ISSN: 0975-5662, August, 2014 www.ijrct.org

Preserving the Data by Anonymous ID assignment using Sturms Theorem

#**P.Sathish Kumar** [1], [1]*M.Tech, Computer Science Engineering, sathishkumar.paka@gmail.com*

#**K.Arpitha**[2], Assistant *Professor, Department of CSE, kottearpitha@gmail.com*

#**Md.Fareeduddin**[3], [3]Associative *Professor, Department of CSE,* mfareed_kzmn@yahoo.co.in

[#] *TallaPadmavathi Engineering College, Warangal, Telangana, India*

Abstract:

To overcome the problems in privacy preserving data mining, collision in communications, distributed database access in the wireless communications, the Secure Sum algorithm is used. For sharing private data securely among several parties an algorithm has been used. An anonymous ID assignment technique is used iteratively to assign the nodes with ID numbers ranging from 1 to N. This ID assignment is anonymous in that the identities received are unknown to the other members of the group. When private communication channels used, the resistance to collusion among other members is verified in an information theoretic sense. The required computations are distributed without using a trusted central authority. The algorithm has been compared with the existing algorithm. New algorithm has been developed based on Sturm's theorem and Newton's identities. The numbers of iterations are found out with the help of Markov chain.

1. Introduction:

The anonymous communication plays an important role in internet's popularity for both personal and business purposes. Cloud based website management tools enable the servers to analyze the user's behavior. There are some disadvantages of sharing private data such as applications for anonymity are patient medical records, social networking, electronic voting and many more. In secure multiparty computation which is another form of anonymity allows several multiple parties to share data that remains anonymous. A secure computation function enables multi parties to calculate the sum of their inputs rather than exposing the data. This method is very much popular in data mining operations and enables classifying the complexities of secure multiparty computation. Our main algorithm is built on top of a method that shares simple data anonymously and yields a method that enables sharing of complex data anonymously. With the help of permutation methods the assigned ID are known only to the nodes which are being assigned IDs. There are several applications where network nodes needs dynamic unique IDs. One such application is grid computing where the services are requested without disclosing the identities of the service requestor. We differentiate anonymous communication and anonymous ID assignment, consider a situation where N parties wish to display their data altogether, in N slots on a third party site,

anonymous ID assignment method assigns N slots to the users whereas anonymous communication allows the users to conceal their identities In our network the identities of the parties are known but not the true identity. In this project we use an algorithm for sharing simple integer data which is based on secure sum. This algorithm is used in every iteration of anonymous ID assignment. Here we consider all the nodes to be semi honest. Even though they follow a set of rules for communication if they happen to see information they might intrude. Mental poker is the common name for a set of cryptographic problems that concerns playing a fair game over distance without the need for a trusted third party. The term is also applied to the theories surrounding these problems and their possible solutions. The name stems from the card game poker which is one of the games to which this kind of problem applies. A similar problem is flipping a coin over a distance. The disadvantage is that allow only authorized actors to have access to certain information while not using a trusted arbiter. It has applications to other problems in privacy preserving data mining, collision avoidance in communications and distributed database access. The usage of secure multiparty computation is being avoided with the usage of Sturm's theorem to make sure that the information about the nodes are not revealed. In the current system the main goal is to provide anonymous id for each node. Each node will have a secure communication of simple and complex data. Those data's may be from static data or dynamic data. By implementing secure sum hides permutations method and anonymous id assignment (AIDA) method the permutation methods are kept anonymous to each other. Hence here encoding technology is used to create anonymous ID and the ID is being assigned to the user by the central authority and the receiver receives the data and decodes it with the key that is known only to the sender and the receiver which might not be known to the other semi honest node that might intrude.

2. Existing System:

Mental Poker is the playing of poker without cards over a telecommunications device (phone or more realistically internet). It doesn't include a trusted third party dealer or a source of randomness and as such it seems that someone (the dealer) will always know what cards have been given out or alternatively, that players will be able to lie about the cards they have. In fact it has been proven using information theory that the mental poker problem is impossible, however we aim to show that using cryptographic techniques it can be made infeasible to determine the other players cards so that the problem is solved practically.

The first discussion of Mental Poker was in 1933 when Niels Bohr, his son Christian, Felix Bloch, Carl Friedrich and Werner Heisenberg attempted a game while on vacation. The attempt was unsuccessful. The first serious attempt on the problem was by Adi Shamir, Ronald Rivest and Leonard Adleman in 1979 in [SRA]. It is this scheme, which relies on commutative encryption,

which this paper concentrates on. The authors first proved, in an information theoretic sense, that the problem is unsolvable and then went on to offer a solution. Their protocol worked for two players and didn't require a trusted third party. However it did not offer confidentiality of strategy, requiring the players to reveal their hands at the end of each game.

