iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** IV **Month of publication:** April 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

A Novel Secure Protocol for Mining in Horizontally Distributed Databases

B. Erubabu[1], K.N. Brahmaji Rao[2]
[1]PG Student, Baba Institute of Technology and Sciences, Vishakapatnam, A.P.INDIA
[2]Associate Professor, Baba Institute of Technology and Sciences, Vishakapatnam, A.P.INDIA

***Abstract*—For Secure mining of association rules in horizontally distributed databases we propose aprotocol based on the Fast Distributed Mining (FDM) algorithm which is an unsecured distributed version of the Apriori algorithm. This protocol is of two novel secure multi-party algorithms—one that computes the union of private subsets that each of the interacting players hold, and another that tests the inclusion of an element held by one player in a subset held by another. The proposed protocol enhances privacy with respect to the protocol of Kantarcioglu and Clifton[18]. In terms of communication rounds, communication cost and computational cost,it is simpler and is significantly more efficient.**
***Index Terms*—association rules, distributed databases, Fast Distributed Mining, Apriori Algorithm**

I. INTRODUCTION

In the problem of secure mining of association rules in horizontally partitioned databases there are several sites(or players) that hold homogeneous databases. 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. In this context we would like to protect the information in individual transactions in the different databases and also more global information in which the goal defines the problem of secure multiparty computation. In such problems, there are M players that hold private inputs, $x_1,x_2,\ldots x_m$, and they wish to securely compute $y=f(x_1,x_2,\ldots x_m)$for some public function f. The players can run on their own if a protocol is devised and in order to arrive at the required output y.If no player can learn from his view of the protocol such a protocol is considered perfectly secure. The computation is carried out by trusted third party.Yao [32] was the first to propose a generic solution for this problem in the case of two players. The other generic solutions, for the multi-party case, were later proposed in [3], [5], [15]. Here in the existing problem, the partial databases are the inputs, and the list of association rules are the output.These rules hold in the unified database with support and confidence no smaller than the given thresholds s and c. These mentioned generic solutions rely upon a description of the function f as a Boolean circuit, they can be applied only to small inputs and functions which are realizable by simple circuits. Since ours is of more complex settingsdifferent methods are required for this computation. In such cases, some relaxations of the notion ofperfect security might be inevitable when looking for practical protocols, provided that the excess information is deemed benign [18], [28], [29], [31], [34].
In the problem in [18]Kantarcioglu and Clifton devised a protocol for its solution. The main part of the protocol is a sub-protocol for the secure computation of the union of private subsets that are held by the different players which is the most costly part of the protocol. The cryptographic primitives such as commutative encryption, oblivious transfer, and hash functions are in which the implementation relies upon. This is also the only part in the protocol in which the players may extract from their view of the protocol information on other databases, beyond what is implied by the final output and their own input. The leakage of information renders the protocol not perfectly secure, the perimeter of the excess information is explicitly bounded in [18] and it is argued there that such information leakage is innocuous, whence acceptable from a practical point of view. The proposal is an alternative protocol for the secure computation of the union of private subsets. In terms of simplicity and efficiency as well as privacythe proposed protocol improves upon that in [18]. The proposed protocol does not depend on commutative encryption and oblivious transfer.It leaks excess information only to a small number (three) of possible coalitions, unlike the protocol of [18] that discloses information also to some single players. In addition, we claim that the excess information that our protocol may leak is less sensitive than the excess information leaked by the protocol of [18]. We propose a protocol that computes a parameterized family of functionsi.e threshold functions, in which the two extreme cases correspond to the problems of computing the union and intersection of private subsets. We solve the problem of the set inclusion problem namely, the problem where Alice holds a private subset of some ground set, and Bob holds an element in the ground set, and they wish to deter-mine whether Bob's element is within Alice's subset, without revealing to either of them information about the other party's input beyond the above described inclusion.

