IJRTS

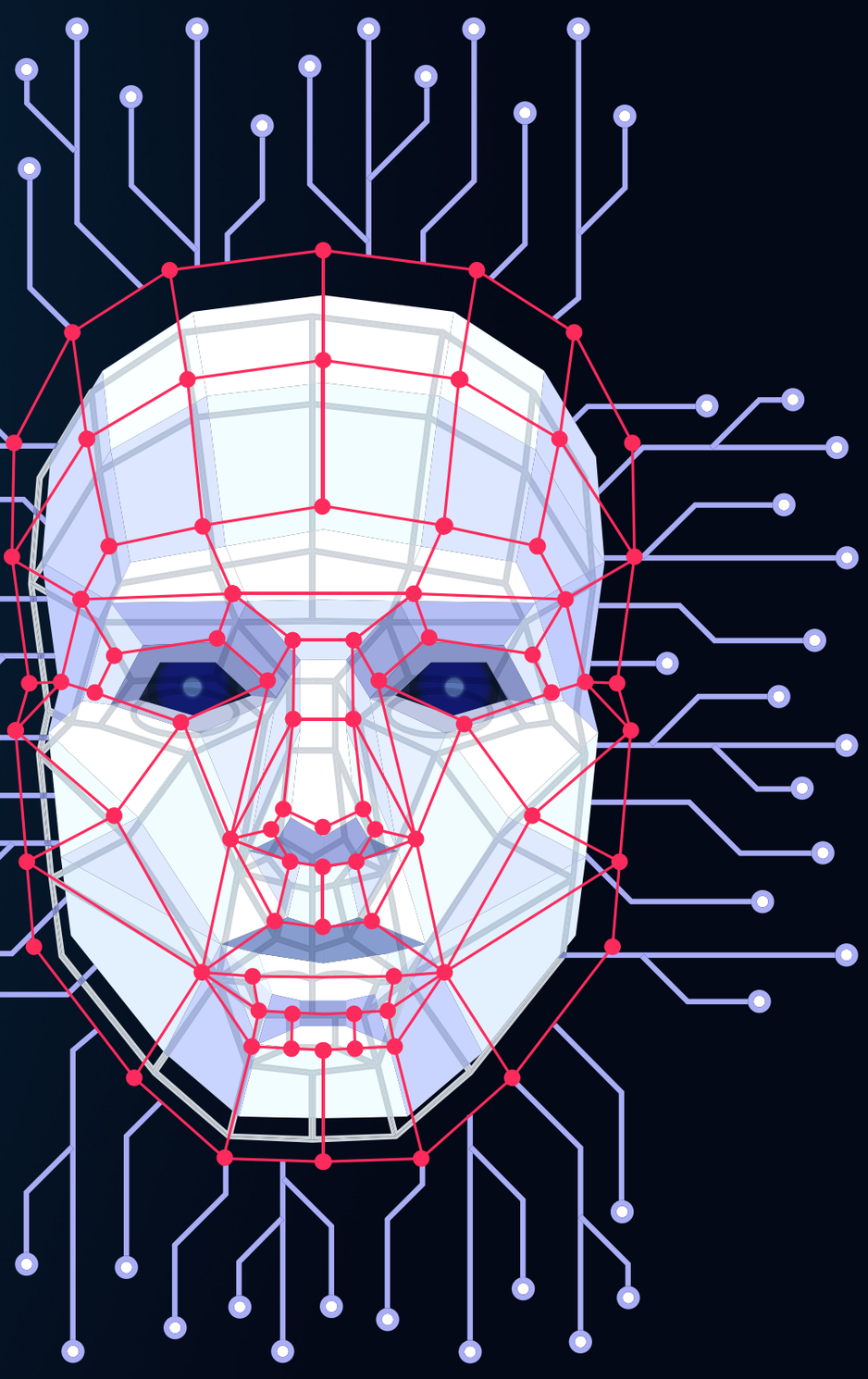
Publications

A Venture of IJRTS Trishtha Foundation

RECENT DEVELOPMENTS IN

Intelligence, Research & Technology

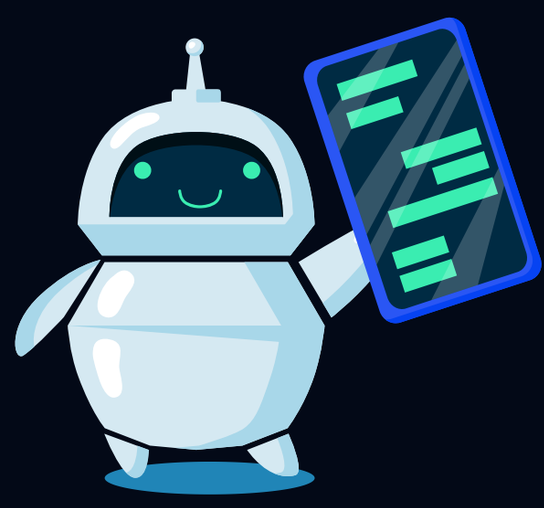
ISBN
978-81-964047-2-7



An Edited Book

Chief Editor

Dr. Vipin Mittal



Recent Developments in Intelligence, Research & Technology
An Edited Book
ISBN: 978-81-964047-2-7

A Study of the Safe Multiparty Equality Check with Homotopy Cryptosystem and the Design of Safe Multiparty Computing Protocols for Privacy

Dr. Satish Kumar

Associate Professor, Department of Computer Science
Government College Narnaul, Distt. Mohindergarh, Haryana, India
s9467723723@gmail.com

Abstract

The subject of information security where each party wants to compute some function of their private inputs without disclosing their inputs to one another is called Secure Multi-Party Computation (SMC). For example two banks want to perform certain data mining operation over union of their individual databases but no bank wants to disclose its private database to other. In such scenario SMC becomes relevant. we present our research work which is aimed to designing novel SMC protocols for preserving privacy of an individual data while joint computation is taking place. Due to huge growth of the Internet and its easy access by common man opportunities are increasing for cooperative computations by many physically distributed parties. All these parties have their own data and for the sake of mutual benefit, they want to compute some function of these data. Traditionally, the computation is easy; every one supplies the data input to get the value of the function. But as this model is growing the need of confidentiality of data is also increasing. Multiple parties want to know the result of computation but they are also concerned about secrecy of their data inputs. The subject of SMC has been evolved from two party comparison problems to multiparty image template matching problems. Many specific SMC problems have been defined and analyzed by researchers like Private Information Retrieval, Selective Function Evaluation, Privacy-Preserving Database Query, Privacy-Preserving Geometric Computation, Privacy-Preserving Statistical Analysis, Privacy-Preserving Intrusion Detection and Privacy-Preserving Cooperative Scientific Computation. Based on these general SMC problems many real life applications emerged like Privacy-Preserving Electronic Voting, Privacy-Preserving Bidding

and Auctions, Privacy-Preserving Social Network Analysis, and Privacy-Preserving Signature and Face Detection. Many protocols assume the network to be secure while devising SMC algorithms. In this work we proposed novel and improved secure sum computation algorithms, and equality check algorithms which are application to insecure networks. We used public key cryptography techniques to hide the value of the data while it is being travelling on network lines. We also used additive homomorphic public key cryptosystem where the sum of encrypted values is same as the encryption of the sum. Another technique we used in our algorithms is the Shamir's Threshold encryption scheme which allows a value to be reconstructed out of certain minimum number of shares of the value. Our proposed secure sum algorithms and secure equality check algorithms are based on real model as well as Ideal model of SMC. In one of the proposed algorithms we coin the term Semi-Ideal model as it is a hybrid of ideal and real model of SMC. The reason behind working on the secure sum is that all other arithmetic operations can be implemented using addition operation. Secure equality is also used in many database searches as well as in the process of authentication. Our works are suitable for honest but curious parties. Further, the algorithms can be devised which are suitable for malicious parties.

