomorphically, this can be done by exposing a novel ciphertext which accepts several other forms of homomorphic operations. Homomorphic decryption of this category is termed as bootstrapping. Using a lattice construction, Gentry instantiated the SHE schemes that was suggested by him. This is a common primitive among several other types of homomorphic encryption constructions.

C) **Garbled Circuits**

The garbled circuit SMC protocol that is generally used in SMC was developed by A. Yao[26] and provided the first protocol for evaluating arbitrary functions in privacy-preserving manner. Using only a symmetric encryption scheme and an oblivious transfer protocol, Yao's protocol allows functions represented as boolean circuits to be obliviously evaluated in a constant number of communication rounds. The protocol requires at least two participants. The first is the generator, who is responsible for obscuring the bit values and $\delta$ gate functionality for the chosen function will be evaluated. Then a set of garbled input values and the garbled circuit are given as input to evaluator and evaluator is responsible for obliviously evaluating the circuit. The protocol proceeds as follows:

1. **Garbling**: For every wire present in the circuit, the generator creates two random strings $w_0^i$ and $w_1^i$, which are the garbled wire labels for the bit values of 0 and 1 on the $i^{th}$ wire. Next, he garbles each gate by encrypting the entries in a truth table that corresponds to the gate’s functionality. For simplicity, only two input gates are considered, but the same operations can be used to garble a gate with any number of inputs or outputs. For a gate executing the arbitrary boolean operation $\star$ which takes input wires $i$ and $j$, and outputs on wire $k$, it encrypts each entry as:

$$\text{Enc}_{\text{key}}(w_{i}^{b_i}, w_{j}^{b_j})^{b_k}$$

(2)

where $b_i$ and $b_j$ are the logical bit values for wires $i$ and $j$ respectively. After permuting the entries in each garbled truth table, the generator sends all of the garbled gates and the input wire values that correspond to his secret input to the evaluator.

2. **Oblivious Transfer**: For each of the evaluator’s secret input bits $b_i$, the generator and evaluator executes a 1-out-of-2 oblivious transfer. This protocol allows the evaluator to receive the garbled wire label $w_i^{b_i}$ without the generator learning the secret value $b_i$. Moreover, the evaluator learns nothing about the wire label $w_i^{1-b_i}$. 

fully trusted third party. SMC constructions can be divided into three broad categories based on working technique: Secret-Sharing, Homomorphic Encryption, and Garbled Circuits.
3. Evaluation: Given the garbled input values for both parties and the garbled gates, the evaluator can perform oblivious evaluation of the circuit. For each gate, SHE possesses the correct wire labels to decrypt exactly one entry in the garbled truth table, which allows her to decrypt subsequent entries in subsequent gates.

4. Output: Once the evaluator is done with possessing wire labels for each of the output wires, she can deliver these garbled wire values back to the generator, who can recover the output using his original mappings between garbled wire labels and logical bit values. The generator can then optionally deliver this output to the evaluator.

CONCLUSION AND FUTURE WORK

The increasing use of internet have been making the people dependent upon the cyber world. Privacy-preserving computation techniques are necessary for users to compute over internet jointly without any threat to their confidential information. Secure Multi-party Computation is an important technique for such purpose. SMC is progressing very fast, almost all the joint computation into the secureperforming form. Solving a normal form of cooperative computation is generally known to us, from that SMC can be applied to all this type of function to convert them into the privacy-preserving computation. With the discussed SMC problem, during any necessity, they can work in other computational areas in solving certain critical computational problems. During this work, we focused on bringing this aspect to the knowledge of researchers requiring their usefulness.

It is expected that, on working with various such problems, we acquire deep knowledge and skills in solving similar problems. Some of the useful building blocks while solving similar problems are, solutions to an existing problem, it could significantly help to carry out problems of a secured multi-party computation work.

REFERENCES