A. The Fast Distributed Mining Algorithm

The proposed protocol and the protocol of [18], are based on the Fast Distributed Mining (FDM) algorithm of Cheung et al. [8], which is considered as an unsecured distributed version of the Apriori algorithm. Any s-frequent item set must be also locally s-frequent in at least one of the sites is the main idea behind the Apriori algorithm.In order to find all globally s-frequent item sets, each playerreveals his locally s-frequent item sets and then the players check each of them to see if they are s-frequent also globally. The FDM algorithm proceeds as follows:

1) Initialization: It is assumed that the players have already jointly calculated F_s^{k-1}. The goal is to proceed and calculate F_s^k.
2) Candidate Sets Generation: Each player P_m computes the set of all (k–1)-item sets that are locally frequent in his site and also globally frequent; namely, P_m computes the set $F_s^{k-1,m} \cap F_s^{k-1}$. He then applies on that set the Apriori algorithm in order to generate the set $B_s^{k,m}$ of candidate k-item sets.
3) Local Pruning: For each $X \varepsilon B_s^{k,m}$, P_m computes $supp_{m(}X)$. He then retains only those item sets thatare locally s-frequent. We denote this collection of item sets by $C_s^{k,m}$.
4) Unifying the candidate item sets: Each player broadcasts his $C_s^{k,m}$ .and then all players compute $C_s^k := U^M_{m=1} C_s^{k,m}$
5) Computing local supports. All players compute the local supports of all item sets in C_s^k.
6) Broadcast mining results: Each player broadcasts the local supports that he computed. From that, everyone can compute the global support of every item set in C_s^k. Finally, F_s^k is the subset of C_s^k that consists of all globally s-frequent k-item sets.

In the first iteration, when k = 1, the set $C_s^{1,m}$ that the m^{th} player computes (Steps 2-3) is just $F_s^{1,m}$, namely, the set of single items that are s-frequent in D_m. The complete FDM algorithm starts by finding all single items that are globally s-frequent. It then proceeds to find all 2-item sets that are globally s-frequent, until it finds the longest globally s-frequent item sets. If the length of such item sets is K, then in the $(k+1)^{th}$ iteration of the FDM it will find no (k+1)-item sets that are globally s-frequent, in which case it terminates.

The privacy violation in FDM algorithm occurs in two stages: In Step 4, where the players broadcast the item sets that are locally frequent in their private databases, and in Step 6, where they broadcast the sizes of the local supports of candidate item sets. Kantarcioglu and Clifton [18] proposed secure implementations of those two steps. The modification is regard to the secure implementation of Step 4, which is the more costly stage of the protocol, and the one in which the protocol of [18] leaks excess information.

Similar to [18], we assume that the players are semi-honest, namely, they follow the protocol but try to extract as much information as possible from their own view [17], [26], [34].

II. PROPOSED FULLY SECURE PROTOCOL

The players may dispense the local pruning and union computation in the FDM algorithm (Steps 2-4) and, instead, test all candidate item sets in $Ap\eth F_s^{k-1}Þ$ to see which of them are globally s-frequent. Such a protocol is fully secure, as it reveals only the set of globally s-frequent item sets but no further information about the partial databases. However, as discussed in [18], the protocol would be much more costly since it requires each player to compute the local support of $|Ap(F_s^{k-1})|$ item sets instead of only $|C_s^k|$ item sets.In addition, the players will have to execute the secure comparison protocol of [32] to verify inequality for $|Ap(F_s^{k-1})|$ rather than only $|C_s^k|$ item sets. Both types of added operations are very costly: the time to compute the support size depends linearly on the size of the database, while the secure comparison pro-tocol entails a costly oblivious transfer sub-protocol. Since, as shown in [9], $|Ap(F_s^{k-1})$ is much larger than $|C_s^k|$, the added computing time in such a protocol is expected to dominate the cost of the secure computation of the union of all locally s-frequent item sets. Hence, the enhanced security offered by such a protocol is accompanied by increased implementation costs.

III. EXPERIMENTAL EVALUATION

In this chapter we describe the synthetic database which we used for our experimentation and how the database was split horizontally into partial a database which describes the experiments we conducted. The results are also given in this section.

A. Synthetic Database Generation

We use the databases in our experimental evaluation are synthetic databases. These databases are generated using the same

techniques that were introduced in [1].These were also used in subsequent studies such as [8], [18], [23]. The parameter values that were used in generating the synthetic database which were used in Table 1. The reader is referred to [8], [18], [23] for a description of the synthetic generation method and the meaning of each of those parameters.