INTRODUCTION

Present age is rapidly becoming the "electronic" or "online" age. Almost every transaction of our day-to-day life involves electronic transactions. May it be transferring money by the bank or individual from one account to another account. We are almost in the age of e-banking. Due to the huge growth of e-commerce, orders are being placed online, whether it is business-to-business or business-tocustomer. These orders are being processed using Supply Chain Management (SCM) software and the database is maintained in association with Customer Relationship Management (CRM) software. The modules such as SCM and CRM are nowadays integrated in Enterprise Resource Planning (ERP) software. All these software are using the database distributed along multiple sites with the help of the Internet. Bidding and auction activities are also being performed online. The computation on the database may be done at a central site or it may be distributed at multiple physical sites in a distributed system. Owing to easy access of the Internet by common man, its fast transactions, and huge growth in every sector people are being dependent on it for almost all kinds of computations. A new paradigm is rapidly emerging where multiple parties jointly work and perform computation on their data but they are bothered about the privacy of their data while the

ISBN: 978-81-964047-2-7

computation is being performed by them. In other words these joint parties do not have full trust on each other. But for the sake mutual benefit they have to work in cooperation. These cooperating parties wish to evaluate some common function of their sensitive data but do not want to disclose individual data to one another. In other words it is a paradigm of cooperative computation of multiple parties where all the parties are interested in knowing the result of joint computation on their data but they are unwilling to disclose actual value of their private data to other parties involved in the computation. All want that the privacy or confidentiality of their individual data must be preserved while the computation is being performed. Therefore it is a paradigm of privacy preserving computation. This area of information security where multiple cooperating parties wish to evaluate a common function of their data jointly such that the data is not disclosed to other parties is known as Secure Multiparty Computation (SMC). Consider a scenario where multiple banks jointly with to find common defaulters from their databases so that the list of wilful defaulters as well a list of defaulters from multiple banks can be prepared. This is in common interest of all the banks involved in the computation. But all the banks are concerned about the privacy of their database. As they are competitors in the same sector, they don't want to disclose their database to other banks. In this scenario some protocol is needed where all the banks may know the list of common defaulters keeping their database secret.

Many specific protocols for SMC. Few of them are listed below:

1. Private Information Retrieval (PIR)
2. Selective Function Evaluation
3. Privacy-Preserving Database Query
4. Privacy-Preserving Geometric Computation
5. Privacy-Preserving Statistical Analysis
6. Privacy-Preserving Intrusion Detection
7. Privacy-Preserving Cooperative Scientific Computation

Since the inception of SMC many interesting and real life applications emerged. Few of them are listed below:

1. Privacy-Preserving Electronic Voting
2. Privacy-Preserving Bidding and Auctions
3. Privacy-Preserving Social Network Analysis
4. Privacy-Preserving Signature and Face Detection

ISBN: 978-81-964047-2-7

Early research of the SMC focussed on circuit evaluation techniques using combinational logic circuits. But these techniques were complex and expensive. Nowadays there are three main techniques used to devise SMC protocols:

1. Cryptographic SMC methods
2. Randomization SMC Methods
3. Anonymization SMC methods

Other set of protocols in these are used to check the equality of data with multiple parties keeping the private data secret. These protocols can be used for matching or verification of a particular data without violating or hurting the privacy of actual data. We propose a novel multiparty equality protocol using homomorphic cryptosystem. The protocol is also suitable for insecure networks as the data flow in encrypted form. We have proposed Secure MultiEqualityCheck protocol based on homomorphic public-key cryptosystem [10]. In another protocol we used hashing function for generating hash code of the actual data. A hash function produces a fixed size value when a variable-size input is given. This is infeasible to produce the actual value when a hash code of that value is given. But if so happens it is called collision. We assume that there exists a collision-resistant hash function. Based on this two protocols EqualityHashCheck for ideal model and EqualityHashCheck for real model are proposed.

EXAMINING PROBLEMS

We have observed that many protocols for SMC assume the network to be secure. Therefore we propose novel protocols which are suitable insecure networks. This is achieved by using some cryptographic technique and not sending the data in its actual form. For example the secure sum protocol proposed by Cliften et al. [4] uses a random number that is added to the actual data and the partial sum is sent on the network lines. The parties are arranged in a ring. If some intruder intercepts both incoming and outgoing line of a party, the difference of the partial sum he gets is the actual data of the party. Similarly, if two neighbours to a middle party share what they send and what they receive, the difference is the actual data of the middle party. In actual practice this is possible as the lines are always insecure. We use cryptographic methods to design the protocols for insecure networks. In our protocols the above cases of data breach are not possible. In our secure sum protocols we encrypt the data with either symmetric or asymmetric encryption before sending on the lines. Thus, the privacy of the data against intruders and colluding neighbours is preserved. We proposed secure protocols both for real model and ideal model of SMC. In ideal model the

ISBN: 978-81-964047-2-7

trustworthiness of the third party is a concern. With homomorphic cryptosystem and threshold encryption scheme we ensure that the privacy of the data is also preserved against third party. We also proposed multiparty equality check protocols in ideal as well as real model. We use additive homomorphic encryption in one protocol and hash function in other two protocols. Thus, we have eight secure sum protocols and three multiparty equality check protocols proposed. All the protocols are suitable for semi-honest adversaries.

Novel secure sum protocols

Secure sum with symmetric key for insecure networks

This approach of getting secure sum is slight extension of the secure sum protocol of Cliften et al. [4]. All the parties are arranged in a logical ring. All the parties share a symmetric key in advance. A party is chosen as protocol initiator. It chooses a secret random number, and adds to its private data. Now, the party encrypts this sum with the shared key, and sends the encrypted value to the immediate neighbour party in the ring. The receiving party first decrypts the ciphertext to recover the partial sum, and adds its private data to the recovered sum. Now, it encrypts and sends in similar fashion as the previous party did. This process of receive, decrypt, add, encrypt, and send is repeated until the protocol initiator recovers the sum of all the data plus random number. It subtracts the random number to get the sum of actual data and distributes to all the parties. The improvement over the secure sum protocol of Cliften et al. [4] is that our protocol is suitable for insecure networks as the data flow in encrypted form, no intruder can learn the data or the partial sum. We analyse the computation and communication complexity of the protocol.

Secure sum with asymmetric key for insecure networks

The secure sum with symmetric key protocol provides an improvement over secure sum protocol. But there are two limitations with it. One, securely sharing the symmetric key. The key distribution on insecure network is itself is a serious problem. Another, as all parties share a common key and when the key is compromised, private of all the data will be violated. In order to overcome this drawback, we propose using public key cryptography with secure sum architecture. In the new protocol all the parties use the same ring topology used by previous protocols and communicate in only one direction. All the parties are required to generate a public-private key pair using certain algorithm like RSA. Each party must share its public key with the previous party in the ring. Thus, the protocol initiator has already got the public key of the next party. The private keys of all the parties are their secret. The protocol initiator party chooses a secret random number, and adds to its private data. Now, it is

ISBN: 978-81-964047-2-7

encrypted by the public key of the next party, and the ciphertext is sent to the immediate neighbour party. The party receiving the ciphertext first decrypts it with its private key. The partial sum is recovered as it was encrypted with the public key of the receiving party. Now, this party adds its data to this partial sum and encrypts with the public key of the next party, and sends this ciphertext to the next party. This process of receive, decrypt, add, encrypt, and send is repeated until the protocol initiator recovers the sum of all the data plus random number. It subtracts the random number to get the sum of actual data and distributes to all the parties. The improvement is that as there are different encryption and decryption keys at each point, the whole network cannot be compromised with compromising a single or multiple keys. Another, the distribution of public key can be done on insecure network, and public key is no longer a secret. But the cost of generation of public-private key pair at each party is a drawback.

Secure sum against colluding neighbours for insecure networks

Both the protocols secure sum with symmetric key and secure sum with asymmetric key can see colluding neighbours attack. To solve this problem we propose a protocol secure sum against colluding neighbours in which all parties except the initiator party encrypt with the public key of the initiator. Thus, the encrypted value at the initiator contains data with multiple encryptions with the public key of the initiator. When it is decrypted multiple times with the private key of the initiator, the secure sum is obtained.

Secure sum using homomorphic cryptosystem

We proposed a novel protocol in which we use the additive homomorphic property of a public key cryptosystem. The property ensures that the sum of ciphertexts is equal to ciphertext of sum of individual data. We use again propose to use the ring topology as used in the previous protocols. The concept of the initiator and random number is again applicable here. The new thing is that here the initiator party generates public-private key pair using some homomorphic public key cryptosystem. The random number and the private key are the secrets of the initiator but it distributes its public key to all cooperating parties who wish to compute privacypreserving sum. The initiator begins by adding random number to its secret data, and encrypting by its public key. It sends the encrypted value to the immediate party in the ring. The receiving party simply adds its public-key encrypted value to the received sum, and sends new sum to the next party. This process is repeated until the initiator receives sum of all the ciphertext encrypted with its public key. As per the homomorphic property it is equal to the ciphertext of the sum of data plus random number. When it is decrypted with the

private key of the initiator, secure sum plus random number is obtained. The initiator subtracts the random number to get the secure sum. The sum is distributed to all the parties.

