Secure Multiparty Computation of a Social Network

Varsha Bhat Kukkala*, Jaspal Singh Saini† and S.R.S. Iyengar‡
Department of Computer Science, Indian Institute of Technology Ropar
Punjab, India - 140001
Email: *varsha.bhat@iitrpr.ac.in, †jaspal.singh@iitrpr.ac.in, ‡sudarshan@iitrpr.ac.in

***Abstract*—The society today is better connected as a result of advancement in technology. The study of these social interactions and its resulting structure, is an integral component in the field of network science. However, the study of these social networks is limited to the availability of data. Privacy concerns restrict the access to network data with sensitive information. Networks that capture the relations such as trust, enmity, sexual contact, are a few examples of sensitive networks. A study of these sensitive networks is important in unraveling the behavioral aspects of the concerned individuals. The current paper proposes a multiparty computation algorithm that allows the construction of the unlabeled isomorphic version of the underlying network. The algorithm is information theoretic secure and works under the malicious adversarial model with the threshold of one third total corrupt parties.**

***Index Terms*—Multiparty computation, Social networks, Information theoretic security**

I. INTRODUCTION

The concept of a social network is no more limited to the field of sociology, where it was first introduced. Today, the idea is widely applied to a myriad of domains, such as biology, chemistry, marketing, economics and epidemiology. The presence of common topological characteristics, across different networks, is the reason for its wide applicability. Social networks are modeled as graphs with social entities represented as nodes and the edges of the graph capturing the interrelationship between the entities. Some of the most frequently studied networks include friendship networks (both online as well as offline), human-contact networks, communication networks, citation networks, etc. Studying these networks has helped in making several observations and better the understanding of phenomena such as information cascades and communication patterns, spread of diseases [1], influence [2], etc. Several online social networks, like Facebook, Twitter and LiveJournal, have constantly been a playground for analyzing the network structure and its implications. However, a social network innately houses sensitive information of the concerned individuals. The resulting privacy concerns have been a major impediment to the study of such networks.

A study of sensitive interrelationships like trust, hatred, sexual contact, etc., can have an unforeseen impact on our understanding of the behavioral patterns observed across individuals. For example, the amount of hatred fostered in a team can have correlations to the team's overall productivity. However, gathering data of the hate network would be a challenge, as individuals would not be willing to reveal the sensitive information about the team members they dislike. A surveillance of the trust network, over time, in a defense institute can better form military teams and select team leaders. Most commonly, studies conducted on networks with sensitive information, acquire data through surveys [3] and then anonymize it. The fear of sensitive information being leaked prevents most of the users from sharing their private information [4], [5] or could even lead to reporting false information. To address these drawbacks of surveys, there is a necessity for constructing a protocol that can generate the underlying network while guaranteeing the privacy of the participating individuals.

We require a protocol that generates the underlying network securely, by amalgamating the data that is available distributedly. It must be done in a way that privacy and integrity of the inputs is ensured while guaranteeing the correctness of the generated output. This is precisely what constitutes a multiparty computation (MPC). It involves a set of parties, who follow a protocol (specific sequence of instructions) for computing a function of their private data. The process must ensure that nothing but the final result is revealed. The first MPC protocol was proposed by Yao [6], which allowed two parties to compare and determine who among the two is richer, without revealing each other's wealth. Multiparty computation has evolved as a separate branch of cryptography, whose tools and techniques have been used in addressing numerous problems, such as, computing approximations on distributed data, auctions, private matching and set intersection, secure rank computation, privacy preserving data classification and data mining.

The contribution of the current work is to provide a graph construction protocol that is proven to be correct, private and robust, in accordance to the requirements of a standard MPC protocol. The proposed solution is secure in the information theoretic setting with a threshold of $n/3$ corrupt parties. The protocol allows the construction of the unlabeled isomorphic version of the weighted directed graph on n individuals, who are participating in the protocol. Each individual, henceforth referred to as a party, reports her adjacency list (all her outgoing edges) as her private input. The algorithm guarantees that the parties do not learn any additional information apart from the data that can be gathered from just their input and output. The proposed protocol can be easily modified to construct the unweighted undirected graph as well.