TABLE 1: Parameters for Generating the Synthetic Database

Parameter	Interpretation	Value
N	Number of transactions in the whole database	500,000
L	Number of items	1000
A_t	Transaction average size	10
A_f	Average size of maximal potentially large itemsets	4
N_f	Number of maximal potentially large itemsets	2000
CS	Clustering size	5
PS	Pool size	60
Cor	Correlation level	0.5
MF	Multiplying factor	1800

B. Distributing the Database

Given a generated synthetic database D of N transactions and a number of players M, we create an artificial split of D into M partial databases, D_m, $1 \le m \le M$, in the following manner: For each $1 \le m \le M$ we draw a random number $w_m$from a normal distribution with mean 1 and variance0.1, where numbers outside the interval [0.1,1.9] are ignored. Then, we normalize those numbers so that$\sum_{m=1}^{M} w_m = 1$.We randomly split D into m partial databases of expected sizes of $w_m N$, $1 \le m \le M$, as follows: Each transaction $t \varepsilon D$ is assigned at random to one of the partial databases, so that $Pr(t \varepsilon D_m) = w_m$, $1 \le m \le M$.

C. Experimental Setup

By comparing the performance of two secure implementations of the FDM algorithm i.ein the first implementation, we executed the unification step (Step 4 in FDM) using Protocol UNIFI-KC, where the commutative cipher was 1,024-bit RSA [25]; 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 Section 3. The two implementations with respect to three measures were tested.

1) Total computation time of the complete protocols (FDM-KC and FDM) over all players. That measure includes the time to identify the globally s-frequent item sets and the Apriori computation time.
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 i.eN—the number of transactions in the unifieddatabase,M—the number of players, ands—the threshold support size. In our basic configuration, we took N=500,000, M=10, and s=0.1. In the first experiment set, we kept M and s fixed and tested several values of N. In the second experiment set, we kept N and s fixed and varied M. In the third set, we kept N and M fixed and varied s. The results in each of those experiment sets are shown in the next section. All experiments were implemented in C# (.net 4) and were executed on an Intel(R) Core(TM)i7-2620M personal computer with a 2.7 GHz CPU, 8 GB of RAM, and the 64-bit Windows 7 Professional SP1 operating system.

D. Experimental Results

The values of the three measures that were listed in the section above as a function of N are shown in the figure 1. In all of those experiments, the value of M and s remained unchanged—M=10 and s=0.1. Fig. 2 shows the values of the three measures as a function of M; here, N¼500,000 and s=0.1. Fig. 3 shows the values of the three measures as a function of s; here,N = 500,000 and M = 10. The first set of experiments shows that N has little effect on the runtime of the unification protocols, UNIFI-KC and UNIFI, nor on the bit communication cost. However, since the time to identify the globally s-frequent item sets (see Section 3) does grow linearly with N, and that procedure is carried out in the same manner in FDM-KC and FDM, the advantage of Protocol FDM over FDM-KC in terms of runtime decreases with N. While for N=100,000, Protocol FDM is 22 times faster than Protocol FDM-KC, for N = 500,000 it is five times faster.

The second set of experiments shows how the computation and communication costs increase with M. The improvement factor in

the bit communication cost, as offered by Protocol UNIFI with respect to Protocol UNIFI-KC, is in accord with our analysis. Finally, the third set of experiments shows that higher support thresholds entail smaller computation and communication costs since the number of frequent item sets decreases.