Secure sum using threshold encryption

This approach of getting secure sum uses Shamir's threshold encryption scheme [6] which ensures that if a value is broken into shares; the same value can be reconstructed by adding certain minimum number of shares. We also use additive homomorphic cryptosystem based on public key cryptography. In this protocol we used a hybrid architecture of real as well as ideal SMC model. We coin the term Semi-ideal SMC model for this architecture. A Trusted Third Party (TTP) assists run the protocol. But there is a limited role of the TTP. It generates public-private key pair using additive homomorphic public key cryptosystem, and shares of the private key using Shamir's threshold encryption scheme. The TTP distributes public key and a share of the private key to all the parties. Now, remaining part of the protocol is run by the parties in real model. The parties first get the sum of ciphertexts of their private data with public key encryption. Now, parties recover the private key by getting sum of their private key shares. The protocol initiator decrypts the ciphertext sum with the recovered private key to get the actual secure sum. The sum is now announced to the parties.

Randomization approach for secure sum in ideal model

The protocol for secure sum proposed by Cliften et al. [4] and many other protocols in the literature employ real SMC model where no TTP exists. We proposed a simple secure sum protocol using randomization approach for ideal model. In uses a TTP to assist in secure sum computation. All the parties simply choose and agree on a common random number which is hidden from the TTP. The participating parties who seek to calculate the sum of their sensitive data multiply their data with the secret random number and provide their product to the TTP. The TTP computes sum of all such products and return back to all the parties. All the parties divide the received sum to get the actual sum of data. The privacy of data among parties is preserved as parties never communicate data with one another. The privacy of the data is preserved against TTP as it does not know the random number. The network lines can be insecure as the intruder cannot learn actual data as it is protected by the random number.

Novel Multiparty Equality Check protocols

Real life scenarios exist where multiple parties are interested to check the equality of their data. But the problem becomes crucial when these parties are worried about the privacy of their individual data. This problem can be solved with protocols of SMC. It motivated us to devise protocols for multiparty equality check. We proposed protocols using homomorphic

cryptosystem, and using hash function. We propose protocols for ideal as well as real SMC model.

Secure Multiparty Equality check using homomorphic cryptosystem

This approach of comparing data of multiple parties is an extension to two-party equality check problem. Two parties can check the equality of their data using homomorphic encryption preserving the privacy of their individual data. We call the protocol as EqualityCheck. For multiparty case the parties are arranged in a logical ring. This is a real model as no TTP is present. The protocol begins when all the parties generate their public-private key pair using homomorphic cryptosystem. A protocol initiator party check equality with the next party using EqualityCheck. If equality holds it checks the equality with next party. Thus, a pair-by-pair equality check continues until the protocol initiator is reached. If the equality holds at the end of the ring, the result is declared as "All are Equal" and if the equality breaks at any point in the ring, the result is declared as "All not Equal". The parties encrypt using public key and protect their data using a random number.

- To maintain privacy of individual data input while computation: The protocols discussed in this thesis allow parties to compute the sum of their inputs while maintaining their data privacy. We use additive homomorphic cryptosystem and threshold encryption to protect the privacy during computation. Our multiparty equality check protocols use encryption and hashes to protect the privacy during equality check. No party ever passes actual data to other party. Thus the privacy of the individual data is maintained.

- To get the correct result of the computation: Our proposed protocols in this thesis perform addition operation over the encrypted data of the parties. The decryption of this ciphertext is the sum of these actual data. Thus, we are able to provide the correct result to the parties while maintaining their privacy. Similarly, we get the result of equality check while preserving privacy.

- To reduce probability of data leakage: In this thesis protocols are designed keeping an eye over the colluding neighbors who want to learn the secret data of the middle party. We successfully reduced the probability of data leakage in case two neighbors collude to learn the data of a middle party in our protocols is almost nil due to use of cryptographic techniques and the hashes

ISBN: 978-81-964047-2-7

- To minimize communication cost: We aim to achieve the security with minimum communication cost. For the same reason we analysed our work for minimum communication. In real model we build unidirectional ring between the parties. The links are assumed to be insecure. In ideal model parties communicate with TP only reducing the communication

- To develop Semi-ideal model of SMC: We aim to develop semi-ideal model of SMC in which we use properties of both real model and ideal model to achieve privacy and correctness of the result. The term semi-ideal SMC model is coined by us in [9].Such a model is highly economical and less vulnerable to security threats.

- To develop secure sum algorithms: Many secure sum algorithms are developed and their performance is analyzed for semi honest parties

Scope of the work

The relevancy of the SMC solutions is enhancing day-by-day because a large volume of cooperative computations are being taking place over data of many parties. Many organizations jointly work on a single project and they frequently share their sensitive information. But each of the organizations is also worried about the privacy of its data. In such scenario SMC solutions play significant role. Secure sum computation is an important example as well as the component of the toolkit for the SMC solution. Equality of data with many joint parties could be of real life use. In this thesis we proposed protocols and the corresponding algorithms for the secure sum computation and secure equality check.

- ❖ Banking Sector: Nowadays many banks work cooperatively and frequently share important information about their customers. SMC solutions allow each of the banks to perform cooperative computations while keeping their individual information a secret. For example the bank may wish to get details of common defaulters from their individual databases. But due to their business interest they do not want to disclose actual database to other banks. A privacy-preserving SMC solution called privacy-preserving data mining is useful in this scenario.
- ❖ Query Over a Group of Databases: When many organizations want to cooperatively work over a group of their databases, traditionally each must know the database of the

ISBN: 978-81-964047-2-7

other. SMC techniques allow executing queries over the group of these databases without disclosing the individual database to each other. For example police organizations of different Indian states want to find certain information over their collective data records; they can use SMC techniques to explore criminal activities, terrorist activities and many other valuable patterns without disclosing their actual database to other police organization. The same technique can be used by intelligence organizations of different countries.

- Joint Audit: Two or more organizations working as partners can allow audit over their joint accounts without disclosing the individual account details to one another
- Privacy-Preserving Social Network Analysis: The police or the intelligence organizations perform social network analysis to find the relation between their social behavior and the criminal activity but the laws prevent them to do so because of the privacy concern of the citizens. SMC solutions can be used here by the government allowing to perform analysis over its database without the organization knowing exact details of the individual records.
- Privacy-Preserving Auction Management: Auction can be managed without disclosing the individual details to the participating parties.
- Privacy-Preserving Electronic Voting: Online voting can be performed using the technique of SMC which allows individual voting pattern to be a secret.
- Privacy-Preserving Mobile Phone Services: The mobile phone service can be implemented such the location information of the subscriber cannot be known to the service provider or to any other party.
- Privacy-Preserving Medical Diagnostic System: It allows the patients to know their disease without disclosing their identity to others or even the result of the diagnostic.
- Privacy-Preserving Monitoring in Wireless Sensor Networks: The data sensed by the nodes of a wireless sensor network may be made confidential by using SMC techniques because the sensor network is also deployed at sensitive places like military battlefield where there in a need to make the retrieved data secure while allowing computation over the data by the set of nodes.
- Privacy-Preserving Signature and Face Detection: Sometimes it becomes necessary to keep the signature or the face of a person secret while matching it with the stored database. In this situation SMC techniques are useful. Our Equality check protocols can be used here.

ISBN: 978-81-964047-2-7

SECURE SUM PROTOCOLS FOR INSECURE NETWORKS

Secure sum protocol permits calculating the sum of sensitive data multiple cooperating but distrustful parties without revealing their data to one another. Many Protocols are proposed by the researchers but there may be a presumption of secure network lines while devising the protocol. In this chapter secure sum algorithms applicable to insecure networks have been devised. We dropped the assumption of the secure network. To preserve the privacy of the data we used cryptographic techniques so that the actual value of the data is not visible to the user. The data provided by one party to another may be encrypted using symmetric or asymmetric encryption methods. The secure sum algorithms employing a physical ring structure faces a threat of colluding neighbours to capture the data of a middle party in the ring. To overcome this threat we devised a protocol which prevents colluding neighbours hacking the data of a middle victim. We consider developing algorithms for secure sum because all other arithmetic operations can be implemented using addition operation. Many practical situations arise when privacy of data becomes a concern. On the other hand knowing the result of common computation is in the mutual interest of the joint parties. Consider following scenario:

1. Four brothers living independently want to know the total wealth of family but no brother wants to disclose his individual wealth to others.
2. All the students in a class want to know the average marks obtained by students in a test but no student wish to show his marks to other students.
3. Group of a mobile service provider companies wants to know the total number of customers of all the companies in an area but no company wants to disclose its number to other companies in the group.

A protocol was proposed by Cliften et al. in 2002 [4] for computing the secure sum where the parties are set in a ring structure. One of the parties begins the protocol by adding random number to its data. The sum of all the private data plus random number reaches the originator party. The party subtract the random number and sends sum to all the parties. In this secure sum computation scheme the data flow in unencrypted form not preserving the privacy. Another threat is that to parties to a middle victim can share their data to capture middle party's data.

In this chapter we propose protocols which are also suitable for insecure networks. We used symmetric and asymmetric encryption on the data flowing on the network lines. It gives confidentiality to the sensitive data. There are three protocols proposed by us in this chapter.