The SRA protocol was shown to leak at least one bit of information: whether the card was a quadratic residue or not. This was first shown in [Lip] by R. Lipton in 1981. There were suggestions to overcome this problem but there was still no guarantee that other information was not leaked. With SRA seemingly flawed other protocols using other approaches were used. In [GM] Goldwasser and Micalli used probabilistic encryption to solve the mental poker problem as discussed in this report. However it required the players to show their cards at the end of each hand to prevent cheating. In 1987 Crepeau in [Cre] provided a solution that allows complete confidentiality of strategy so cards don't need to be shown if a player has already lost. Crepeau used zero knowledge proof, basically a way of showing that you know a secret without saying what that secret is. Although it was a good theoretical solution to the problem it is computationally infeasible, in 1994 it took 8 hours to shuffle a deck using the protocol as shown in [Edw]. Much research has been done since then into implementation but that is beyond the scope of this report.

We consider the SRA algorithm as given in [SRA] for mental poker between 2 players.

This protocol depends on the existence of a commutative encryption scheme, one in which EB(EA(M)) = EA(EB(M)) i.e. the encryption of a message is the same if we encrypt with key A then B as if we encrypt with B then A. The encryption scheme requires a large prime number, n, chosen by both players. Note that _(n) = n − 1. We choose the plaintext, ciphertext and key spaces to all be Zn. The key A must be chosen such that gcd(A, _(n)) = 1. Then encryption is carried out by EA(M) = MA (mod n) and decryption is simply DA(C) = CA−1(mod n). So DA(EA(M)) = (MA)A−1 = M as required. Also EB(EA(M)) =(MA)B = (MB)A = EA(EB(M)) so commutativity holds as we wished.

We also require a way to encode the 52 cards as integers modulo n. In [SRA] it is suggested to use the ASCII values of the card names, for example "THE TWO OF CLUBS", however we will require only that the values of numbers chosen are in the same order as the unsorted cards. It would seem simplest to encode the two of clubs as 1, the three of clubs as 2, and so on, however then EA(1) = 1 so it easy to determine which card is the two of clubs.

3. The algorithm

An algorithm for shuffling cards using commutative encryption would be as follows:

1. Alice and Bob agree on a certain "deck" of cards. In practice, this means

ISSN: 0975-5662, August, 2014 www.ijrct.org

they agree on a set of numbers or other data such that each element of the set represents a card.

2. Alice picks an encryption key A and uses this to encrypt each card of the deck.
3. Alice shuffles the cards.
4. Alice passes the encrypted and shuffled deck to Bob. With the encryption in place, Bob cannot know which card is which.
5. Bob picks an encryption key B and uses this to encrypt each card of the encrypted and shuffled deck.
6. Bob shuffles the deck.
7. Bob passes the double encrypted and shuffled deck back to Alice.
8. Alice decrypts each card using her key A. This still leaves Bob's encryption in place though so she cannot know which card is which.
9. Alice picks one encryption key for each card (A_1, A_2, etc.) and encrypts them individually.
10. Alice passes the deck to Bob.
11. Bob decrypts each card using his key B. This still leaves Alice's individual encryption in place though so he cannot know which card is which.
12. Bob picks one encryption key for each card (B_1, B_2, etc.) and encrypts them individually.
13. Bob passes the deck back to Alice.
14. Alice publishes the deck for everyone playing (in this case only Alice and Bob, see below on expansion though).

The deck is now shuffled.

During the game, Alice and Bob will pick cards from the deck, identified in which order they are placed in the shuffled deck. When either player wants to see their cards, they will request the corresponding keys from the other player. That player, upon checking that the requesting player is indeed entitled to look at the cards, passes the individual keys for those cards to the other player. The check is to ensure that the player does not try to request keys for cards that do not belong to that player.

3.1 Weakness or disadvantage

The encryption scheme used must be secure against known-plaintext attacks: Bob must not be able to determine Alice's original key A (or enough of it to allow him to decrypt any cards he does not hold) based on his knowledge of the unencrypted values of the cards he has drawn. This rules out some obvious commutative encryption schemes, such as simply XORing each card with the key.