II. RELATED WORK

General protocols have been proposed for securely evaluating any computable function [7], [8], [9]. These theoretical models cannot be put to practical use due to the large blow

up in the communication and computation overheads involved. Thus, problem specific efficient protocols are constructed [10], [11].

Securely computing algorithms on a network has been studied in the recent past. Brickell and Shmatikov [12] look at two party protocols for computing all pair shortest paths and single source shortest paths securely, in the cryptographic and the semi-honest adversarial model. In this protocol, each party possesses a graph, such that both the parties are interested in computing algorithms over the union of the two input graphs. Hu, Chow and Lau [13] discuss on how one can detect people belonging to the same community with minimum information being leaked. Such a detection allows to suggest friends in a social network. Zeng et al. [14] also propose a technique for secure link prediction in online social networks. Aly et al. [15] study the problem of computing shortest paths in a graph securely. Aly and Vyve [16] address the problem of finding minimum mean cycle and the minimum cost flow problems in a multiparty setting. Blanton et al. [17] propose a data oblivious method for computing graph algorithms, such as BFS, shortest paths, minimum spanning tree and network flow problems.

Securely generating the underlying graph has been previously studied by Frikken and Golle [18]. It is assumed that the network information is held in a distributed manner, where each individual possesses some partial information of the network. The drawbacks of the protocol is that it uses special parties called authorities, who help compile the collected data into the required graph. Also, the use of threshold Elgamal encryption scheme and re-encryption mix nets in the protocol, amounts to increased communication and computation cost. It is to be noted that their protocol is restricted to the cryptographic security model. The protocol proposed in the current paper avoids the use of dummy parties and is information theoretically secure, thereby overcoming the above mentioned drawbacks. Bhat et al. [19] propose an information theoretic solution for compiling an isomorphic version of a distributedly held graph in the semi-honest setting with a threshold of $\sqrt{n}$ corrupt parties. The protocol proposed in the current work can withstand up to $n/3$ corrupt parties in the malicious adversarial model.

III. Preliminaries

A multiparty computation protocol is an algorithm, using which a set of n parties $P_1, P_2, \ldots, P_n$ can compute any function f over their private information *securely*. We will consider the field $\mathbb{F}_p$ (where p is a prime number) with the modular operations of addition ($+$) and multiplication ($*$). A set of parties are said to be *corrupt* if they collaborate to reveal information about the set of honest parties[1]. In order to model corruption, we assume the presence of a central adversary, who controls all corrupt parties. A protocol is said to be secure in the malicious adversarial model if it is *correct, private* and *robust*. A detailed discussion on security of MPC protocols is available in [20], however we briefly describe the security requirements below:

- Correctness: A protocol is said to be correct if the output of the protocol matches the required function evaluation on the private inputs of the parties.
- Privacy: A protocol is said to be private if the adversary learns nothing more than the inputs and outputs of the corrupt parties, during the run of the protocol i.e. all the information gathered by the corrupt parties during the run of the protocol can be efficiently computed using only the inputs and outputs of the corrupt parties.
- Robust: In a robust protocol, a set of corrupt parties do not gain any influence by deviating from the pre-described protocol i.e. any influence that a set of corrupt parties gain by deviating from the protocol, can also be achieved without any deviation from the pre-described protocol.

Further, a protocol is said to be information theoretic secure if it is secure in the presence of an adversary with unbounded computation power.

We use a verifiable secret sharing scheme (VSS) for secret sharing the private information of all the parties and then securely computing on it. A discussion on VSS schemes like Shamir secret sharing is available in [20]. In Shamir secret sharing, a party P shares a secret s in the following two step process:

- Party P selects a polynomial $f(x) = s + a_1x + a_2x^2 + \ldots + a_tx^t$, for some fixed t, where $a_i \in \mathbb{F}_p$ for all $1 \leq i \leq t$.
- Party P sends the value of the function f evaluated at i i.e. $f(i)$ to party P_i, for all $1 \leq i \leq n$.

Using Shamir secret sharing scheme has various advantages, a comprehensive account of which is available in [20]. The VSS implementation using Shamir secret sharing is information theoretic secure in the malicious adversarial model with less than $n/3$ corrupt parties.