A secure sum with symmetric key protocol uses a shared secret key by all the parties jointly performing the computation. An initiator party sends its data plus random number encrypted with the shares secret key to the next party in the structure. The receiving party decrypts the sum with the same key to get the data of sending party plus the random number. Now, it adds its data, encrypts with the key and sends encrypted sum to the next party. The next party does the same steps; receive, decrypt, add, encrypt and send. At the end the originator receives the sum of all the data plus random number encrypted with the key. The initiator party decrypts and subtracts the random number from the sum to get the actual sum of the data. In the second protocol secure sum with asymmetric key each party generates its public- private key pair with certain algorithm like RSA. Each succeeding party shares its public key with the previous party. The initiator party adds a random number to its data and encrypts with the public key of the next party to which it sends the encrypted sum. The receiving party decrypts with its private key to get the partial sum. The steps; receive, decrypt, add own data, encrypt with the public key of the next party, and send to the next party continues till the originator is reached. The originator gets sum plus random number at the end and therefore it subtracts the random number to get the actual sum. But the threat of the middle party being victim by its two colluding neighbours prevails in this method. The conspiring neighbours may share the partial sum and take difference of the partial sum to get the actual data of the middle party. To deal with such a threat we propose another method Secure sum against colluding neighbours the protocol initiator party generates publicprivate key pair and distributes its public key to all the joint parties. The initiator encrypts its data with its public key and sends to the next party. No random numbers are chosen. The next party simply adds its data to the received encrypted number, encrypts with the public key of the initiator and sends to the next party. This process is repeated until the protocol initiator is reached. At the initiator multiple decryption operation is to be performed to get the actual sum of the individual data. Since the private key is with the initiator only, no two parties can collude to get the data of the middle party

Secure sum with symmetric key

This method uses a shared secret key which is with all the parties. It also has a unanimously elected initiator party similar to Clifton et al. [4]. The encryption of the sensitive data with symmetric key provides privacy. Thus, the protocol is suitable for insecure networks.

Informal description of secure sum with symmetric key

The proposed protocol uses shared secret key which is assumed to be acquired by all the cooperating parties in advance. One of the parties is to initiate the protocol and it is assumed that there is a consent among parties that who initiates the protocol. The protocol initiator party chooses a random number (which is its secret), and adds to the private data with it. Then it encrypts the partial sum so obtained and sends to the adjacent party in the topology. The party receiving this partial sum decrypts it with the shared key to recover the partial sum sent by the initiator. Now, it adds its sensitive data to compute a new partial sum, encrypts it with the shared key, and sends to the next party. The next party will do the same steps. This process is repeated until the protocol initiator is reached. At this stage the initiator will recover the sum of all the private data and the random number. By subtracting the random number actual sum of all the data is obtained. This sum is broadcasted to all the parties. The encryption by the parties maintains the privacy of partial sum at the network lines. The random number maintains privacy of the data among parties.

Secure sum with asymmetric key

In the secure sum with symmetric key the security of the shared secret key is a great concern. When the key is compromised at any point in the network, the whole objective of the protocol becomes unachievable. The method in this section uses public key cryptography where each party generates a pair of public and private keys. Each of the parties shares its public key with the immediate preceding party in the ring topology. The parties send the private data by encrypting with the public key of the receiver. The receiver can decrypt the received encrypted data with its private key to recover the partial sum.

Informal Description of Secure sum with asymmetric key

All the parties wishing for joint secure sum computation has to generate publicprivate key pair. The succeeding party shares its public key to the previous party. There is consent among the cooperating parties that some party will act as a protocol initiator. The initiator chooses a random number, and adds to its private data. It now encrypts this partial sum with the public key of the next party, and sends the encrypted partial sum to the next party in the topology as shown in the Fig.3.5. The receiving party recovers the partial sum by decrypting with its private key. Now, it adds its private data to the recovered partial sum, and encrypts with the public key of the next party in the ring. It sends this encrypted partial to the next party in the ring. Now the receiving party performs the same steps, and sends the public-key encrypted partial sum to the next party. This process continues until the initiator party is reached. The

ISBN: 978-81-964047-2-7

initiator party recovers the sum of all the data plus random number. It removes the random number and then broadcasts the result to all the parties.

Secure sum against colluding neighbours

The protocols described in this chapter so for face the problem of colluding neighbours who can share their partial sum to capture the private data of a middle party. Another limitation of the previous protocols is that all the parties perform encryption and decryption of the partial sum. To reduce this computation overhead and solve the problem of colluding neighbours we propose a new protocol called secure sum against colluding neighbours in which the parties except the initiator only perform encryption and not decryption. All the parties use the public key of the initiator party for encryption. The initiator decrypts multiple times to recover the secure sum.

The information security problem where multiple cooperating but distrustful parties wish to evaluate some function of their private data such that during joint computation their data is not revealed to other cooperating parties is called Secure Multiparty Computation (SMC). Secure sum computation is an instance of SMC where multiple parties compute sum of the sensitive data of all such that the result is correctly received and the privacy of the individual data is preserved. Many algorithms for secure sum computation is available in the literature. Many of them are applicable to secure network lines only. In this chapter we devised three novel secure sum protocols which can be used on insecure network lines. To maintain the privacy on the network lines we used symmetric and asymmetric encryption techniques. In first protocol secure sum with symmetric key we used symmetric encryption technique where the data is encrypted with a common secret key. But the problem of key distribution is critical here. When the key is compromised the whole network data will be hacked. In another protocol secure sum with asymmetric key all the parties generate their public – private key pair. The party sending data to other party encrypt the secret data with the public key of the recipient. Now the recipient decrypts it to recover the partial sum. In both the above techniques a party called initiator uses a random number that is added to the initiator data. The initiator recovers the sum and distributes to all. The third method uses single key pair of the initiator.

SECURE SUM PROTOCOLS USING HOMOMORPHIC AND THRESHOLD ENCRYPTION

Computing sum securely has been discussed in chapter 3. In secure sum multiple parties working on a joint task compute the sum of their individual data such that the result is known

to all the cooperating parties and the individual data privacy is preserved. The secure sum computation can work as a component of privacypreserving data mining tool kit because addition is a basic operation in computer system. All other operations can be implemented using add operation. The protocols assuming secure networks are impractical as almost all network lines are insecure. In chapter 3 of this thesis symmetric and asymmetric encryption is used to protect privacy of the data flowing on the network lines. These protocols need huge number of encryptions and decryptions. In this chapter we use additive homomorphic properties of a cryptosystem to optimize number of encryptions and decryptions. The property ensures that the sum of the encrypted values is same as the encrypted sum of data. First protocol of this chapter named as Secure homomorphic sum uses additive homomorphic property of a cryptosystem as devised by Paillier in 1999 [5]. In another protocol for secure sum named as Secure threshold sum we apply Shamir's threshold scheme together with the additive homomorphic encryption. The threshold scheme shows that a piece of information can be reconstructed from minimum k number of pieces out of total n pieces of information. Less than k pieces are not sufficient to reconstruct the original information. In both the cases our protocol is suitable for semi-honest adversary who follow the steps honestly as listed in the protocol but may perform some extra steps to learn private values with parties. We use ideal model of SMC in Secure homomorphic sum where there is no third party involved during computation. We coin the term semi-ideal SMC model in case of Secure threshold sum as it is the hybrid of the ideal and real model of SMC.

Informal Description of Secure asymmetric sum protocol

The protocol secure asymmetric sum works in similar way as the protocol secure random sum because both use similar architecture of ideal SMC model. The difference is that the secure random sum use random numbers for privacypreservation and the protocol secure asymmetric sum uses cryptographic approach using additive homomorphic cryptosystem. A protocol initiator party, normally P0 generate public private key pair (PU, PR) and distribute to all the participating parties. It is assumed in our protocol that this distribution is achieved by some secure means in advance. All the parties encrypt their private data with the private key and send the ciphertext to the TP. The TP takes sum of all such ciphertexts and sends back the computed sum to the participating parties. Owing to the additive homomorphic properties the sum of ciphertext is equal to the ciphertext of the sum of the private data. The participating parties decrypt the sum to get the sum of the private data. The private key is with the participating parties only. Thus the TP doesn't know anything about the actual data.

