Privacy Preserving Clustering on Horizontally Partitioned Data using Zero Knowledge Proof

Ms. Smita Patel[1] Prof. Sanjay Tiwari[2]
[1] M. Tech Student [2]Associate Professor
[1, 2]C. S. E. Department
[1, 2]Arya Institute of Engineering & Technology, Rajasthan Technical University India

Abstract---Zero Knowledge Proof is one of the classical construct that acts as cryptographic primitive in basic multiparty protocols implementing identification schemes and secure computation. A few recent results that convert protocols working in semi-honest models to malicious models using interactive zero knowledge proofs and research on protocol implementing zero knowledge proofs in multiparty environments efficiently have pave the path for similar solutions. This research focuses on providing solution in malicious model by implementing zero knowledge proof system where multiple parties intend to perform privacy preserving clustering on horizontally partitioned database.

Keywords: zero Knowledge Proof, cryptography, secret sharing, privacy preserving data mining, Secure Multiparty Computation

I. INTRODUCTION

In our era Knowledge is not only information anymore, it is an asset. Extract key Data from large database is the data mining. Such databases are frequently distributed among several organizations who would like to Cooperate in order to extract global knowledge. That time prevent the parties for privacy from directly sharing the data among them [1].

The extracted information could be in the form of patterns, clusters or classification models and its results can be applied to various fields including customer relationship management, market basket analysis and bioinformatics.

An approach that was taken toward privacy protection in privacy preserving data mining uses cryptographic techniques, most often the secure computation technique is used [3,4,5,6]. This approach became very popular for two reasons: first, cryptography offers a well-defined model for privacy, which includes methodologies for proving and quantifying it. Second, here exists a vast toolset of cryptographic algorithms and constructs which can be used for implementing privacy-preserving data mining algorithms [1].

In PPDM Even though successful exchange of data, guarantee is needed that whatever shared is correct and actually comes from a trusted source. Accomplishment of such conditions supports the protocol and can make it valuable of use in malicious models. In secret sharing scheme dealer distributes shares to parties such that only authorized subsets of parties can reconstruct the secret. General protocol for multiparty computation, Byzantine agreement, threshold cryptography, access control, attribute-based encryption, oblivious transfer [2].

There are some problems with semi-honest model; no assumption is made on the collusion of parties involved [7] [8].For this reason choose malicious environments as better option for preserving privacy. That is expensive in communication. So in theory, convert semi-honest problems can be change to malicious models using zero knowledge proofs[8].

Zero Knowledge Proof Used SMC Based Privacy Preserving Protocol. In this use Shamir's Secret Sharing Scheme with Cyrel's Identification Scheme with Proposed Privacy Preserving Data Mining.

II. BACKGROUND THEORY

Privacy Preserving Data Mining

A large database be present in society today .Medical data, Consumer purchase data, Census data, Communication and media-related data, Data gathered by government agencies can this data be utilized For medical research, For improving customer service, For homeland security. Sharing of data is necessary for full utilization. Pooling medical data can improve the quality of medical research, Pooling of information from different government agencies can provide a wider picture, What is the health status of citizens that are supported by social welfare?, Are there citizens that receive simultaneous support from different agencies?, Data gathered by the government (e.g., census data) should be publicly available, The huge amount of data available means that it is possible to learn a lot of information about individuals from public data like Purchasing patterns ,Family history, Medical data. Data mining techniques can disclose critical information about business transactions, compromising the free competition in a business setting [9]. Thus, there is a strong need to prevent disclosure not only of confidential personal information, but also of knowledge which is considered sensitive in a given context. For this reason, much research effort has been devoted to addressing the problem of privacy preserving in data mining. As a result, several data mining techniques, incorporating privacy protection mechanisms, have been developed based on different approaches

The issues of privacy and security of data emerged at a relatively premature stage in the development of data mining. To make easy these shortcomings, various directions have been actively pursued:

Data sanitization: The data points deemed sensitive cannot be directly data mined. It is anticipated that such modifications of data will not significantly impact the main findings given the total volume of data.

Data distortion/data perturbation/data randomization: It reaches the goal of privacy preserving by modifying individual data. The distortion does affect the values of individuals but its impact on the discovery and quantification of the main relationships is likely to be still quite negligible.

Cryptographic methods: This involves implementation of cryptographic techniques to devise protocols for communication among related entities in data mining environment. Cryptographic techniques are used mostly when multiple parties interact and want to hide their own content and still learn useful results from other parties. The one constraint faced by cryptographic solutions is the high communication and computation costs. The privacy achieved is but assumed to be as strong as the underlying technique used, but the cost of communication is high.

