IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed ( Group -I) Journal Volume 11, Iss 12, 2022

# SECURE ASSOCIATION RULE MINING IN HORIZONTALLY DISTRIBUTED DATABASESUSING THRESHOLD-C AND UNIFY PROTOCOLS

**CH.RAVI, G ARPITHA**
Professor, Dept. of CSE, CMR Institute of Technology, Hyderabad, India.
[2]Associate Professor, Dept. of CSE, TKR College of Engineering and Technology, Hyderabad, India.

**Abstract:** Data mining is the drastic and fast growing area today which is used to extract knowledge from huge data collections but often these collections are separated among various parties. Privacy accountability may prevent the parties from honestly giving out the data and some sort of information about the data. In this project we propose a protocol for secure association rule mining in horizontally distributed databases. The existing integral protocol is that of Kantarcioglu and Clifton well known as K&C protocol. This protocol is based on an unsecured distributed version of the named as Fast Distributed Mining (FDM) algorithm of Cheung et al. The main constituents in our protocol are two novel secure multi-party algorithms one that process the union of private subsets that each of the interacting players hold and another that check whether an element held by one player is included in a subset held by another. This protocol suggests enhanced privacy with respect to the former protocols. In addition, it is not complex and is prominently more powerful in terms of communication cost, communication rounds and computational cost.

**Keywords:** FDM, DW, FDM –KC, UNIFY-KC

## I. INTRODUCTION

Data mining can extract important knowledge from large data collections but sometimes these collections are split among various parties. Privacy liability may pre-vent the parties from directly sharing the data, and some types of information about the data. Data mining technology has become prominent as a means of identifying patterns and trends from large quantities of data. Data mining and data are housing co-jointly: most popular tools operate by gathering all data into a central site then running an algorithm against that data. However, privacy liability can prevent building a centralized warehouse data may be distributed among several custodians none of which are allowed to transfer their data to another site. In Horizontally partitioned database there are several layers that hold homogeneous database. The goal is to find all association rules with support at least s and confidence at least c, for some given minimal support size s and confidence level c, that hold in the unified database, while minimizing the information disclosed about the private databases held by those players. That goal defines a problem of secure multi-party computation. If there existed a trusted third party, the players could surrender to him their inputs and he would perform the function evaluation and send to them the resulting output.

In the absence of such a trusted third party, it is needed to devise a protocol that the players can run on their own in order to arrive at the required output y. Such a protocol is

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed ( Group -I) Journal Volume 11, Iss 12, 2022

considered perfectly secure if no player can learn from his view of the protocol more than what he would have learnt in the idealized setting where the computation is carried out by a trusted third party. In previous year various techniques are applied for secure mining of association rules in horizontally partitioned database. These approaches use various techniques such as data perturbation, homomorphic encryption, keyword search and oblivious pseudorandom functions etc. These privacy preserving approaches are inefficient due to

- Homo-morphic encryption
- Higher computational cost
- In some of the techniques data owner tries to hide data from data miner.

Our proposed protocol based on two novel secure multiparty algorithm using these algorithms the protocol provides enhanced privacy, security and efficiency as it uses commutative encryption. In this project we propose a protocol for secure mining of association rules in horizontally distributed database. This protocol is based on: FDM Algorithm which is an unsecured distributed version of the Apriori algorithm. In our protocol two secure multiparty algorithms are involved:

- Computes the union of private subsets that each interacting players hold.
- Tests the inclusion of an element held by one player insubset held by another.
- In Horizontally partitioned database there are several players that hold homogeneous database. Our protocol offers enhanced privacy with respect to the current leading K and C protocol simplicity, more efficient in terms of communication rounds, communication cost and computational cost. In our problem, the inputs are the partial databases and the required output is the list of association rules that hold in the unified database with support and confidence no smaller than the given thresholds s and c, respectively.