The notation $[a]$ represents that the secret a is distributedly held by all the parties using a VSS scheme. Further we assume that the VSS scheme under consideration provides the following operations:

- Addition
  $[c] \leftarrow [a] + [b]$ i.e. c contains the sum $(a + b)$.
- Multiplication
  $[c] \leftarrow [a] * [b]$ i.e. c contains the product $(a * b)$.
- Comparison
  $[c] \leftarrow [a > b]$, where c contains 1 if $(a > b)$ else c contains 0.
- Equality
  $[c] \leftarrow [a = b]$, where c contains 1 if a equals b, else c contains 0.
- Release
  $a \leftarrow [a]$ implies that the distributedly held value a is released in public.

The work of [9] provides a secure implementation of addition and multiplication, using which [21] provided a secure implementation for comparison and equality operations.

[1]Any party that is not corrupt is said to be honest.

For an $n \times n$ matrix A, the notation $[A]$ signifies that all the entries of the matrix are distributedly held using a VSS scheme with the above mentioned properties. The release operation $A \leftarrow [A]$ signifies that each entry of the matrix A is released in public.

An adjacency matrix $A = (a_{ij})_{n \times n}$ can also be represented as a vector of adjacency lists i.e. $A = (v_i)_{n \times 1}$, where v_i is the i^{th} adjacency vector of A or the i^{th} row of matrix A.

IV. The Proposed Protocol

In this section, we provide a protocol for securely computing a random isomorphic unlabeled version of a network distributedlty held by a set of parties. The graph to be constructed may be distributedly held by the n parties in various forms. For example, each party may hold a row of the adjacency matrix as her private information. Such scenarios may arise in the case of trust networks, enmity networks and sexual networks, where each individual has her outgoing links as her private information. It may also be the case that each party may hold information about a subgraph in the network. Such scenarios may arise in the case of financial networks[2] and distributed social networks. The details of these various forms of inputs are discussed in [22]. In this paper, we assume that each party P_i possesses an adjacency vector v_i as her private input, such that the adjacency matrix A under consideration equals $(v_i)_{n \times 1}$. This protocol can easily be extended for other input forms of parties, using the functionalities available in [22].

Further we breifly describe the proposed protocol $isomorphic\_graph\_construction()$. The protocol starts by constructing $[A]$ i.e. a distributedly held matrix in steps 1-3, which implies that all the entries of the matrix A are distributedly held. In steps 4-14, we assign a unique random number r_i to each party P_i, for all $1 \leq i \leq n$. The r_i values are distributedly held by the n parties such that no party learns any of the r_i values. The above step is implemented by assigning a random number to each party, and then checking if any two random numbers match. If yes, we repeat this procedure until we find a unique sequence of random numbers. In steps 15-19, we calculate a random permutation $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ of the set $\{1, 2, \ldots, n\}$. We do so by assigning σ_i as the cardinality of the set $\{r_j | r_i > r_j, 1 \leq j \leq n\}$. Next we permute the adjacency matrix A to construct another matrix A'', such that $(i,j)^{th}$ entry of matrix A is set as the $(\sigma_i, \sigma_j)^{th}$ entry of A''. We do so by constructing a matrix A' from A, which in turn helps in constructing the matrix A''. In steps 21-24, we construct the matrix A' from A, such that $(i,j)^{th}$ entry of matrix A is set as the $(\sigma_i, j)^{th}$ entry of A'. In steps 25-28 we construct the adjacency matrix to be output A'' from the matrix A' using a column shuffle operation i.e. $(\sigma_i, j)^{th}$ entry of matrix A' is set as the $(\sigma_i, \sigma_j)^{th}$ entry of A''. Hence, A' is obtained by permuting the rows of A, while A'' is obtained by permuting the columns of A'. Here, both the row and column permutations are with respect to σ.

[2]The financial network is distributedly held between a set of banks.