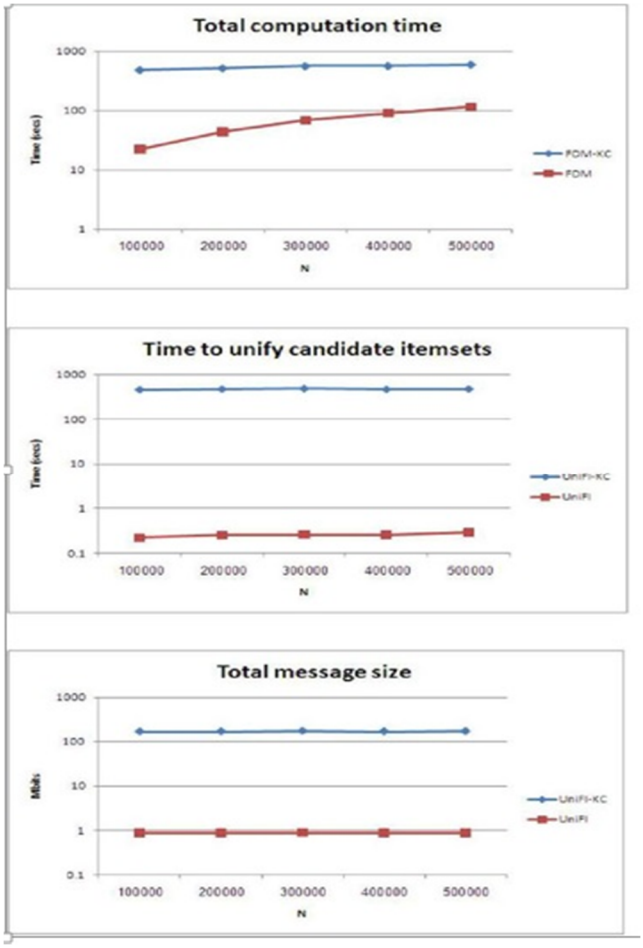


Fig. 1. Computation and communication costs versus the number of transactions N

IV. CONCLUSION

The protocol is proposed for secure mining of association rules in horizontally distributed databases that improves significantly upon the current leading protocol [18] in terms of privacy and efficiency. In our proposed protocol the main ingredient is a novel secure multi-partyprotocol for computing the union (or intersection) of private subsets that each of the interacting players hold. The other ingredient is a protocol that tests the inclusion of an element held by one player in a subset held by another. Those protocols exploit the fact that the underlying problem is of interest only when the number of players is greater than two.

This study suggests to devise an efficient proto-col for inequality verifications that uses the existence of a semi-honest third party. This further improve upon the communication and computational costs of the second and third stages of the protocol of [18].This study also suggests is the implementation of the techniques presented here to the problem of distributed association rule mining in the vertical setting [31], [33], the problem of mining generalized association rules [27], and the problem of subgroup discovery in horizontally partitioned data [16].