## II. PROPOSED APPROACH

### A. Secure Multiparty Protocol for Computing the or Of Private Binary Vectors

Protocol UNIFI-KC securely computes of the union of private subsets of some publicly known ground set (Ap(Fk−1s )). Such a problem is equivalent to the problem of computing the OR of private vectors. Indeed, if the ground set is $\Omega = \{\omega_1, \ldots, \omega_n\}$, then any subset B of Ω may be described by the characteristic binary vector $b = (b_1, \ldots, b_n) \in Z_2^n$ where $b_i = 1$ if and only if $\omega_i \in B$. Let $b_m$ be the binary vector that characterizes the private subset held by player $P_m$, $1 \le m \le M$. Then the union of the private subsets is described by the OR of those private vectors, $b := \bigvee_{m=1}^{M} b_m$. Such a simple function can be evaluated securely by the generic solutions suggested in [3], [5], [15]. We present here a protocol for computing that function which is much simpler to understand and program and much more efficient than those generic solutions. It is also much simpler than Protocol UNIFIKC and employs less cryptographic primitives. Our protocol computes a wider range of functions, which we call threshold functions.

 

### B. THRESHOLD Protocol

Let $P_1, \ldots, P_M$ be M players where $P_m$ has an input binary Vector $b_m \in Z_2^n$, $1 \le m \le M$. Protocol 2 (to which we refer as THRESHOLD henceforth) computes, in a securemanner, the output vector b := $T_t(b_1, \ldots, b_M)$, for some $1 \le t \le M$. Let a = (a(1), . . . , a(M)) := $\sum_{m=1}^{M} b_m$ be the sum of

the input binary vectors. Since a(m) $\in Z_{M+1}$ = {0, 1, . . .

,M}, for all $1 \le m \le M$, the sum vector a may be seen as a

vector in Zn M+1. The main idea behind the protocol is to use the secure summation protocol of [6] in order to compute shares of the sum vector a and then use those shares to securely verify the threshold conditions in each component. Since a $\in Z_{M+1}^n$, each player starts by creating random shares in $Z_{M+1}^n$of his input vector(Step 1); namely,

$P_m$ selects M random vectors in $Z_{M+1}^n$ that add up to $b_m$, $1 \le$

$m \le M$. In Step 2, all players send to all other players the corresponding shares in their input vector. Then (Step 3), player $P_\ell$, $1 \le \ell \le M$, adds the shares that he got and arrives at his share, $s_\ell$, in the sum vector a:= $\sum_{m=1}^{M}=1$ $b_m$. Namely,

a = $\sum_{\ell=1}^{M} s_\ell$ mod (M + 1) and, furthermore, any M − 1 vectors out of $\{s_1, \ldots, s_M\}$ do not reveal any information on the sum a. In Steps 4-5, all players, apart from the last one,

send their shares to $P_1$ who adds them up to get the share s.

Now, players $P_1$ and $P_M$ hold additive shares of the sum vector a: $P_1$ has s, $P_M$ has $s_M$, and a = (s + $s_M$) mod (M + 1). It is now needed to check for each component $1 \le i \le n$ whether

Tt (b1, b2,……..bm) = 1

=2

(S(i) + S(i)) mod (M+1) < t

Whenever inequality (2) holds, we set b(i) = 0; otherwise, we set b(i) = 1.

### C. SETINC (Set Inclusion computation) Protocol

We proceed now to discuss the secure verification of inequality (2). That inequality is equivalent to the following set inclusion:

$$(s(i) + s_M(i)) \bmod (M+1) \in \{j : 0 \le j \le t-1\} \quad (3)$$

The inclusion in (3) is equivalent to

$$s(i) \in \Theta(i) := \{(j - s_M(i)) \bmod (M+1) : 0 \le j \le t-1\} \quad (4)$$

The value of s(i) is known only to P1 while the set Θ(i) is known only to $P_M$. The problem of verifying the set inclusion in Eq. (4) can be seen as a simplified version of the privacy preserving keyword search, which was solved by Freedman et. al. [13]. In the case of the OR function, t = 1, which is the case relevant for us, the set Θ(i) is of size 1, and therefore it is the problem of oblivious string comparison, a problem that was solved in e.g. [12]. However, we claim that, since M > 2, there is no need to invoke neither of the secure protocols of [13] or [12]. Indeed, as M > 2, the existence of other semi honest players can be used to verify the inclusion in Eq. (4) much more easily. This is done in SETINC protocol which we proceed to describe next.