Protocol 1 $isomorphic\_graph\_construction()$

1: **for** i = 1 to n **do**
2: **for** j = 1 to n **do**
3: Party P_i shares a_{ij}
4: **for** i = 1 to n **do**
5: **for** j = 1 to n **do**
6: Party P_i shares r_{ij}
7: **for** i = 1 to n **do**
8: $[r_i] \leftarrow \sum_{j=1}^{n} [r_{ji}]$
9: $[flag] \leftarrow 0$
10: **for** i = 1 to n-1 **do**
11: **for** j = i+1 to n **do**
12: $[flag] \leftarrow [flag] + [r_i = r_j]$
13: **if** flag != 0 **then**
14: goto Step 4
15: **for** i = 1 to n **do**
16: $[\sigma_i] \leftarrow 1$
17: **for** i = 1 to n **do**
18: **for** j = 1 to n **do**
19: $[\sigma_i] \leftarrow [\sigma_i] + [r_i > r_j]$
20: $[A'] \leftarrow [0]_{n \times n}$, $[A''] \leftarrow [0]_{n \times n}$
21: **for** i = 1 to n **do**
22: **for** j = 1 to n **do**
23: **for** k = 1 to n **do**
24: $[a'_{jk}] \leftarrow [a'_{jk}] + [\sigma_i = j] * [a_{ik}]$
25: **for** i = 1 to n **do**
26: **for** j = 1 to n **do**
27: **for** k = 1 to n **do**
28: $[a''_{kj}] \leftarrow [a''_{kj}] + [\sigma_i = j] * [a'_{ki}]$
29: $A'' \leftarrow [A'']$

The running time of the protocol depends on the number of times the goto statement in step 14 of the protocol is executed. To analyze the same, we define a random variable X to represent the number of times the goto step is executed. Let exp represent the exponential operator.

Lemma 1. $E[X] \leq exp\ (n^2/p)$

Proof. The random variable X is a geometric random variable with the probability of success equal to $\prod_{i=1}^{n-1} (1 - i/p)$.

$$\implies E[X] = \left(\prod_{i=1}^{n-1} \left(1 - \frac{i}{p} \right) \right)^{-1}$$

$$\implies E[X] \leq \left(1 - \frac{n}{p} \right)^{-n}$$

$$\implies E[X] \leq exp\ (n^2/p)$$

□

Hence, the expected number of times that the goto step is executed can be made smaller than any constant number. This is achievable by using a sufficiently large prime number p, for

a given n. The proposed protocol makes the above assumption regarding the field size p.

Next we analyze the distribution of the permutation $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ generated by reassigning labels to all the vertices in the network. Let S_n represent the permutation set consisting of all $n!$ permutations of the set $\{1, 2, \ldots, n\}$.

Lemma 2. *$\sigma \in_R S_n$ i.e. σ is a randomly generated permutation of the set $\{1, 2, \ldots, n\}$*

Proof. This follows directly from the fact that the sequence of numbers $(r_1, r_2, \ldots, r_n)$ are guaranteed to be unique random numbers and by the definition of σ_i as the cardinality of the set $\{r_j | r_i > r_j, 1 \leq j \leq n\}$. □

Theorem 1. *The proposed isomorphic graph construction protocol is information theoretic secure under the malicious adversarial model with less than $n/3$ corrupt parties.*

Proof. The correctness of this protocol follows from Lemma 2 and the fact that in steps 21-24 we obtain a row permuted matrix A' and in steps 25-29 we column permute A' to obtain the isomorphic network A'' with respect to the permutation σ. The privacy and robustness of the proposed protocol follows directly from the privacy and robustness of the VSS scheme under consideration. □

The computation cost for constructing a random isomorphic network in a non-secure manner for a given n node network would be at least $\Theta(n^2)$. This is because we would need to access each entry of the adjacency matrix, at least once, for constructing a random isomorphic version of it. As shown below, our protocol for computing an isomorphic version uses $\Theta(n^3)$ operations, which has an extra factor of n compared to the non-secure variant.

Theorem 2. *The proposed isomorphic graph construction protocol on an average uses $\Theta(n^3)$ addition operations, $\Theta(n^3)$ multiplication operations and $\Theta(n^3)$ comparison/equality check operations.*

Proof. This follows directly from the structure of the proposed protocol and the fact that the goto step in the protocol is executed a constant number of times on an average, for a sufficiently large field size p. □

V. Conclusion

In this paper, we propose a multiparty computation protocol for securely constructing an unlabeled random isomorphic version of a graph that is distributedly held by a set of n parties. The proposed protocol is information theoretic secure in the malicious adversarial model, tolerating less than $n/3$ corrupt parties. The proposed protocol can be used to study the behavioral aspects of individuals while guaranteeing the privacy of their sensitive data. Before releasing sensitive data in public, the data is generally anonymized. The current work performs naive anonymization, on a distributedly held network, without the use of a trusted third party. One can further implement multiparty computation protocols for network specific anonymization techniques.

