SECURE MULTI-PARTY COMPUTATION - A SURVEY

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

ABSTRACT-This is particularlytrue if the data from several sources are combined for some common task may be beneficial and speed up the task. For example, by combining datafrom several of its institutions, a state can discover trends or pin-point problematicissues.However, this is often avoided due to privacy concerns, as the combineddata set becomes an attractive target for both insideand outside attackers. In this internet era, privacy and security of data arehighly demanded and it is a sensitive issue. Secure Multi-Party Computation (SMC) is a technology which permits data to be processed with privacyso that the computation servers could not seeany actual data values. Firstpractical implementations of this emerging technology is started in the 2000s.This technology is now matureenough to be used for privacy-preserving data analysis on real data. SMCease 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 ata 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 trustthe 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 twoor 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.

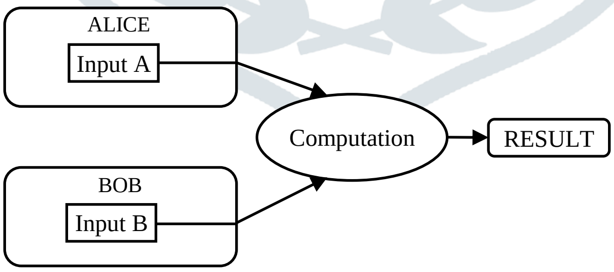

Fig 1: The Block Diagram of Multiparty Computation Model

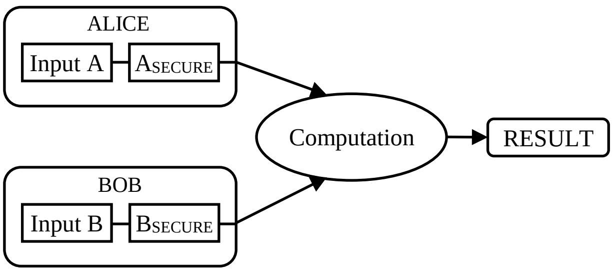

Fig 2: The Block Diagram of Secure Multiparty Computation Model

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 aprivacy-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. Goldreichet 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 short protocol depends on the size of the input domain, and on the complexity of expressing such a computation.

According to literature, many researchers haveworked to convert traditional computation into secure multi-party computation by applyingany secure mechanism to preserve the privacy of data. Privacy is 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 other's 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 pointsP_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 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 datasetD_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 twoor more bilaterallynon-trusting parties to compute a shared output while keeping their respective inputs hidden from the other participating parties. A number of different techniqueshave been used to achieve this impressive security guarantee.

This is a technology that allows joint computation on data from the different parties insuch a way that the data remains private while only the intended result becomes known. Moreprecisely, MPC allows a number of parties $P_1, P_2, P_3, \dots, P_n$ with input $x_1, x_2, x_3, \dots, x_n$ to computea function $(y_1, y_2, y_3, \dots, y_n) = f(x_1, x_2, x_3, \dots, x_n)$ while guarantee interesting security propertiessuch as input and output privacy (that the input$x_i$and output$y_i$remain private to each partyP_i) and correctness (that only $f$is evaluated and not some other function f' is evaluated), even if some of theparties may be acting maliciously.

A. Yao[1] developed the first protocol for SMC in 1982. Currently, research in secure multipartycomputation 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. SMCgenerally describes any techniquein a privacy-preservingway for evaluating some functionality. The security provided by these schemes ismost commonly demonstrated using areal/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 idealworld where the function is evaluated by a trusted third party, who then distributes someoutput to all participants[10]. While many variations on this paradigm are used to provethe security of different constructions, this basic concept instinctively demonstrates that theSMC protocol provides equivalent security to the best achievable solution of a fully trustedthird party.SMC constructions can be divided into three broadcategories based on working technique: Secret-Sharing, Homomorphic Encryption, and Garbled Circuits.