The parties do not communicate any data with one another. Therefore the privacy of the data is preserved from one another. In this chapter we devise two protocols to get the sum of the private data of certain number of joint but distrustful parties. The problem is called secure sum problem in the literature. It is an instance of more general problem of information security called Secure Multiparty Computation or SMC. In SMC multiple parties compute some function of their private data without disclosing the individual data to one another. The first protocol in this chapter Secure Homomorphic sum allows cooperating parties to compute sum of their private data using additive homomorphic properties. It uses a ring topology in which parties are supposed to be arranged. One of the parties is unanimously chosen as a protocol initiator. The party generates its publicprivate key pair, distribute the public key to all, and helps in computing the sum of encrypted values. This sum is equivalent to the ciphertext of the sum of the data. By decrypting with the private key, the initiator gets the secure sum. In the second protocol, Secure threshold sum of this chapter additive homomorphic property along with the threshold encryption scheme is used to get the secure sum. A novel SMC model which removes the dependence on the trustworthiness of the TTP is proposed. The TTP generates public-private key pair using additive homomorphic property and shares of the private key using threshold encryption scheme. The protocol is improved in many ways. It is applicable to insecure networks and semihonest adversary. Another secure sum protocol, secure random sum in ideal model is proposed where a TP helps the participating parties computing the sum of private data. A common random number is used to protect the privacy of the data from other parties as well as from the TP. In the last protocol of this chapter, secure asymmetric sum protocol, the architecture of secure random sum is use. But instead of using the random number, a publicprivate key pair using homomorphic cryptosystem is used for privacy preservation. The sensitive data is encrypted by the public key and then sent to the TP the TP computes the sum of these ciphertexts and sends sum to the parties. The parties recover the sum by decrypting it with the private key. The parties do not communicate directly. Therefore the privacy of the data is preserved among them. The TP receives data encrypted with the public key. As TP doesn't possess the private key, it cannot recover the data. The protocol is suitable for semi-honest adversary. Future work can be extended to devise the secure sum protocols for malicious adversaries. Zero Knowledge proof concepts will be helpful in this case. Our protocols are privacy preserving protocols. Protocols can be devised which protect other security properties like integrity and nonrepudiation.

SECURE MULTIPARTY EQUALITY CHECK PROTOCOLS

Much has been discussed in the previous chapters about the information security problem SMC. Many times the parties simply desire to check their individual data for equality but they are worried about the privacy of their data. During privacypreserving data mining operation the sensitive data may be present at multiple sites who are involved in computation. Parties need results but they are afraid of the privacy of their data. In this chapter we have proposed a protocol which allows multiple joint parties to check their data for equality without revealing individual data to one another. We have used randomisation technique as well as the secure additive homomorphic property of the encryption to get the correct result. The protocol is suitable for semi-honest adversary.

Secure MultiEqualityCheck protocol

In our proposed protocol we have extended the two-party EqualityCheck protocol to protocol which can perform equality check of the data of more than two parties. We have proposed architecture for multiparty equality check. We also provide informal and formal description of the proposed MultiEqualityCheck algorithm. We also analyse it for its performance.

Informal description of Secure MultiEqualityCheck protocol

Referring to the Fig.5.3, multiple parties P0 to Pk-1 run MultiEqualityCheck by choosing unanimously one party as initiator. Normally, P0 is chosen as an initiator. All the parties generate their public-private key pair using additive homomorphic cryptosystem. Initially, the party P0 and P1 run EqualityCheck algorithm to check equality of their data. If the equality holds, then the party P1 and P2 run EqualityCheck algorithm. In this way equality is checked pair-by-pair. If equality continues to hold till the initiator is reached, the result that all the data are equal is declared. If equally breaks at any point in the ring the result that the data are not equal is declared.

The privacy preserving function evaluation is the need of today's world as the sensitive data from multiple sources may be shared by multiple sites for joint computation. Secure equality check is one of the cases where a set of parties need to compute the data for equality. For example two police stations situated far apart want to check some biometric data of a criminal such that both the stations do not want to disclose actual data to one another. In this case two-party equality check would be useful. In our work we have extended the two-party equality check to multiparty case. Our protocol secure MultiEqualityCheck uses additive homomorphic cryptosystem where the sum of ciphertext is equal to the ciphertext of the sum. Our work is suitable for semi-honest adversaries. The protocol MultiEqualityCheck uses

ISBN: 978-81-964047-2-7

public-private key generation using homomorphic encryption, encryption, and decryption. We proposed a novel equality check scheme using hash function. The protocol EqualityHashCheck ensures checking equality among multiple parties with the help of comparison of hash functions. It has an advantage of less computation as compared to the MultiEqualityCheck. We have proposed EqualityHashCheck for both real and ideal model of SMC. In real model the parties compare their hashes pair-by-pair until all the parties are traversed. In the ideal model the hashes are provided to a TP. It is the TP which checks whether all the hashes are equal and provides result to all the parties. The privacy is preserved from TP as it receives hash of data not the actual data. The hash function is an irreversible function. That mean given the hash, it is infeasible to compute data from the hash. Similarity, privacy is preserved among parties as they do not communicate with each other. But, the cost of maintaining the TP is an overhead.

LITERATURE SURVEY

when multiple parties with their private data want to compute some common function of their data jointly such that the privacy of their data is preserved from one another, the problem is called Secure Multiparty Computation (SMC) [11]. In this scenario all the participating parties do not have trust on one another. But the common function evaluation is in their common interest. Mainly, there are two goals of SMC problem, privacy and correctness of the result. Owing to the heavy use of the Internet which is associated with the distributed system, the scenario of SMC has become highly relevant. The parties geographically distributed on distributed sites may wish to evaluate a function providing their data. But at the same time they are worried about privacy of their data. During online transaction many sites cooperatively work to process the transactions. But each site does not necessarily want to show the exact value of the data it shares with other party. Therefore in this age of large number of online transactions the SMC is has become highly relevant. Parties frequently need to perform joint computation on their sensitive data while keeping confidentiality of the data. Consider a situation where two or banks wish to know cooperatively the details of a customer from their individual databases. Knowing the details about the customer is in mutual interest of all the participating banks. But no want wants to share their database to other banks as they are competitor in the same sector.

The development of networking, the Internet, and distributed system has provided huge opportunities for joint computations where multiple parties perform SMC on their secret data. Due to the concern of the privacy and worldwide regulations made by made by different

countries, there exist many scenarios where SMC becomes applicable. Consider following scenarios which will let you understand the practical applicability of the SMC solutions:

1. A person having his DNA pattern wants to about the genetic diseases associated with his DNA pattern. He wants to do some query from a server storing different DNA patters and the diseases associated with those DNA patterns. But at the same time the person does not want to disclose his exact DNA patter to the server. This privacy-preserving database query can be implemented using SMC solutions.
2. A group of mobile service providers wish to compute together to prepare list of total subscribers active in an area within some specified time interval. This could be due to some police investigation and asked by the intelligence of a country. The companies do not want to disclose their customer details. This is a case of privacy-preserving union of sensitive databases and can be solved using SMC solutions.
3. A group of students wish to know their average marks obtained in an examination but know student wish to disclose his actual marks to other students of the group. This is a case of privacy preserving sum computation which is called as secure sum in the literature. The sum in this example is divided by number of students to get the secure average.
4. Five real brothers who live separately wish to compute their total wealth but know brother wish reveal his wealth to other brothers. This problem can also be solved using secure sum solutions.
5. A bank wants to some loan details of a suspicious customer from other bank but the bank don't want to disclose actual customer details to other bank. This privacy-preserving query can be solved using SMC solutions.
6. A group of police stations in a country wish to search details of a criminal from their databases but no police station wish to show its database to other police station. This is a case of privacy preserving data mining.
7. A client computer in a payment system wish to learn that a QR code is matching or not without showing the actual QR code to the server. This is a case of privacy-preserving matching or equality check.

Formally, consider multiple parties P0 to Pk-1 with private data d0 to dk-1 respectively. These k parties want to evaluate function f (d0, ..., dk-1) without revealing their private data to each other. Based on the type of f many specific SMC problems are devised by researchers and thereafter many real life applications emerged. We pointed out these works in our publications . This chapter will explore SMC and will put forward the research gap.

ISBN: 978-81-964047-2-7

SMC Models

The SMC problem needs paradigms for performance analysis and mathematical modelling. The participating parties with private data are the essential stake of the SMC paradigm. But sometimes these parties seek assistance of a Trusted Third Party (TTP) for function evaluation. Based on the presence and absence of the TTP, two SMC paradigms evolved in the literature:

1. Ideal SMC Model

2. Real SMC Model

Ideal SMC Model

In the Ideal SMC Model the cooperating parties take the services of a TTP for evaluating the function of their data. In ideal case the TTP should be fully honest as it does not share any sensitive information of one party with any other party. But researchers in the literature analysed every kind of behaviour of the TTP. Due to presence of the TTP it becomes easier for the participating parties to preserve privacy of their data from one another as they will share least information with one another. If the TTP behaves maliciously the whole objective of the SMC becomes unachievable. The architecture of the Ideal SMC model is depicted in the Fig. 2.1. The Ideal Model is easier to implement. But there are two drawbacks of this model. One, the cost of maintaining the TTP, and second its trustworthiness. But it is the model is more popular is practice as the TTP is a government agency or an organisation approved by the government. In case of any dispute among the participating parties, the TTP plays an important role to reach at the decision.