Assessing the relative performance of PPDM algorithms is a very difficult task, as it is often the case that no single algorithm outperforms others on all possible criteria. The relative merit of individual module that comprised by the PPDM algorithm are also rated.

Role of Secure multiparty computation can be found in [10].

Adversarial model's primary definition of available in SMC.

PPDM in SMC far away from accuracy[17].

Elements	Computational Cost	Privacy Preserving	Accuracy of Mining	Scalability
Hiding Purpose:				
Data Hiding	Low	Contingent	Contingent	High
Rule Hiding	High	Contingent	Contingent	Low
Data Mining Tasks:				
Classification	Low	N/A	Contingent	High
Clustering	High	N/A	Contingent	Low
Association Rule	Low	N/A	Contingent	High
Privacy Preserving Technique:				
Sanitation	Medium	Medium	Medium	Low
Distortion	Low	High	Low	High
Blocking	Medium	Low	Medium	Low
Generalization	Low	High	Medium	High
Cryptography	High	High	High	Low

Table. 1: Relative performance of PPDM Components [9]

A. Techniques

1) Shamir's Secret Sharing Scheme

Secret sharing scheme applied when there is an honest majority among the participants. Secret sharing scheme develops strong sharing in cryptography. A major problem with secret-sharing schemes is that the shares' size in the best known secret-sharing schemes realizing general access structures is exponential in the number of parties in the access structure. Though impractical, [11] proposed new protocol based on additive secret sharing that was proved more efficient and secure than the state of the art.

2) Threshold decryption

Threshold decryption is utilized by many efficient protocols providing solutions in both semi-honest [12] and malicious models [7].

3) Oblivious transfers

Oblivious transfer is a simple functionality involving two parties. It is a basic building block of many cryptographic protocols for secure computation. Using an implementation of oblivious transfer, and no other cryptographic primitive, it is possible to construct any secure computation protocol [13]. Oblivious transfer was suggested by Rabin in [14].

Input: The senders input are a pair of strings (x_0,x_1) and the receiver's input is a bit s Î {0, 1}.

Output: The receiver's output is x_s (and nothing else), while the sender has no output.

It is a little more challenging to construct oblivious transfer protocols which are secure against malicious adversaries. In order to adapt the oblivious transfer protocol described above, we must ensure that the receiver chooses the public keys appropriately. This can be done using zero-knowledge proofs that are used by the receiver to prove that it chooses the keys correctly. Oblivious transfer is often the most computationally intensive operation of secure protocols, and is repeated many times. Each invocation of oblivious transfer typically requires a constant number of invocations of trapdoor permutations (i.e., public-key operations, or exponentiations). It is possible to reduce the amortized overhead of oblivious transfer to one exponentiation per a logarithmic number of oblivious transfers, even in the case of malicious adversaries. Further, one can extend oblivious transfer in the sense that one has to compute, in advance, a small number of oblivious transfers. This then allows one to compute an essentially unlimited number of transfers at the cost of computing hash functions alone [14].

III. Zero-Knowledge Proof

A zero-knowledge proof is a way that a "prover" can prove possession of a certain piece of information to a "verifier" without revealing it.

This is done by manipulating data provided by the verifier in a way that would be impossible without the secret information in question. A third party, reviewing the transcript created, cannot be convinced that either prover or verifier knows the secret

A. Interactive proof protocols

In an interactive proof system, there are two parties:

- An (all powerful) Prover, often called Peggy (a randomized algorithm using a private random number generator);
- A (little (polynomially) powerful) Verifier, often called Vic (a polynomial time randomized algorithm using a private random number generator).

Prover knows some secret, or knowledge, or a fact about a specific object, and wishes to convince Vic, through a communication with him, that he has the above knowledge.

For example, both Prover and Verifier possess an input x and Prover wants to convince Verifier that x has a certain properties and that Prover knows how to proof that the interactive proof system consists of several rounds. In each round Prover and Verifier alternatively do the following.

1) Receive a message from the other party.
2) Perform a (private) computation.
3) Send a message to the other party.

Communication starts usually with a challenge of Verifier and a response by Prover.

At the end, Verifier either accepts or rejects Prover's attempts to convince Verifier [15].

Example - graph non-isomorphism

A simple interactive proof protocol exists for computationally very hard graph non-isomorphism problem.