Protocol SETINC involves three players: P1 has a vectors = (s(1), . . . , s(n)) of elements in some ground set Ω; PM, on the other hand, has a vector _ = (Θ(1), . . . , Θ(n)) of subsets of that ground set. The required output is a vector b

= (b(1), . . . , b(n)) that describes the

 

corresponding set inclusions in the following manner: b(i) = 0 if s(i) ∈ Θ(i)

and b(i) = 1 if s(i) ∉ Θ(i), 1 ≤ i ≤ n. The computation in the protocol involves a third player $P_2$. (When Protocol SETINC is called from Protocol THRESHOLD, the ground set is Ω =

$Z_{M+1}$ and the inputs s(i) and Θ(i) of the two players are as in Eq. (4), 1 ≤ i ≤ n.). The protocol starts with players $P_1$ and $P_M$ agreeing on a keyed hash function $h_K(\cdot)$ (e.g., HMAC [4]),and a corresponding secret key K (Step 1). Consequently (Steps 2-3), P1 converts his sequence of elements s = (s(1), . . . , s(n)) into a sequence of corresponding –signatures‖ s′ = (s′(1), . . . , s′(n)), where s′(i)

= $h_K$(i, s(i)) and PM does a similar conversions to the subsets that he holds. Then, in Steps 4-5, P1 sends s′ to P2, and PM sends to P2 the subsets Θ′(i), 1 ≤ i ≤ n, where the elements within each subset are randomly permuted. Finally (Steps 6-7), P2 performs the relevant inclusion verifications

on the signature values. If he finds out that for a given 1 ≤ i

≤ n, s′(i) ∈ Θ′(i), he may infer, with high probability, that s(i) ∈ Θ(i) (see more on that below), whence he sets b(i) =

0. If, on the other hand, s′(i) ∉ Θ′(i), then, with certainty, s(i) ∉ Θ(i), and thus he sets b(i) = 1.

Two comments are in order:

- If the index i had not been part of the input to the hash function (Steps 2-3), then two equal components in P1's input vector, say s(i) = s(j), would have been mapped to two equal signatures, s′(i) = s′(j). Hence, in that case player P2 would have learnt that in P1's input vector the ith and jth components are equal. To prevent such leakage of information, we include the index i in the input to the hash function.
- An event in which s′(i) ∈ Θ′(i) while s(i) ∉Θ(i) indicates a collusion; specifically, it implies that there

exist θ′ ∈ Θ(i) and θ″ ∈ Ω \ Θ(i) for which $h_K$(i, θ′) = $h_K$(i, θ″). Hash functions are designed so that the probability of such collusions is negligible, whencethe risk of a collusion can be ignored. However, it is possible for player PM to check upfront the selected random key K in order to verify that for all 1 ≤ i ≤ n, the sets Θ′(i) = {hK(i, θ) : θ ∈ Θ(i)} and Θ″(i) = {hK(i,θ) : θ ∈ Ω \ Θ(i)} are disjoint.

We refer hereinafter to the combination of Protocols THRESHOLD and SETINC as Protocol THRESHOLD-C; namely, it is Protocol THRESHOLD where the verifications of the inequalities in Steps 6-8, which are equivalent to the verification of the set inclusions in Eq. (4), are carried out by Protocol SETINC.

### D. An Improved Protocol for the Secure Computation ofAll Locally Frequent Item sets

As before, we denote by $F_s^{k-1}$ the set of all globally frequent (k − 1)-item sets, and by Ap($F_s^{k-1}$) the set of k-

item sets that the Apriori algorithm generates when applied on $F_s^{k-1}$. All players can compute the set Ap($F_s^{k-1}$) and decide on an ordering of it. (Since all item sets are subsets of A = {$a_1$, . . . , $a_L$}, they may be viewed as binary vectors in $\{0, 1\}^L$ and, as such, they may be ordered lexicographically.) Then, since the sets of

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed ( Group -I) Journal Volume 11, Iss 12, 2022

locally frequent

k-item sets, $C_s^{k,m}$ , 1 ≤ m ≤ M, are subsets of Ap($F_s^{k-1}$),they may be encoded as binary vectors of length nk := |Ap($F_s^{k-1}$)|. The binary vector that encodes the union $C_s^k$:=

$U_{m=1}^{M}$ $C_s^{k,m}$ is the OR of the vectors that encode the sets

$C_s^{k,m}$ , 1 ≤ m ≤ M. Hence, the players can compute the

union by invoking Protocol THRESHOLD-C on their binary input vectors. This approach is summarized in Protocol 4 (UNIFI).