4. Proposed System

This paper builds an algorithm for sharing simple integer data on top of secure sum. The sharing algorithm will be used at each iteration of the algorithm for anonymous ID assignment (AIDA). This AIDA algorithm, and the variants that we discuss, can require a variable and unbounded number of iterations. Increasing a parameter in the algorithm will reduce the number of expected rounds. However, our central algorithm requires solving a

ISSN: 0975-5662, August, 2014 www.ijrct.org

polynomial with coefficients taken from a finite field of integers modulo a prime. That task restricts the level to which can be practically raised.

4.1 Sturms Theorem

The usage of secure multiparty computation is being avoided with the usage of Sturm's theorem to make sure that the information about the nodes are not revealed. In the current system the main goal is to provide anonymous id for each node. Each node will have a secure communication of simple and complex data. Those data's may be from static data or dynamic data. By implementing secure sum hides permutations method and anonymous id assignment (AIDA) method the permutation methods are kept anonymous to each other.

4.1.1 Sturm's Theorem AIDA

It is possible to avoid solution of the Newton polynomial entirely. Sturm‟s theorem allows the determination of the number of roots of a real polynomial p(x) in an interval (a,b) based on the signs of the values of a sequence of polynomials derived from p(x) . The sequence of polynomials is obtained from a variant of the Euclidean Algorithm. As in the previous variant, the power sums are collected and the Newton Polynomial is formed. However, the field used for computation is the field of rational numbers **Q**. The test p′(ri)=0 is again sufficient to determine whether or not ni has received. There is a computational advantage which is arises in that nodes which do not need to solve the Newton polynomial p(x) to determine the (now implicitly) shared values. Assume that x=0 is not a root of p(x) as xk has been factored out immediately if applicable. Each node ni which has received an assignment must count separately multiple roots and also forms g(x)=gcd(p(x),p′(x)). A multiple roots version of Sturm‟s theorem [32] is then applied to calculate the number of roots for the polynomial p(x) in the range (0,ri). (Note that ri itself is not a multiple root allowing application of the theorem). The polynomial g(x)=gcd(p(x),p′(x)) is a by-product of this computation. The same Sturm procedure is applied to g(x) thus obtaining a count of the multiple roots in the same range,(0,ri). The collected power sums Pi are integers. To guarantee the privacy and the compute sums using a field GF(P) with P greater than any possible value of Pi. Our timings showed that using Sturm‟s theorem is not currently competitive with the various methods of polynomial solution using the "prime modulus" approach and runs twice as slow as best. Although, the construction is straight forward. The application of Sturm‟s theorem requires the use of an ordered field resulting in the large polynomial coefficients. Unfortunately, the analog of this result which is usable for a finite field of the corresponding polynomial coefficient. Still, some results in this direction are available.

5. Conclusions

The required computations are distributed without using a trusted central

authority. The algorithms for assigning the anonymous ID is examined by the trade-offs between the requirements of communication and computations. The new algorithms are built on top of a secure sum data mining operation using Newton‟s identities and Sturm‟s theorem. An algorithm for the distributed solution of certain polynomials over finite field will enhances the scalability of the algorithms.

6. References:

1. SRA - Shamir, A., Rivest, R. and Adleman, L., ”Mental Poker”, in Mathematical Gardner, D.E. Klarner, ed., Wadsworth International, 1981, pp. 37-43.
2. Cre - C. Crepeau, A Zero-Knowledge Poker Protocol that Achieves Confidentiality of Players’ Strategy or How to Achieve an Electronic Poker face, Crypto ’86, 1987, pp. 239-247.
3. FM - Fortune, S. and Merrit, M., ”Poker Protocols”, in Advances in Cryptology: Proc. of Crypto 84, G. R. Blakley and D. Chaum, eds., Lecture Notes in Computer Science 196, SpringerVerlag, Berlin, 1985, pp. 454- 464.
4. A. Shamir, R. Rivest, and L. Adleman, "Mental Poker", Technical Report LCS/TR-125, Massachusetts Institute of Technology, April 1979.
5. Goldwasser, S. and Micali, S. 1982. Probabilistic encryption & how to play mental poker keeping secret all partial information. In Proceedings of the Fourteenth Annual ACM Symposium on theory of Computing.
6. [STA05] Stamer, H. Efficient Electronic Gambling: An Extended Implementation of the Toolbox for Mental Card Games. WEWoRC 2005, LN P-74, 1-12, 2005
7. [SCH98] Schindelhauer, C. A Toolbox for Mental Card Games. Tech. Rep. of Medizinische Universitat Lubeck.
8. [GOL05] Golle, P. Dealing Cards in Poker Games. In Proceedings of the International Conference on Information Technology: Coding and Computing, (2005)