Input: Two graphs G 1 and G 2, with the set of nodes {1,…,n }

Protocol: Repeat n times the following steps:

1) Vic chooses randomly an integer i Î {1,2} and a permutation p of {1,…,n }. Vic then computes the image H of G i under permutation p and sends H to Peggy.
2) Peggy determines the value j such that G J is isomorphic to H, and sends j to Vic.
3) Vic checks to see if i = j. Vic accepts Peggy's proof if i = j in each of n rounds.
4) Completeness: If G 1 is not isomorphic to G 2, then probability that Vic accepts is clearly 1

Soundness: If G 1 is isomorphic to G 2, then Peggy can deceive Vic if and only if she correctly guesses n times the i Vic choose randomly. Probability that this happens is 2 -n. Observe that Vic's computations can be performed in polynomial time (with respect to the size of graphs).

B. Interactive proof systems

An interactive proof protocol is said to be an interactive proof system for a secret/knowledge or a decision problem P if the following properties are satisfied.

Assume that Prover and Verifier possess an input x (or Prover has secret knowledge) and Prover wants to convince Verifier that x has a certain properties and that Prover knows how to proof that (or that Prover knows the secret).

(Knowledge) Completeness: If x is a yes-instance of P, or Peggy knows the secret, then Vic always accepts Peggy's “proof" for sure.

(Knowledge) Soundness: If x is a no-instance of P, or Peggy does not know the secret, then Vic accepts Peggy's “proof" only with very small probability [15].

C. Cheating

- If the Prover and the Verifier of an interactive proof system fully follow the protocol they are called honest Prover and honest Verifier.
- A Prover who does not know secret or proof and tries to convince the Verifier is called cheating Prover.

A Verifier who does not follow the behavior specified in the protocol is called a cheating verifier.

D. Zero-knowledge proof protocols informally

Very informally An interactive “proof" protocol at which a Prover tries to convince a Verifier about the truth of a statement, or about possession of a knowledge, is called “zero-knowledge” protocol if the Verifier does not learn from communication anything more except that the statement is true or that Prover has knowledge (secret) she claims to have.

Example The proof n = 670592745 = 12345 ´ 54321 is not a zero-knowledge proof that n is not a prime

Informally A zero-knowledge proof is an interactive proof protocol that provides highly convincing evidence that a statement is true or that Prover has certain knowledge (of a secret) and that Prover knows a (standard) proof of it while providing not a single bit of information about the proof (knowledge or secret). (In particular, Verifier who got convinced about the correctness of a statement cannot convince the third person about that.)

More formally A zero-knowledge proof of a theorem T is an interactive two party protocol, in which Prover is able to convince Verifier who follows the same protocol, by the overwhelming statistical evidence, that T is true, if T is indeed true, but no Prover is not able to convince Verifier that T is true, if this is not so. In additions, during interactions, Prover does not reveal to Verifier any other information, except whether T is true or not. Consequently, whatever Verifier can do after he gets convinced, he can do just believing that T is true.

Similar arguments hold for the case Prover possesses a secret.

E. Illustrative example

(A cave with a door opening on a secret word)

Alice knows a secret word opening the door in cave. How can she convince Bob about it without revealing this secret word?[16]

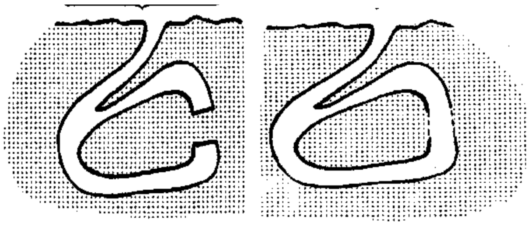

Fig. 1: Zero Knowledge Proof

F. Properties of Zero-Knowledge Proofs

Completeness – A prover who knows the secret information can prove it with probability 1

For all x 2 L,

Prob[t Ã (P; V)(x); V (x; t) = ACCEPT] = 1:

Soundness – The probability that a prover who does not know the secret information can get away with it can be made arbitrarily small

For all X =2 L, and any Turing machine P0, it holds that

Prob[t Ã (P0; V)(x); V (x; t) = ACCEPT] • 1=2:

IV. PROPOSED APPROACH

p (p1, p2, …, pn) is the number of parties involved in protocol
k (k1, k2, …, kn) is the number of cluster means
a (a1, a2, …, an) is the number of attributes which stays same in horizontal partitioning
t (t1, t2, …, tn) is the number of transactions with each party
For each party pi , do in parallel

A. Initialize

```
1) randomly select k0 to kn from t as initial cluster means
2) set them at global k-means
```

B. Do k-means clustering

```
1) For each ki in global k-means,
      i) Create distance metric for each t
2) For each ti,
      i) Select the nearest cluster mean k
3) For each ki,
      i) Calculate the mean mi of all ti in ki
4) For each ki,
      i) new ki = mi
5) Repeat steps 1 to 5 until there is no change in k - means.
```