Volume 3 Issue IV, April 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

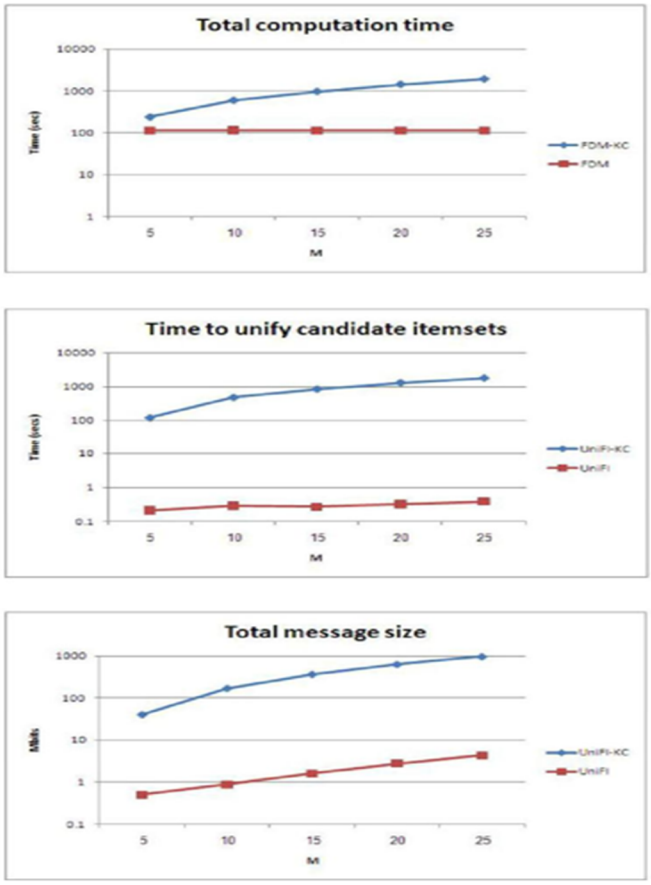