## IV. MODULES DESCRIPTION

### A. Privacy Preserving Data Mining

Previous work in privacy preserving data mining has considered two related settings. One, in which the data owner and the data miner are two different entities, and another, in which the data is distributed among several parties who aim to jointly perform data mining on the unified corpus of data that they hold. In the first setting, the goal is to protect the data records from the data miner. Hence, the data owner aims at anonym zing the data prior to its release. The main approach in this context is to apply data perturbation. The idea is that. Computation and communication costs versus the number of transactions N the perturbed data can be used to infer general trends in the data, without revealing original record information. In the second setting, the goal is to perform data mining while protecting the data records of each of the data owners from the other data owners. This is a problem of secure multiparty computation. The usual approach here is cryptographic rather than probabilistic.

### B. Distributed Computation

We compared the performance of two secure implementations of the FDM algorithm Section In the first implementation (denoted FDM-KC), we executed the unification step using Protocol UNIFI-KC, where the commutative cipher was 1024-bit RSA in the second implementation (denoted FDM) we used our Protocol UNIFI, where the keyed-hash function was HMAC [4]. In both implementations, we implemented Step 5 of the FDM algorithm in the secure manner that was described in later. We tested the two implementations with respect to three measures: 1) Total computation time of the complete protocols (FDMKC and FDM) over all players. That measure includes the Apriori computation time, and the time to identify the globally s-frequent item sets, as described in later.2) Total computation time of the unification protocols only (UNIFI-KC and UNIFI) over all players. 3) Total message size. We ran three experiment sets, where each set tested the dependence of the above measures on a different parameter: • N — the number of transactions in the unified database,

### B. Frequent Itemsets

We describe here the solution that was proposed by Kantarcioglu and Clifton. They onsidered two possible settings. If the required output includes all globally s-frequent item sets, as well as the sizes of their supports, then the values of Δ(x) can be revealed for all x∈ . In such a case, those values may be computed using a secure summation protocol (e.g. [6]), where the private addend of Pm is suppm(x) − sNm. The more interesting setting, however, is the

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed ( Group -I) Journal Volume 11, Iss 12, 2022

one where the support sizes are not part of the required output. Click on unify algorithm button to apply the unify algorithm onto the frequent dataset:

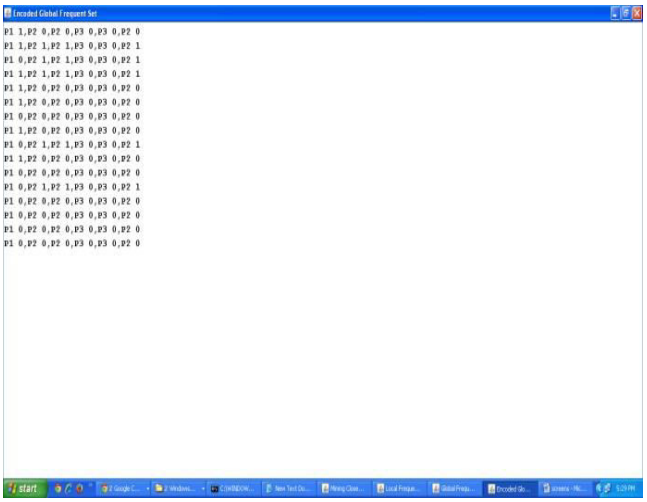


**Fig.2.**

### C. Association Rules

Once the set Fs of all s-frequent itemsets is found, we may proceed to look for all (s, c)-association rules (rules with support at least sN and confidence at least c), as described in [18]. For X, Y ∈ Fs, where X ∩ Y = ∅, the corresponding association rule X ⇒ Y has confidence at least c if and only if supp(X ∪ Y )/supp(X) ≥ c, or, equivalently, $C_{X,Y} := M\Sigma_{m=1}(\text{suppm}(X \cup Y) - c \cdot \text{suppm}(X)) \geq 0$ . (10) Since $|C_{X,Y}| \leq N$, then by taking q = 2N+1, the players can verify inequality (10), in parallel, for all candidate association rules, as described.