SMC SOLUTION TECHNIQUES

When the researchers paid attention to the requirement of privacy-preservation during computation, they started providing solutions using combinational logic circuits. The inputs to these circuits were the sensitive data and the results were obtained at the outputs. As an example, consider a case where two parties wish to check the comparative magnitudes of their data without sharing actual magnitudes with one another. A combinational logic circuit named digital comparator can be useful. It accepts two binary numbers at the imput, and provides the result at the output. The comparator can be treated as the TTP of the Ideal SMC Model. The scheme is shown in the Fig.2.3 [14, 15]. The result is obtained over three output lines; E, G, and L. If E = 1, both the numbers are equal. If G = 1, first number is greater than the second number. If L = 1, the first number is less than the second number. Assuming the input lines to be secure, the result is obtained at the output preserving the privacy of two

ISBN: 978-81-964047-2-7

parties' data from one another. But, for even two-party case the circuit is complex and expensive. For multiparty the complexity and the cost reaches so high as not practically affordable. Soon the researchers realised to use some other economical method which can be practically used. In this connection the work of Yao in 1982 [1] used cryptographic technique to provide solution to millionaires problem. The problem allows two millionaires to know who is richer without sharing actual wealth to one another in Real SMC Model. Yao used symmetric encryption techniques to achieve the result. Yao's two-party SMC problem was extended to multiparty SMC problem by some researchers. For example Goldreich et al. [16] proposed an algorithm for multiple parties to compute some function and leaking incomplete information. But this is suitable when majority of parties were honest. Generalised solutions for multiple SMC problems are proposed by the researchers but these solutions are inefficient. Goldreich et al. [18] showed that specific solutions for a particular SMC problem are more efficient than these general solutions. Presently, there are three techniques which are used to provide solutions to SMC problems:

1. Randomization techniques
2. Cryptographic techniques
3. Anonymization techniques

Randomization SMC techniques

In randomization approach participating parties use random numbers to preserve privacy during computation. The protocols are designed such that the presence of the random number does not influence the result of the computation. But the random number is used to hide the actual value of the data from other parties involved in the computation. Many of the programming languages available today have in-built random function that can be used in these protocols. To understand the randomization technique we explain the secure sum algorithm presented by Cliften et al. [4] where they propose an algorithm for computing sum of individual data from multiple parties such that their privacy is preserved. This is the most easily understood SMC solution using random number. They proposed the protocol by arranging the cooperating parties in a logical ring structure. This is a real SMC model as there is no TTP. All the party unanimously agree on a party which initiates the protocol. The protocol initiator party first chooses a secret random number. It adds the random number to its private data, and sends the partial sum to the next party in the ring. The network lines are assumed to be secure, and therefore it is assume that no intruder can intercept the network lines to capture the data. There is always a unidirectional communication in the logical ring.

The party receiving the partial sum adds its private data to the partial sum and gets new partial sum. This partial sum is sent to the next party in the logical ring structure. This process is repeated until the protocol initiator party is reached. At this point the protocol initiator party gets sum of all the data plus random number. The initiator subtracts random number to get the sum of actual data of all participating parties. The sum is then broadcasted to all the participating parties.

Cryptographic SMC techniques

The cryptographic approach for providing solutions to SMC problems uses symmetric and asymmetric encryption techniques. In symmetric encryption method a common key is used for encryption as well as decryption. But an asymmetric encryption method uses different keys for encryption and decryption. The second method is also known as public key cryptography. Different researchers used their approach and the literature of SMC has many building blocks which can be used to solve SMC problems [19]. These building blocks are listed below:

1. Yao's Millionaires Problem.
2. Homomorphic Cryptosystem.
3. Oblivious Transfer.
4. Private Matching.

MOTIVATION OF THIS WORK

Although the secure sum protocol proposed by Cliften et al. [4] is a milestone in secure arithmetic computation. It has many limitations when we apply it to for the real life applications. One, as mentioned earlier is the threat of two colluding neighbours getting a middle party hacked. This drawback can be removed by many ways. The colluding neighbours simply take difference of what they send and what they receive. There is an immense need to provide protocols which deal with this kind of threat. This is because the secure sum is an integral part of privacypreserving distributed data mining. In our earlier work, we proposed segmentation approach for secure sum computation where the sensitive data of the participating parties is broken into segments. The parties then get the sum by adding segments in different rounds. It also eliminates random number and the threat of getting the middle party victim. The protocol we named as k-secure sum protocol [56]. It has been analysed that the protocol provided better security against data leak. In another version of a protocol based on segmentation approach we suggested distributed the segments among parties before the sum computation. The protocol was called as distributed k-secure sum

protocol [57]. In another protocol we proposed an architecture where each party must change its position after one each round of segment computation, so the two neighbours could only hack segments not full data. The protocol was named as changing neighbours k-secure sum protocol [58]. In thesis we used cryptographic approach with random numbers to deal with the problem of colluding neighbours. Another problem with the secure sum protocol was that it assumed the network to be secure. We dropped this assumption and devised protocols for insecure networks. Our protocols are based on symmetric cryptography, asymmetric cryptography, homomorphic cryptosystem, and threshold encryption. We also devised protocols for equality check among multiple parties.

SMC is an interesting paradigm of computing where multiple parties cooperative evaluate some function of their common interest without revealing actual data to one another. The milestone for this problem is a millionaires' problem proposed by Yao in 1982 [1] where two millionaires can cooperatively work to determine who is richer without disclosing exact wealth to one another. It has been predicted that due to privacy concerns soon SMC may become an integral part of our computing environment. The solutions to SMC problems earlier based on the circuit evaluation. Modern SMC solutions use randomization method, cryptographic methods, and anonymization techniques. Many specific SMC problems and their solutions are suggested by the researchers. Based on these solutions many real life applications are developed. The primary goal of the SMC is to get the correct result while preserving the privacy of the data involved in the computation. Researchers proposed SMC problems and solutions with an objective to reduce communication and computation complexities. In this chapter we explored the subject of SMC and various techniques used by researchers, and specific as well as real life applications provided by them. We also pointed out our earlier research contribution to the subject. We are able to find the research gap between earlier work and the work needed. Earlier protocols assumed the network lines to be secure. Therefore we have developed protocols for insecure networks. Our secure sum protocols work appropriate on insecure lines. Similarly, we have extended two-party equality check protocols to multiparty equality check protocols. These are also suitable for insecure networks and semihonest adversaries.

CONCLUSION AND FUTURE SCOPE

A Voyage to Secure Multiparty Computation

Secure Multiparty Computation is a problem of information security where multiple joint parties cooperatively compute some function of their private data without revealing the actual

ISBN: 978-81-964047-2-7

data to each other. In modern distributed computing environment the parties may be geographically distributed at different sites. These sites will have to provide data for computation purpose. But all these sites are concerned about the privacy of their data. SMC solutions provide this privacy. The millionaires' problem proposed by Yao [1] is considered to be the milestone to modern SMC era. The problem provides solution as how two millionaires can determine together who is richer without disclosing actual wealth to one another. He provided cryptographic solution to the problem. This two-party problem was extended to multiparty case by many researchers. As of today, there are three techniques which are used to provide SMC solutions; Randomization, cryptographic, and anonymization methods. The randomization technique uses random numbers to hide the actual data during computation. The random number is removed while declaring the result. Secure sum protocol of Cliften et al. [4] is an easier example of randomization approach. In cryptographic techniques, the data is encrypted during computation so that its actual value is not revealed to other parties. Decryption operation is performed while declaring the result. This technique may exploit homomorphic property of the cryptosyatem which ensures that sum of the ciphertext is equal to the ciphertext of sum of data. Another important component of the cryptographic technique could be an oblivious transfer where a sender has many inputs and a receiver needs one of them. The receiver sends the index for the input selection and the corresponding input is received by the receiver. But the sender is not aware about the index value and the receiver is not aware about the exact position of the received input. In anonymization method the identity of the participating parties is hidden by anonymizers. Therefore the privacy of data will be maintained during the computation. The architecture of the SMC solution has two flavours in the literature; Ideal SMC model and Real SMC model. In an Ideal SMC model there exists a third party among the participating parties which assists the secure function evaluation. This model is easier to implement. But the trustworthiness of the third party is to be taken into consideration while devising the protocols. In Real SMC model there is no third party and the participating parties only run the protocol to compute the function securely. This model is difficult to implement but is less expensive due to absence of third party. In this thesis we proposed a third model which is a hybrid of above two models.