Fig. 2. Computation and communication costs versus the number of players M

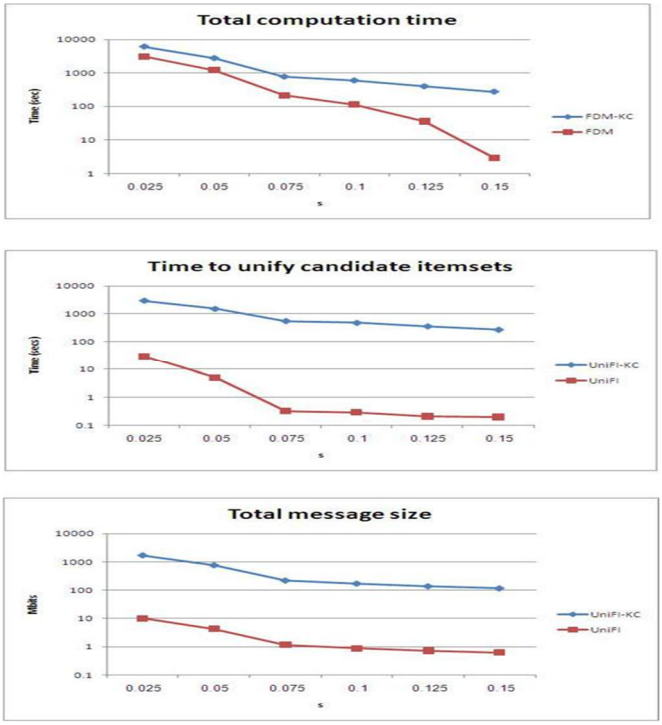
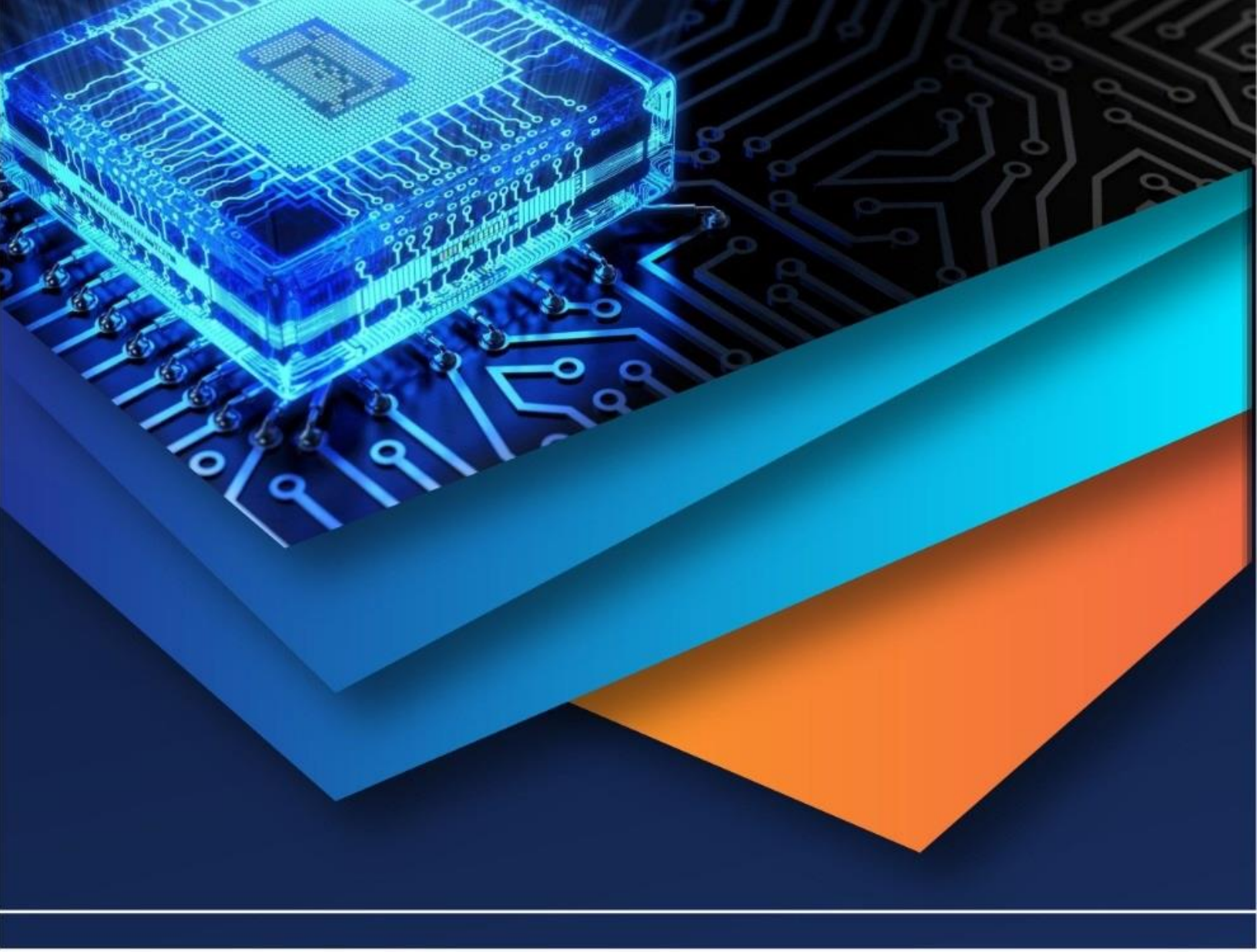