References

[1] M. Salathé, M. Kazandjieva, J. W. Lee, P. Levis, M. W. Feldman, and J. H. Jones, "A high-resolution human contact network for infectious disease transmission," *Proceedings of the National Academy of Sciences*, vol. 107, no. 51, pp. 22 020–22 025, 2010.

[2] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in twitter: The million follower fallacy." *ICWSM*, vol. 10, no. 10-17, p. 30, 2010.

[3] S. Helleringer and H.-P. Kohler, "Sexual network structure and the spread of hiv in africa: evidence from likoma island, malawi," *Aids*, vol. 21, no. 17, pp. 2323–2332, 2007.

[4] J. Black, "The perils and promise of online schmoozing," *BusinessWeek Online, February*, vol. 20, p. 2004, 2004.

[5] L. Garton, C. Haythornthwaite, and B. Wellman, "Studying online social networks," *Journal of Computer-Mediated Communication*, vol. 3, no. 1, pp. 0–0, 1997.

[6] A. C. Yao, "Protocols for secure computations," in *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*. IEEE, 1982, pp. 160–164.

[7] 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*. ACM, 1987, pp. 218–229.

[8] D. Chaum, C. Crépeau, and I. Damgard, "Multiparty unconditionally secure protocols," in *Proceedings of the twentieth annual ACM symposium on Theory of computing*. ACM, 1988, pp. 11–19.

[9] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proceedings of the twentieth annual ACM symposium on Theory of computing*. ACM, 1988, pp. 1–10.

[10] Z. Yang, S. Zhong, and R. N. Wright, "Privacy-preserving classification of customer data without loss of accuracy." in *SDM*. SIAM, 2005, pp. 92–102.

[11] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Advances in CryptologyCRYPTO 2000*. Springer, 2000, pp. 36–54.

[12] J. Brickell and V. Shmatikov, "Privacy-preserving graph algorithms in the semi-honest model," in *Advances in Cryptology-ASIACRYPT 2005*. Springer, 2005, pp. 236–252.

[13] P. Hu, S. S. Chow, and W. C. Lau, "Secure friend discovery via privacy-preserving and decentralized community detection," *arXiv preprint arXiv:1405.4951*, 2014.

[14] Y. Zheng, B. Wang, W. Lou, and Y. T. Hou, "Privacy-preserving link prediction in decentralized online social networks," in *Computer Security–ESORICS 2015*. Springer, 2015, pp. 61–80.

[15] A. Aly, E. Cuvelier, S. Mawet, O. Pereira, and M. Van Vyve, "Securely solving simple combinatorial graph problems," in *Financial Cryptography and Data Security*. Springer, 2013, pp. 239–257.

[16] A. Aly and M. Van Vyve, "Securely solving classical network flow problems," in *Information Security and Cryptology-ICISC 2014*. Springer, 2014, pp. 205–221.

[17] M. Blanton, A. Steele, and M. Alisagari, "Data-oblivious graph algorithms for secure computation and outsourcing," in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*. ACM, 2013, pp. 207–218.

[18] K. B. Frikken and P. Golle, "Private social network analysis: How to assemble pieces of a graph privately," in *Proceedings of the 5th ACM workshop on Privacy in electronic society*. ACM, 2006, pp. 89–98.

[19] V. B. Kukkala, S. Iyengar, and J. S. Saini, "Secure multiparty graph computation," in *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, 2016, pp. 1–2.

[20] R. Cramer, I. Damgard, and J. B. Nielsen, "Secure multiparty computation and secret sharing-an information theoretic approach," *Book draft*, 2012.

[21] I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, "Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation," in *Theory of Cryptography*. Springer, 2006, pp. 285–304.

[22] V. B. Kukkala, J. S. Saini, and S. Iyengar, "Network deprived sna: An alternative to anonymization."