A) Secret-Sharing

Secret-sharing SMC techniques encompass a variety of protocols that allow to split private input into encrypted parts and shared between the parties[11]. These shares arethen processed in a privacy-preserving interactive protocol and combined at the end torecover 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 combinedwith the secret shares during the online computation to make it feasible for non-linear operationssuch as multiplication 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 tobe executed for computing AND gates. Recentdevelopments 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 underlyingsecret sharing protocols[14][15][16]. These optimizations have been augmentedwith further research into constructing optimal arithmetic circuit representations for common functions[17][18]. However, secret-sharing protocols are generally found optimal for largenumbers of participants and tend to be less practical in the two-party setting.

B) Homomorphic Encryption

Homomorphic encryption usesseveral encryption schemes that perform severalhomomorphic operations to the wholecipher value. Especially, given an encryption technique$\left(Gen(\cdot), Enc_{key}(\cdot), Dec_{key}(\cdot)\right)$, two messages M_a, M_b holds some operation ◆ over normal text and operation ★ over encrypted value that is given as:

$$Decrypt_{key}\left(Encrypt_{key}(M_a) \bigstar Encrypt_{key}(M_b)\right) = M_a \blacklozenge M_b \qquad (1)$$

For instance,the homomorphic additive scheme turns it to an additive operation, allowing the plaintext messages to get added still remainingas encrypted. This process allows for a single homomorphic operationwhich 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-purposeprotocols[20][21][22][23]to sort out a particular function in a suitablefeasible way. Although, generic computation of encrypted type data is not possible if an operation is not supporting universal homomorphic operation sets.

In order to examine the arbitrary functions from encrypted data, Rivest et al.[24] suggested a basic fully-homomorphicencryption (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, additionis required to allow.Secondly, 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 noble scheme for achieving those desired properties, it used the concept of bootstrappableencryption plan.Thebasisof itis dependent onthe use of basic Somewhat Homomorphic Encryption (SHE), this scheme allows a universal set of homomorphic operations, although it is only limited toevaluation of limited depth circuits. When SHEscheme's depthlimit happens to behaving the greater depth that of a decryption circuit, decryption ofciphertext can be done homomorphically, this can be done by exposing a noble ciphertext which acceptsseveral other forms of homomorphic operations. Homomorphic decryption of this category is termed as bootstrapping.Using a lattice construction, Gentry instantiated the SHE schemes that wassuggested by him. This is a commonprimitive amongseveral other types of homomorphic encryption constructions.

C) Garbled Circuits

The garbled circuit SMC protocol that is generally used in SMC was developed byA. Yao[26] and provided the first protocol for evaluating arbitrary functions in aprivacy-preserving manner. Using only a symmetric encryption scheme and an obliviousransfer protocol, Yao's protocol allows functions represented as boolean circuits to beobliviously evaluated in a constant number of communication rounds. The protocol requiresat least two participants. The first is the generator, who is responsible for obscuring the bitvalues and0 gate functionality for the chosen function will be evaluated. Thena set of garbled input values and the garbled circuit are given as input to evaluater and evaluator is responsible for obliviouslyevaluating the circuit.The protocol proceeds as follows:

1. Garbling: For every wire present in the circuit, the generator creates two random stringsw_i^0 and w_i^1, 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 ★ which takes input wires i and j, and outputs on wire k, it encrypts each entry as:

$$Enc_{w_i^{b_i} \| w_j^{b_j}}\left(w_k^{b_i \bigstar b_j}\right) \qquad (2)$$

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 $w_i^{1-b_i}$.

3. Evaluation: Given the garbled input values for both parties and the garbled gates, the evaluator can performan 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 outproblems of a secured multi-party computation work.