## V. CONCLUSION

We proposed a protocol for secure association rule mining in horizontally distributed databases that pick up appreciably upon the existing foremost protocol [18] in terms of privacy and efficiency. One of the key constituents in our designed protocol is a novel secure multi-party protocol for computing the union of private subsets that each of the interacting players hold. Another constituent is a protocol that tests the inclusion of an element held by one player in a subset held by another. Those protocols make use of the fact that the underlying trouble is of interest only when the number of players is greater than two. One research problem that this study recommend was described to devise an efficient protocol for inequality verifications that uses the existence of a semi-honest third party. Such a protocol might enable to further improve upon the communication and computational costs of the second and third stages of the protocol of [18], as described in Sections 3 and 4. Other research problems that this study suggests is the implementation of the techniques presented here to the problem of distributed association rule mining in the vertical setting, the problem of mining generalized association rules, and the problem of subgroup discovery in horizontally partitioned data [16].

## VI. REFERENCES

[1] Tamir tassa, ―Secure Mining of Association Rules in Horizontally Distributed Databases‖, IEEE transactions on knowledge and data engineering, 2013.

[2] J. Vaidya and C. Clifton, ―Privacy preserving association rule mining in vertically partitioned data,‖ in The Eighth ACM SIGKDD InternationalConference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, July 23-26 2002, pp. 639–644.

[3] D. Beaver, S. Micali, and P. Rogaway. The round complexity of secure protocols. In STOC, pages 503–513, 1990.

[4] M. Bellare, R. Canetti, and H.

Krawczyk. Keying hash functions for message authentication. In Crypto, pages 1– 15, 1996.

[5] M.Kantarcioglu and C. Clifton., ―Privacy-preserving distributed mining of association rules on horizontally partitioned data‖, IEEE Transactions on Knowledge and Data Engineering, 16:1026–1037,2004.

[6] A. Ben-David, N. Nisan, and B. Pinkas. FairplayMP - A system for secure multi-party computation. In CCS, pages 257–266, 2008.

[7] J.C. Benaloh. Secret sharing homomorphisms: Keeping shares of a secret secret. In Crypto, pages 251–260, 1986.

[8] R.Agrawal and R. Srikant.,―Privacy-preserving data mining‖, SIGMOD Conference, pages 439–450, 2000.

[9] A.V. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, ―Privacy preserving mining of association rules‖, In KDD, pages 217–228, 2002.

[10] M. Kantarcioglu, R. Nix, and J. Vaidya,―An efficient approximate protocol for privacy-preserving association rulemining‖, In PAKDD, pages 515–524, 2009.

[11] M. Freedman, Y. Ishai, B. Pinkas, and O. Reingold.,
―Keyword search and oblivious pseudorandom functions‖, In TCC, pages 303–324, 2005.

[12] X. Lin, C. Clifton, and M.Y. Zhu. Privacy-preserving clustering with distributed EM mixture modeling. Knowl. Inf. Syst., 8:68–81, 2005.

[13] R. Fagin, M. Naor, and P. Winkler. Comparing Information WithoutLeaking It. Communications of the ACM, 39:77–85, 1996.

M. Freedman, Y. Ishai, B. Pinkas, and O. Reingold. Keyword search and oblivious pseudorandom functions. In TCC, pages 303–324, 2005.

[13] Y. Lindell and B. Pinkas. Privacy preserving data mining. In Crypto, pages 36–54, 2000.

[14] J.S. Park, M.S. Chen, and P.S. Yu. An effective hash based algorithm for mining association rules. In SIGMOD Conference, pages 175–186, 1995.

[15] S.C. Pohlig and M.E. Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance. IEEE ransactions on Information Theory, 24:106–110, 1978.

[16] R.L. Rivest, A. Shamir, and L.M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21(2):120–126, 1978.

[17] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In STOC, pages 218–229, 1987.

[18] A. Schuster, R. Wolff, and B. Gilburd. Privacy- preserving association rule mining in large-scale distributed systems. In CCGRID, pages 411– 418, 2004.

[19] R. Srikant and R. Agrawal. Mining generalized association rules. In VLDB, pages 407–419, 1995.

T. Tassa and D. Cohen. Anonymization of centralized and distributed social networks by sequential clustering. IEEE Transactions on knowledge and Data Engineering, 2012.

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed ( Group -I) Journal Volume 11, Iss 12, 2022