Secure Sum Protocols and their performance

Secure sum computation is an instance of the generalised SMC problem where the cooperating parties compute sum keeping their privacy. The work of Cliften et al. [4] is considered to be milestone where they used a random number to compute the sum of private

ISBN: 978-81-964047-2-7

data of all the parties securely. We also focussed on secure sum as all other arithmetic operations can be implemented using addition operation. There are two limitations of secure sum protocol; first it assumes network lines to be secure and second the problem of colluding neighbours in the architecture. But in actual practice the network lines are always insecure. Therefore in this thesis we devised secure sum protocols for insecure networks. They used real SMC model. We proposed secure sum protocols for both the SMC models; real as well as ideal SMC model. We also devised protocol for our novel proposed semi-ideal model. The secure sum protocols proposed in this thesis use symmetric and asymmetric cryptography for making them secure on thee insecure network lines. Our secure sum with symmetric key protocol [7] uses a common shared secret key by all the participating parties. The parties also use random numbers similar to Cliften's secure sum protocol for preserving privacy among them. The parties first encrypt the partial sum before sending it to the next party. Therefore the confidentiality is maintained on the insecure network lines. Parties decrypt the received encrypted partial sum and add their private data, encrypt again, and send to next party. Although, the privacy is achieved among parties themselves and on the insecure network lines but the computation involved in the process of decrypt-add-encrypt is very large. Another problem with this is sharing the symmetric key over insecure network limes is also an issue. When the shared key is compromised the whole scheme fails. In another protocol proposed by us in thesis named secure sum with asymmetric key protocol [7] we suggest an architecture where each of the parties generates its public private key pair. Each party shares its public key with the previous party in the ring. The party adds and encrypts the partial sum with the public key of the recipient. The recipient decrypts the encrypted sum with its secret private key. Rest process is similar with secure sum with symmetric key protocol. This method solves problem of key distribution among parties. Both of these protocols suffer from the problem of colluding neighbours. To remove this problem we proposed another protocol named as Secure sum against colluding neighbours protocol[7]. It uses multiple encryptions and decryptions for privacy. The protocols discussed so far employ huge computations for encryption and decryption. In order to reduce amount of computation we used additive homomorphic property of a cryptosystem [5]. This property ensures that the sum of encrypted data is equal to encrypted value of the sum. Therefore irrespective of the number of participants the decryption needs to be done only once. Based on this idea we proposed Secure homomorphic sum protocol [8]. We suggest a ring architecture in which the parties communicate in one direction. An initiator party generates its public private key pair using

ISBN: 978-81-964047-2-7

additive homomorphic cryptosystem [5]. The party shares its public key with all the parties. All parties encrypt the private data with the public key and send the partial sum to the next party. This process is repeated until the initiator party is reached. The initiator decrypts with its secret private key to recover the secure sum which is distributed to all the parties. Although the amount of computation is much reduced due to the single decryption operation, the protocol faces non-uniform privilege of initiator party generating, distributing public key, and holding private key a secret. In our other protocol services of a third party are used 87 for generating public-private key pair, distributing priva key to all the parties. The private key is not hold by any single party. Instead we used Shamir's secret sharing concept of threshold encryption [6] where it has been showed that a value can be reconstructed out of certain minimum number of shares of that value. Therefore we suggest the third party generate number of shares of the private key and then distribute one share to each of the parties. The parties can recover the private key only if all provide their share honestly. The protocol is names as Secure threshold sum protocol. The secure sum protocols suggested in [4, 56, 57, 58] and many other researchers are applicable to real SMC model. As pointed out real models are difficult to implement, we have proposed protocols for ideal model where the services of a third party are used. The protocol is named as Secure random sum and is described in chapter 4. The participating parties agree on a common random number which is multiplied to the private data. The products so computed are sent to the third party. The third party computes sum of all such products and sends the sum to all the parties. These parties divide the sum by their secret random number to get the secure sum. The protocol is simple with minimum computation complexity. But sharing of the random number on insecure network lines is an issue. In order to solve the problem of random number distribution we have proposed another secure sum protocol for ideal model named as secure asymmetric sum protocol in ideal model. It is a slightly modified version of Secure random sum protocol. Instead of using random numbers, all the parties generate public private key pair. The data is encrypted with the public key and sent to the third party.

Secure Equality Check Protocols and their performance

The problem of checking equality between two parties has been proposed in [59] which used homomorphic encryption to check equality of two parties in real model. We have extended the problem to multiparty equality check. Our protocol MultiEqualityCheck [10] allows multiple parties to check equality of their data without disclosing their private data. It is based on additive homomorphic public key cryptosystem. It uses real SMC model and suitable for

semi-honest adversaries. It is also applicable to insecure networks. The MultiEqualityCheck protocol use key generation, encryption, and decryption at each party. We also proposed protocol for equality check which simply uses hash function. Each of the parties takes hash of their data and compare these hashes cooperatively. Based on these we developed novel protocol EqualityHashCheck for both real and ideal SMC model. There are obvious benefits of the EqualityHashCheck over MultiEqualityCheck protocol. It does not need any key and random number. Its amount of computation needed is very less. Communication complexity is also very low.

Future Scope

we have proposed novel secure sum protocols and secure equality check protocols. The protocols we propose in this thesis are suitable semi-honest adversaries. These adversaries should obey all the steps described in the protocol but they may also try to learn some private data of other parties. The work can be extended for devising the protocols for malicious adversaries. These adversaries neither follow the steps of the protocol nor remain inactive when it comes to hacking the data of cooperating parties. The protocols when designed for the malicious or corrupt parties must be able to check the inputs for correctness. This is a big deal. But techniques are available in the literature for designing such protocols. There are zero knowledge proof techniques which help no whether an adversary has knowledge of the data and whether the data is correct. These protocols are complex and expensive. But they are robust and need of real life applications. Thus, there is a trade off between its efficiency and cost which needs to be considered while devising such protocols. In this thesis we have proposed some protocols for real model, some for ideal model, and some protocols for both the models. Suitable models can be used to our protocols based on the need. There are many aspects of the network security like privacy, authentication, integrity, and nonrepudiation. Our protocols are designed to deal with the first one, privacy. All these protocols can be designed to take care of other aspects of the network security. Finally, we deigned protocols for scalar values. Protocols can be designed which are applicable for matrix data.

REFERENCES

[1] A. C. Yao, “Protocol for Secure Computations,” in the proceedings of the 23rd annual IEEE symposium on foundation of computer science, pages 160-164, Nov.1982.

[7] R. Sheikh and D. K. Mishra, “Secure Sum Computation for Insecure Networks,” in the proceedings of international conference ICTCS 2016 Udaipur, India.

[8] Sheikh R., Mishra D.K. (2019) Secure Sum Computation Using Homomorphic Encryption. In: Mishra D., Yang XS., Unal A. (eds) Data Science and Big Data Analytics. Lecture Notes on Data Engineering and Communications Technologies, vol 16. Springer, Singapore.

[9] R. Sheikh and D. K. Mishra, "Secure Sum Computation using Threshold encryption for Semi-Ideal Model," in International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-5, January 2020.

[10] R. Sheikh and D. K. Mishra, "Secure Multiparty Equality Check Based on Homomorphic Cryptosystem," in International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-5, March 2020.

[11] W. Du and M.J. Atallah, "Secure Multiparty Computation Problems and Their Applications: A Review and Open Problems," In proceedings of new security paradigms workshop, Cloudcroft, New Maxico, USA, page 11-20, Sep. 2001.

[12] R. Sheikh and D. K. Mishra, "Secure Multi-Party Computation: A Research Proposal," in the proceedings of the International Conference on Computer and Communication, ICCC2012, pages 876-881, Bhopal, India, Jan 2012.

[13] R. Sheikh, B. Kumar and D. K. Mishra, "Secure Multi-party Computation: From Millionaires Problem to Anonymizer," in the Information Security Journal: A Global Perspective, Taylor and Francis, Vol. 20, Issue 1, pages 25-33, USA, 2011.

[14] R. Sheikh, M. Vyas, B. Kumar and D. K. Mishra, "A Simple Hardware Implementation of Yao's Millionaires Problem," accepted for publication in Third CSI National Conference on Education and Research ConfER 2010, Guna, India, Mar. 2010.

[15] R. Sheikh, "Digital Computers: Electronics, Organization and Fundamentals," 9th edition, Nakoda Publishers and Printers, pages 281-283, India, 2009.

[16] O. Goldreich, S. Micali, and A. Wigderson, "How to play any Mental Game," in STOC '87: Proceedings of the nineteenth annual ACM conference on Theory of computing. New York, NY, USA: ACM, pages 218-229 1987.