Fig. 3. Computation and communication costs versus the support threshold

REFERENCES

[1] Tamir Tassa,"Secure Mining of Association Rules in Horizontally Distributed Databases" IEEE Trans. Knowledge and Data Eng., vol. 26, no. 4, pp. 970-983, April. 2014.
[2] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Associa-tion Rules in Large Databases," Proc 20th Int'l Conf. Very LargeData Bases (VLDB), pp. 487-499, 1994.
[3] R. Agrawal and R. Srikant, "Privacy-Preserving Data Mining," Proc. ACM SIGMOD Conf., pp. 439-450, 2000
[4] D. Beaver, S. Micali, and P. Rogaway, "The Round Complexity of Secure Protocols," Proc. 22nd Ann. ACM Symp. Theory of Computing(STOC), pp. 503-513, 1990.
[5] M. Bellare, R. Canetti, and H. Krawczyk, "Keying Hash Functions for Message Authentication," Proc. 16th Ann. Int'l Cryptology Conf.Advances in Cryptology (Crypto), pp. 1-15, 1996.
[6] A. Ben-David, N. Nisan, and B. Pinkas, "FairplayMP - A System for Secure Multi-Party Computation," Proc. 15th ACM Conf. Com-puters. and Comm. Security (CCS), pp. 257-266, 2008.
[7] J.C. Benaloh, "Secret Sharing Homomorphisms: Keeping Shares of Secret Secret," Proc. Advances in Cryptology (Crypto), pp. 251-260, 1986.
[8] J. Brickell and V. Shmatikov, "Privacy-Preserving Graph Algo-rithms in the Semi-Honest Model," Proc. 11th Int'l Conf. Theory andApplication of Cryptology and Information Security (ASIACRYPT), pp.236-252, 2005.
[9] D.W.L. Cheung, J. Han, V.T.Y. Ng, A.W.C. Fu, and Y. Fu, "A Fast Distributed Algorithm for Mining Association Rules," Proc. FourthInt'l Conf. Parallel and Distributed Information Systems (PDIS), pp.31-42, 1996.
[10] D.W.L Cheung, V.T.Y. Ng, A.W.C. Fu, and Y. Fu, "Efficient Min-ing of Association Rules in Distributed Databases," IEEE Trans.Knowledge and Data Eng., vol. 8, no. 6, Dec. 1996.
[11] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Trans. Information Theory, vol. IT-31, no. 4, July 1985
[12] A.V. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy Preserving Mining of Association Rules," Proc. Eighth ACMSIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp.217-228, 2002.
[13] R. Fagin, M. Naor, and P. Winkler, "Comparing Information with-out Leaking It," Comm. ACM, vol. 39, pp. 77-85, 1996.
[14] M. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword Search and Oblivious Pseudorandom Functions," Proc. Second Int'lConf. Theory of Cryptography (TCC), pp. 303-324, 2005.
[15] M.J. Freedman, K. Nissim, and B. Pinkas, "Efficient Private Match-ing and Set Intersection," Proc. Int'l Conf. Theory and Applications ofCryptographic Techniques (EUROCRYPT), pp. 1-19, 2004.
[16] O. Goldreich, S. Micali, and A. Wigderson, "How to Play Any Mental Game or a Completeness Theorem for Protocols with Hon-est Majority," Proc. 19th Ann. ACM Symp. Theory of Computing(STOC), pp. 218-229, 1987.
[17] H. Grosskreutz, B. Lemmen, and S. Ruping,€ "Secure Distributed Subgroup Discovery in Horizontally Partitioned Data," Trans.Data Privacy, vol. 4, no. 3, pp. 147-165, 2011.
[18] W. Jiang and C. Clifton, "A Secure Distributed Framework for Achieving k-Anonymity," The VLDB J., vol. 15, pp. 316-333, 2006.
[19] M. Kantarcioglu and C. Clifton, "Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 9, pp. 1026-1037,Sept. 2004.
[20] M. Kantarcioglu, R. Nix, and J. Vaidya, "An Efficient Approxi-mate Protocol for Privacy-Preserving Association Rule Mining," Proc. 13th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD), pp. 515-524, 2009.
[21] L. Kissner and D.X. Song, "Privacy-Preserving Set Operations," Proc. 25th Ann. Int'l Cryptology Conf. (CRYPTO), pp. 241-257, 2005.
[22] X. Lin, C. Clifton, and M.Y. Zhu, "Privacy-Preserving Clustering with Distributed EM Mixture Modeling," Knowledge and Informa-tion Systems, vol. 8, pp. 68-81, 2005.
[23] Y. Lindell and B. Pinkas, "Privacy Preserving Data Mining," Proc.Crypto, pp. 36-54, 2000.
[24] J.S. Park, M.S. Chen, and P.S. Yu, "An Effective Hash Based Algo-rithm for Mining Association Rules," Proc. ACM SIGMOD Conf., pp. 175-186, 1995. S.C. Pohlig and M.E. Hellman, "An Improved Algorithm for Com-putting
[25] Logarithms over gfðpÞ and Its Cryptographic Signifi-cance," IEEE Trans. Information Theory, vol. IT-24, no. 1, pp. 106-110, Jan. 1978.
[26] R.L. Rivest, A. Shamir, and L.M. Adleman, "A Method for Obtain-ing Digital Signatures and Public-Key Cryptosystems," Comm.ACM, vol. 21, no. 2, pp. 120-126, 1978.
[27] A. Schuster, R. Wolff, and B. Gilburd, "Privacy-Preserving Associ-ation Rule Mining in Large-Scale Distributed Systems," Proc. IEEEInt'l Symp. Cluster Computing and the Grid (CCGRID), pp. 411-418,2004.
[28] R. Srikant and R. Agrawal, "Mining Generalized Association Rules," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 407-419, 1995.
[29] T. Tassa and D. Cohen, "Anonymization of Centralized and Dis-tributed Social Networks by Sequential Clustering," IEEE Trans.Knowledge and Data Eng., vol. 25, no. 2, pp. 311-324, Feb. 2013.
[30] T. Tassa and E. Gudes, "Secure Distributed Computation of Ano-nymized Views of Shared Databases," Trans. Database Systems, vol. 37, article 11, 2012.
[31] T. Tassa, A. Jarrous, and J. Ben-Ya'akov, "Oblivious Evaluation of Multivariate Polynomials," J. Mathematical Cryptology, vol. 7, pp. 1-29, 2013.
[32] J. Vaidya and C. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," Proc. Eighth ACM SIGKDDInt'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 639-644, 2002.
[33] A.C. Yao, "Protocols for Secure Computation," Proc. 23rd Ann.Symp. Foundations of Computer Science (FOCS), pp. 160-164, 1982.
[34] J. Zhan, S. Matwin, and L. Chang, "Privacy Preserving Collabora-tive Association Rule Mining," Proc. 19th Ann. IFIP WG 11.3 Work-ing Conf. Data and Applications Security, pp. 153-165, 2005.
[35] S. Zhong, Z. Yang, and R.N. Wright, "Privacy-Enhancing k-Ano-nymization of Customer Data," Proc. ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS), pp. 139-147,2005.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY