INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

PRIVACY PRESERVING ASSOSIATION RULE MINING OVER DISTRIBUTED SYSTEM USING FUZZY LOGIC

Anurag Singh[1], Dr. Amod Tiwari[2]

[1]Sainath University,
Dept. of Computer Science and Engineering,Ranchi
anuragphd001@gmail.com
[2]Prof. Bhabha Institute of Technology Kanpur
amodtiwari@gmail.com

Abstract: *Privacy distributed homogenous data is one of the most important properties of an information system must satisfy, by which systems share information among mutual distributed exclusion, by relational entities, the preservation of sensible information has a relevant role. A relatively new trend shows that modern access control techniques are sufficient to guarantee privacy preservation but need to improve the technique when data mining and fuzzy sets are reduced cost of preserved data. Privacy preserving data mining algorithms have been recently introduced with the aim of preventing the discovery of sensible information. In this paper we proposed a modification to privacy preserving association rule mining on distributed homogenous database algorithm. Our algorithm is faster than old one which modified with preserving privacy and accurate results.*

1. INTRODUCTION

When Solution to the distributed homogenous data problem consists of a protocol to be executed among the processes of the distributed system solely by passing messages in order to allow one or some processes to execute private operations with one or several shared resources. In a centrally controlled system [1], it is not too difficult to implement the homogenous data on the shared object. Semaphores and monitors are commonly used. However, in a distributed environment, the solution to this problem becomes far more complex due to the absence of a global or centralized controller. In a distributed system, nodes communicate only by passing messages. A distributed homogenous data algorithm requires an approach such that if a node wishes to enter in homogenous data then all other nodes must be aware of this that a process has already entered in critical section, and hence they themselves cannot enter into their critical section [2]. There are number of techniques available for this. In centralized system there is the problem of congestion because only one administrator is responsible to manage the complete network but it's simple to implement [3].In distributed system message complexity is very high because no node has the information about the availability of the token, so the node that is wishing to enter into critical section has to send the request messages to all the nodes that are the part of the system in search of the token.

2. LITERATURE SURVEY

In [14] Algorithm requires 0 or at most N number of messages to enter into critical section. A node having the token is allowed to enter into the critical section. A single node has the privilege and a node requesting critical section, broadcasts a message to all the other nodes. A site sends the privilege if the token is idle with the site. The site having token can continuously enter critical section until it sends the token to some other site. The request message [13-14] has the format REQUEST (j,n), which means site j is requesting its nth critical section. Each node maintains an array RN of size N for recording latest sequence number received from each of the other nodes. The PRIVILEGE message has the format PRIVILEGE (Q, LN), where Q is queue of nodes requesting critical section and LN is an array of size N where LN[j] is the latest critical section executed by a node j. If RN[j] = LN[j]+1 means a node j has sent a request for its new sequence of critical section, and the node having the privilege adds this to the queue and if token is idle sends the node sends the PRIVILEGE(LN,Q) to the node requesting critical section. Number of message per critical section entry is (N-1) REQUEST messages plus 1 PRIVILEGE message so N messages in all or 0 if the node having the token wants to enter critical section. In this algorithm [16] nodes are arranged in an un-rooted tree structure. All messages are sent along the undirected edges of the tree. Every node knows about the existence of its immediate neighbours. Again a PRIVILEGE message [6] has to be received by a node to enter into critical section. At every node a variable HOLDER points to a node along the path to the PRIVILEGE. A node having the PRIVILEGE the HOLDER points to itself. When a non-privileged node [11, 12] wants to enter critical section it generates a request and adds it to its REQUESTQ, which is a queue maintained by each node. If it has not sent a message along the directed path towards the node pointed by the holder variable, it sends a message along the edge to the token holder. On receiving a message, the nodes forward the message to the token holder along path [5]. However, before forwarding the nodes add the request in their

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

REQUEST Q. When the request reaches to the node having the PRIVILEGE, then if the node is not executing the critical section, it sends the PRIVILEGE to that node from which the message is received. On receiving the PRIVILEGE if the nodes own id is on the top of the queue, it [9,10] executes critical section else sends the PRIVILEGE to the node pointed by the id, and set its holder to point to that node. The number of messages required to execute critical section can be 0 or typically 2D, where D is the diameter of the tree on which the algorithm [7] is running, however this is reduced to maximum of four messages per critical section execution under full load when the topology is proper tree and two messages when it's a chain.

This algorithm [4] makes use of state information which is defined as the set of states of homogenous data processes in the system. Each site maintains information about the state of other sites and uses it to deduce a subset of sites likely to have the token. Consequently, the number of messages exchanged for a critical section invocation is a random variable between 0 and n (n is the number of sites in the system). Sites use sequence numbers to distinguish between a current token request and old delay and token request. Every site keeps a counter. When a site has to execute its CS, it increments its counter and sends the updated value (called sequence number) in token request messages. Each site keeps a record of the highest known sequence number (along with the latest known state information) of each site. By comparing the sequence number in a received message with the latest known sequence number of its sender site, the token to a requesting site with the lowest sequence number is granted. [17] assume that there exist two layers in the system: the application layer (the higher layer) and the GME layer (the lower layer). The interface between the two layers is implemented [15] by using two types of messages: Request-Session and Grant-Session. When the application layer needs to access a session, says Session X, the process running the application layer sends the message Request-Session(X) to the GME layer [8]. Eventually, the GME layer grants the application layer the access to Session X by sending the message Grant Session. The size of messages is 2 x log (m + 1) bits only. Every resource request generates O (n2) messages in the worst case, but zero messages in the best case.

In [16] have presented a new token based protocol for group homogenous data in distributed systems. The protocol uses one single token to allow multiple processes to enter the critical section for a common session. One of the significant characteristics of the protocol is concurrency; throughput and waiting time can be regulated adjusting the time period for which a session is declared. The minimum and the maximum number of messages to enter the CS is 0 and (n + 2) respectively where n is the total number of processes in the system.

3. MODIFICATION

Token ring based distributed systems it is very common that, resources are being shared among various processes, with the condition that a single resource can be allocated to a single process at a time. Therefore, homogenous data is a fundamental problem in any distributed computing system. The number of messages exchanged for an entry into a critical section to take effect will be used as a complexity measure. So, the goal is to find a solution that will synchronize the access among shared resources in order to maintain their consistency and integrity.

The new association rule for distributed system data use fuzzy logic and cryptography keys. In this process provide the protection on every distributed system data by using privacy preserving algorithm with the help of node setting of sets. By the process reduce the computational time and execution time an algorithm for association rule distributed system data.

Find all homeomorphisms from Z to Z and from F_2 to $Z_3 \times Z_3$.. The approach of picking where generators of a group go and then "extending" the homomorphism to the rest of the group very often comes in handy. However, this can only be done when the elements the generators are sent to satisfy all the relations between the generators themselves. Recall that Z3 is generated by R_{120}. Suppose we try to define a homomorphism f: $Z_3 \rightarrow Z$ by letting $f(R_{120})=1$, sending a generator to a generator. Does this extend to a homomorphism? What relation does R120 satisfy that $1 \in Z$ does not? There are many homomorphism from F2 to Z×Z.Take for instance f(a)=(1,0) and f(b)=(0,2).If you're itching for a challenge, try to find all the homeomorphisms from Z×Z to F_2. What do they have in common?

```
While (true)
  Do
  {
    Select a group g € G;
    Request (g);
    - Entry protocol.
       Critical Section
       Release;
  Exit Protocol.
  }
```

Suppose f:G→H is a homomorphism between two groups, with the identity of G denoted eG and the identity of H denoted eH. Show that $f(e_G)=e_H$, that is, identity is sent to identity by any homomorphism. It is clear that use the fact that e=ee and the defining property of homomorphism's.Consider the map f: $Z_9 \rightarrow Z_3$ given by $f(R_m)=R_{3m}$ (recall that Rm is a counterclockwise rotation by m degrees). Is this a homomorphism? Find a homomorphism from Z6 to Z3. is the map f:$Z_6 \rightarrow Z_5$ given by $f(R_m)=R_0$ (the identity) a homomorphism? Find a homomorphism from F_2 to Z×Z The performance parameters of interest are the average time to enter the critical

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

section, the average number of messages per critical section entry and the average information per message. The existing papers on K homogenous data typically present an analytical estimate of the average number of messages required per CS entry. The average number of messages is inadequate to measure the algorithm performance, because an algorithm that requires small number of messages may result in large delays in entering the CS.

4. CONCLUSION

Privacy distributed homogenous data has checked the proposed algorithm and to get right information and satisfied system. The data share information among mutual distributed exclusion, by using relational entities, the preservation of sensible data information has a relevant role using proposed algorithm. The proposed algorithm show that the modern access control techniques and sufficient to guarantee privacy preservation of data information. In the improve technique we have use specific tool of data mining and fuzzy sets to reduced cost of preserved data. The algorithm is great older than other algorithm for privacy preserving data because the fuzzy set "Z" has been introduced with K-homogenous discovery of sensible information.

REFERENCES

[1] Mohamed Naimi, Michel Trehel, and Andre Arnold. A log(n) distributed homogenous data algorithm based on path reversal. J. Parallel Distrib. Comput. 34(1):1-13, 1996

[2] Ichiro Suzuki and Today Kasami. A distributed homogenous data algorithm. ACM Trans. Comput. Syst., 3(4):344-349, 1985.

[3] Kerry Raymond. A tree-based algorithm for distributed mutual exclusion. ACM Trans. Comput. Syst., 7(1):61-77, 1989.

[4] Mukesh Singhal, A heuristically-aided algorithm for homogenous datain distributed systems. IEEE Trans. Comput., 38(5):651-662, 1989.

[5] Quazi Ehsanul Kabir Mamun and Hidenori Nakazato. A new token based protocol for group homogenous datain distributed systems. In ISPDC, pages 34- 41, 2006.

[6] Sebastien Cantarell, Ajoy Kumar Datta, Franck Petit, and Vincent Villain. Token based group homogenous data for asynchronous rings. In ICDCS, pages 691- 694, 2001.

[7] Kerry Raymond, A Tree-Based Algorithm for Distributed Mutual Exclusion, ACM, (1989) ISBN 0734-2071/89/0200-0061.

[8] Supriya Madhuram, Anup Kumar, A Hybrid Approach for Homogenous datain Distributed Computing Systems, IEEE, (1994) ISBN 0-8186-6427-4/94.

[9] Quazi Ehsanul Kabir Mamun, Hidenori Nakazato, A New Token Based Protocol for Group Homogenous datain Distributed Systems, IEEE, 2006, ISBN 0-7695-2638-1/06.

[10] Cormen, Thomas H. Leiserson, Charles E. Rivest, Ronald L. Stein, Cliord, MIT Press and McGraw-Hill, 2009, ISBN 0-262-03293-7.

[11] Andrew S.Tanenbaum, Maarten Van Steen, Distributed systems, Pearson Education, 2007, ISBN 0-13-239227-5

[12] Michael Beye, Zekeriya, Information Security and Privacy Lab, Faculty of EEMCS, Delft University of Technology 2628 CD, Delft, The Netherlands. 978-1-4577-1019-3/11/$26.00 ©2011 IEEE

[13] J Vaidya and C. Clifton, Privacy preserving association rule mining in vertically partitioned data. In Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2002, pp.639-644.

[14] Ichiro Suzuki and Today Kasami. A distributed homogenous dataalgorithm. ACM Trans. Comput. Syst., 3(4):344-349, 1985.

[15] Quazi Ehsanul Kabir Mamun and Hidenori Nakazato. A new token based protocol for group homogenous data in distributed systems. In ISPDC, pages 34-41, 2006.

[16] Kerry Raymond. A tree-based algorithm for distributed mutual exclusion. ACM Trans. Comput. Syst., 7(1):61-77,1989. [4] Mukesh Singhal. A heuristicallyaided algorithm for homogenous datain distributed systems. IEEE Trans. Comput., 38(5):651-662, 1989.

[17] Sebastien Cantarell, Ajoy Kumar Datta, Franck Petit, and Vincent Villain. Token based group homogenous datafor asynchronous rings. In ICDCS, pages 691-694, 2001.