[17] D. K. Mishra, N. Koria, N. Kapoor and R. Baheti, "A Secure Multiparty Computation Protocol for Malicious Computation Prevention for Preserving Privacy during Data Mining," in the International Journal of Computer Science and Information Security, Vol. 3, No. 1, pages 79-85, Jul. 2009.

[18] O. Goldreich, "Secure Multi-Party Computation (Working Draft)," available from http: //www.wisdom.weizmann.ac.il/ home/oded/ public html/ foc.html 1998.

[19] V. Oleshchuk, and V. Zadorozhny, "Secure Multi-Party Computations and Privacy Preservation: Results and Open Problems," Telektronikk: Telenor's Journal of Technology, Vol. 103, No.2, 2007.

[20] I. Ioannidis and A. Grama, "An efficient Protocol for Yao's Milionaires problem," In proceedings of the 36th Hawaii International Conference on System Sciences, 2003.

[21] L. Shindong, W. Daoshun, D. Yiqi and L. Ping, "Symmetric Cryptography Solution to Yao's Millionaire's Problem and an evaluation of Secure Multiparty Computations," In the International journal of Information Sciences, Vol.178, Issue1, pages 244-255, Jan. 2008.

[22] Amirbekyan and V. Estivill-Castro, "Practical protocol for Yao's millionaires problem enables secure multi-party computation of metrics and efficient privacy-preserving k-NN for large data sets," Knowledge and Information Systems. [Online]. Available: http: //dx.doi.org /10.1007/s10115-009- 0233-z, 2009.

[23] C. Cachin, "Efficient Private Bidding and Auctions with an Oblivious Third Party," in Proceedings of the 6th ACM conference on computer and communications security, pages 120–127, Singapore, Nov. 1999.

[24] J. Benaloh, "Dense Probabilistic Encryption," in proceedings of the workshop on selected areas of Cryptography, Kingston, ON, pages 120-128 May 1994.

[25] D. Naccache and J. Stern, "A New Public Key Cryptosystem Based on Higher Residues," in Proceedings of the 5th ACM Conference on Computer and Communications Security, CCS '98, San Francisco, CA, New York, NY, ACM Press, pages 59- 66. Nov. 1998.

[26] M. O. Rabin, "How to Exchange Secrets by Oblivious Transfer," Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[27] S. Even, O. Goldreich and A. Lempel, "A Randomized Protocol for Signing Contracts," Commutations of ACM, 28 (6), pages 637- 647, 1985.

[28] J. Kilian, "Founding Cryptography on Oblivious Transfer," in Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, STOC '88, Chicago, Illinois, New York, NY, ACM Press, pages, 20-31, May 1988.

[29] M. Freedman, K. Nissim and B. Pinkas, "Efficient Private Matching and Set Intersection," Advances in Cryptology, Eurocrypt'2004 Proceedings, LNCS 3027, Springer-Verlag, pages 1-19, May 2004.

[30] R. Agrawal, A. Evfimievski, and R. Srikant, "Information Sharing Across Private Databases," In Proceedings of the ACM SIGMOD, International Conference on Management of Data, San Diego, CA, 2003.

[31] D. K. Mishra, M. Chandwani, "Extended Protocol for Secure Multiparty Computation using Ambiguous Identity," WSEAS Transaction on Computer Research, Vol. 2, Issue 2, Feb. 2007.

[32] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private Information Retrieval ," in proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science, Milwaukee WI, pages 41-50, Oct. 1995.

[33] Y. Gertner, Y.Ishai, , E. Kushilevitz, and T.Malkin, "Protecting Data Privacy In Information Retrieval Schemes," In proceedings of the Thirtieth Annual ACM Symposium on theory of computing, STOC '98, Dallas, TX, 24-26 New York, NY, ACM Press, pages 151-160, May 1998.

[34] B. Chor and N. Gilbao, "Computationally Private Information Retrieval (Extended Abstract)," in proceedings of 29th annual ACM Symposium on Theory of Computing, El Paso, TX USA, May 1997.

[35] Y. Ishai and E. Kushilevitz, "Improved Upper Bounds on Information-Theoretic Private Information Retrieval (Extended Abstract)," in proceedings of the thirty-first annual ACM symposium on Theory of computing, Atlanta, GA USA, May 1999.

[36] G. Di-Crescenzo, Y. Ishai and R. Ostrovsky, " Universal Service-Providers for Database Private Information Retrieval," in proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing, Sep. 1998.

[37] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single Database, Computationally-Private Information Retrieval," in Proceedings of the 38th annual IEEE Computer Society Conference on Foundation of Computer Science, Miami Beach, Florida USA, pages 20-22 Oct. 1997.

[38] C. Cachin, S. Micali and M. Stadler, "Computationally Private Information Retrieval with Polylogarithmic Communication," Advances in Cryptology: EUROCRYPT '99, Lecture Notes in Computer Science, 1592:402–414, 1999.

[39] Y. Gertner, S. Goldwasser and T. Malkin, "A Random Server Model for Private Information Retrieval," in 2nd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM '98), 1998.

40] Saint-Jean, Felipe, A Java Implementation of a Single-Database Computationally Symmetric Private Information Retrieval (cSPIR) protocol, Yale University Technical Report YALEU/DCS/TR-1333, July 2005.

[41] R. Canetti ,Y. Ishai , R. Kumar, M. K. Reiter, R. Rubinfeld and R. N.Wright, "Selective Private Function Evaluation with Applications to Private Statistics," in Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing, PODC '01, Newport, Rhode Island. New York, NY, ACM Press, pages 293-304, 2001.

[42] W. Du and M. J. Atallah, "Privacy-Preserving Cooperative Scientific Computations," In 14th IEEE Computer Security Foundations Workshop, pages 273–282, Nova Scotia, Canada, Jun. 2001.

[43] Y. Lindell, "secure multiparty computation for privacy preserving data mining," IBM, T.J. Watson Research Center, USA, http://u.cs.biu.ac.il/-lindell/ researchstatements /mpc-ppdm.htm/2001.

[44] Quinlan, J.R. (1979). Discovering rules by induction from large collections of examples. In D. Michie (Ed.), Expert systems in the micro electronic age. Edinburgh University Press.

[45] R. Agrawal and R. Srikant, "Privacy-Preserving Data Mining," in Proceedings of the 2000 ACM SIGMOD on management of data, Dallas, TX USA, pages 439-450, May 2000.

[46] Stanley R. M. Oliveira, Osmar R Zaiane, "Privacy-preserving clustering by data transformation," Computer Science JIDM 2003.

[47] W. Du and M. J. Atallah, "Protocols for Secure Remote Database Access with Approximate Matching. In 7th ACM Conference on Computer and Communications Security (ACMCCS 2000), the First Workshop on Security and Privacy in E-Commerce, Athens, Greece, Nov. 2000.

[48] M. J. Atallah and W. Du, "Secure Multi-Party Computational Geometry," in WADS2001: Seventh International Workshop on Algorithms and Data Structures, pages 165–179, Providence, Rhode Island, USA, Aug. 2001.

49] W. Du and M. J. Atallah, "Privacy-Preserving Statistical Analysis," in Proceedings of the 17th Annual Computer Security Applications Conference, pages 102–110, New Orleans, Louisiana, USA, Dec. 2001.

[50] S. L. Warner, "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias," Journal of the American Statistical Association, 60(309):63–69, Mar. 1965.

[51] S. L. Warner, "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias," Journal of the American Statistical Association, 66(336):884–888, Dec. 1971.

ISBN: 978-81-964047-2-7

[52] M. S. Goodstadt and V.Gruson, "The Randomized Response Technique: A Test on Drug Use," Journal of the American Statistical Association, 70(352):814– 818, Dec. 1975.

[53] K. H. Pollock and Y. Bek, "A Comparison of Three Randomized Response Models for Quantitative Data," Journal of the American Statistical Association, 71(356):994–886, Dec. 1976.

[54] J.Biskup and U. Flegel, "On Pseudonymization of Audit Data for Intrusion Detection," in Workshop on Design Issues in Anonymity and Unobservability, pages 161–180, 2000.

[55] J. Biskup and U. Flegel, "Transaction-Based Pseudonyms in Audit Data for Privacy Respecting Intrusion Detection," in Recent Advances in Intrusion Detection, pages 28–48, 2000.

[56] R. Sheikh, B. Kumar and D. K. Mishra, "Privacy-Preserving k-Secure Sum Protocol," in International Journal of Computer Science and Information Security, Vol. 6 No.2, pages 184-188, USA, Nov. 2009.

[57] R. Sheikh, B. Kumar and D. K. Mishra, "A Distributed k-Secure Sum Protocol for Secure Multi-party Computation," in Journal of Computing, USA, Vol. 2, Issue 3, pages 68-72, Mar. 2010.