C. Privately sharing locally generated clustering Information

```
1) For each ki,
      i) For each ai in k,
      a. Generate equations and send the shares
      b. Receive the secret shares from all parties
      c. Prove and verify the mean values shared using ZKP.
            1) Verify the partial sum from each pi.
            2) Prove the partial sum to each pi.
            3) In case of successful verification proceed with protocol
      d. Solve the linear equations
      e. Calculate average value for attribute ai.
2) For each ki received,
      i) update global k-means
3) Repeat steps 2 and 3 until no change in global k-means
```

D. Privacy preserving in proposed approach

- All parties agree upon the random numbers (x1 for Party 1, x2 for Party 2 , x3 for Party 3)
- E.g. (x1, x2, x3) = (3, 6, 2)
- Each Party selects one polynomial which is not known to other parties.
- E.g. Party 1 selects a polynomial 2x2 + x + k, where k is the actual value which they want to send.

V. CONCLUSION

This research concludes that efficient protocols in malicious model can be devised with the help of interactive zero-knowledge proofs systems on existing solutions in semi-honest models. The same can be implemented on data mining algorithms for clustering. Although semi-honest model is considered more near to realistic environment, it does not guarantee protection from colluding parties. The security in malicious model is more preferable where privacy preserving algorithm needs to be run in environment with majority of dishonest participants.

REFERENCES

[1] Alex Gurevich and Ehud Gudes “Recent Research on Privacy Preserving DataMining” Department of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

[2] Amos Beimel “Secret-Sharing Schemes: A Survey”Dept. of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel S. Preethi,B.Ramchandran,"Energy Efficient ROuting Protocols for Mobile Adhoc Networks" IEEE,2011

[3] W. Du and M. J. Atallah. Secure multi-party computation problems and their applications: A review and open problems. In Proceedings of the 2001 New Security Paradigms Workshop, Cloudcroft, New Mexico, Sept. 11-13 2001

[4] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game – a completeness theorem for protocols with honest majority. In 19th ACM Symposium on the Theory of Computing, 1987.

[5] Jaideep Vaidya, Chris Clifton: Secure set intersection cardinality with application to association rule mining. Journal of Computer Security 13(4): 593-622 (2005)

[6] J.Vaidya, C.Clifton. Privacy Preserving Association Rule Mining in Vertically Partitioned Data. In Proceedings of SIGKDD 2002, Edmonton, Alberta, Canada.

[7] K. Emura, A. Miyaji, and M. S. Rahman, “Efficient Privacy-Preserving Data Mining in Malicious Model,” ADMA, vol. 6440, pp. 370-382, 2010.

[8] M. Kantarcioglu and O. Kardes, "Privacy-preserving data mining in the malicious model", International Journal of Information and Computer Security, Vol. 2, No. 4, pp. 353-375, 2008.

[9] H.-P. Kriegel, K. M. Borgwardt, P. Kröger, A. Pryakhin, M. Schubert, and A. Zimek, “Future trends in data mining,” Data Mining and Knowledge Discovery, vol. 15, no. 1, pp. 87-97, Mar. 2007.

[10] Y. Lindell and B. Pinkas, “An Efficient Protocol for Secure Two-Party Computation in the Presence of Malicious Adversaries,” EUROCRYPT, vol. 4515, no. 860, pp. 52-78, 2007.

[11] T. B. Pedersen, A. Levi, M. Doganaay, E. Savas, Y. Saygin and A. Levi, “Distribute Privacy Preserving K-Means Clustering with Additive Secret Sharing”, PAIS, 3-11, 2008.

[12] J. Vaidya, W. Lafayette and C. Clifton, “Privacy-Preserving K-Means Clustering over Vertically Partitioned Data”, ACM SIGKDD ’03, 2003.

[13] Michael O. Rabin, “How to Exchange Secrets with Oblivious Transfer”, Technical Report TR-81, Aiken Computation Lab, Harvard University, 1981.

[14] Y. Lindell and B. Pinkas, “Secure Multiparty Computation for Privacy-Preserving Data Mining”, In The Journal of Privacy and Confidentiality, v. 1, no. 1, pp. 59-98, 2009.

[15] Feigenbaum, J., “Overview of Interactive Proof Systems and Zero-Knowledge”, Contemporary Cryptology, G.J. Simmons, ed., pp. 423-440, IEEE Press 1992

[16] Quisquater, J.J., L. Guillou, T. Berson, “How to Explain Zero-Knowledge Protocols to Your Children”, Advances in Cryptology - CRYPTO ’99, Lecture Notes in Computer Science 435, pp. 628-631, 1990

[17] Smita Patel “Privacy Preserving Data Mining” ISSN: 2277-3754 January 2013