REFERENCES

[1] A. Yao, "Protocols for secure computations," in *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*, 1982.
[2] O. Goldreich, S. Micali and A. Wigderson, "How to play any mental game," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, 1987.
[3] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Annual International Cryptology Conference*, 2000.
[4] M. Atallah and W. Du, "Secure multi-party computational geometry," in *Workshop on Algorithms and Data Structures*, 2001.
[5] W. Du and M. Atallah, "Privacy-preserving cooperative scientific computations," in *csfw*, 2001.
[6] G. Aggarwal, N. Mishra and B. Pinkas, "Secure computation of the k th-ranked element," in *International Conference on the Theory and Applications of Cryptographic Techniques*, 2004.
[7] M. Rabin, "How To Exchange Secrets with Oblivious Transfer," in *IACR Cryptology ePrint Archive*, 2005.
[8] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan, "Private information retrieval," in *n Foundations of Computer Science, Proceedings., 36th Annual Symposium*, 1995.
[9] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single database, computationally-private information retrieval.," in *In Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium*, 1997.
[10] G. Oded, "Foundations of Cryptography: Volume 2, Basic Applications," 2009.
[11] A. Shamir, "How to share a secret," in *Communications of the ACM*, 1979.
[12] M. Naor and B. Pinkas, "Oblivious transfer and polynomial evaluation.," in *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, 1999.
[13] J. Nielsen, P. Nordholt, C. Orlandi and S. Burra, "A new approach to practical active-secure two-party computation.," in *Advances in Cryptology–CRYPTO 2012*, 2012.
[14] P. Pullonen, D. Bogdanov and T. Schneider, "The design and implementation of a two-party protocol suite for Sharemind 3," CYBERNETICA Institute of Information Security, Tech. Rep, 2012.
[15] D. Beaver, "Efficient multiparty protocols using circuit randomization.," in *Annual International Cryptology Conference*, 1991.
[16] M. Atallah, M. Bykova, J. Li, K. Frikken and M. Topkara, "Private collaborative forecasting and benchmarking.," in *Proceedings of the 2004 ACM workshop on Privacy*, 2004.
[17] O. Catrina and S. De Hoogh, "Improved primitives for secure multiparty integer computation.," in *International Conference on Security and Cryptography for Networks*, 2010.
[18] O. Catrina and A. Saxena, "Secure computation with fixed-point numbers.," in *International Conference on Financial Cryptography and Data Security*, 2010.
[19] R. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems.," in *Communications of the ACM 21, no. 2*, 1978.
[20] M. Osadchy, B. Pinkas, A. Jarrous and B. Moskovich, "Scifi-a system for secure face identification.," in *Security and Privacy (SP), 2010 IEEE Symposium*, 2010.
[21] J. Bringer, H. Chabanne, M. Favre, A. Patey, T. Schneider and M. Zohner, "GSHADE: faster privacy-preserving distance computation and biometric identification.," in *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*, 2014.
[22] C. Hazay and Y. Lindell, ""Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries.," in *Theory of Cryptography Conference*, 2008.
[23] A. Miyaji and M. Rahman, "Privacy-preserving data mining in presence of covert adversaries," in *International Conference on Advanced Data Mining and Applications*, 2010.
[24] R. Rivest, L. Adleman and M. Dertouzos, "On data banks and privacy homomorphisms.," in *Foundations of secure computation 4, no. 11*, 1978.
[25] C. Gentry, "Fully homomorphic encryption using ideal lattices.," in *Annual ACM Symposium on Theory of Computing.*, 2009.
[26] A. C. Yao, "How to generate and exchange secrets," in *Proceedings of the IEEE Annual Symposium on Foundations of Computer Science*, 1986.
[27] W. Du and M. Atallah, "Secure multi-party computation problems and their applications: a review and open problems," in *Proceedings of the 2001 workshop on New security paradigms*, 2001.
[28] Y. Gertner, Y. Ishai, E. Kushilevitz and T. Malkin, "Protecting data privacy in private information retrieval schemes," *Journal of Computer and System Sciences*, vol. 60, no. 3, pp. 592-629, 2000.