Implementation of Attribute Based Symmetric Encryption through Vertically Partitioned Data in PPDM

Devendrasinh Vashi, H B Bhadka, Kuntal Patel, Sanjay Garg

Abstract*: The aim of this research article is to implement the symmetric encryption techniques on the vertically partitioned database table. In privacy-preserving data mining mainly sensitive attributes to be protected so that privacy will maintain during data mining. In this proposed algorithm the datasets partitioned in the three different sets. Then for each table for the selected attributes only one symmetrical encryption as well as different symmetric encryption is implemented. Initial data size, encryption execution time and data size after encryption are observed for each file of different data size. In result and analysis examined the performance of execution time and memory occupied after each encryption techniques is discussed and found that hybrid algorithm of using different symmetric encryption for each partitioned table is good as compared to implementing only one encryption on each partitioned table. This algorithm is mainly useful to provide privacy in PPDM in case of distributed data base of health care organizations.

Keywords* : privacy preserving, vertically partitioned data, Data Mining, encryption.

I. INTRODUCTION

In the current era of industry, the majority of the organizations do data mining to find some useful patterns which can be used for business. In same context in insurance agencies one of the industry which can do data mining on the personal and medical data to find some useful information which can be helpful for them to launch new insurance policy in the market. So the question is how to protect the privacy of human? People are ready to provide data if their personal data is protected during the data mining process as some data are sensitive [6]. One of the probable technique is encryption. But the major questions is how to use encryption so that hackers can not easily decrypt in the information.

One of the good solutions of privacy preserving [3] is to use encryption with horizontal and vertical partition [12] data. In this article propose algorithm is implemented based on attribute encryption in vertically partitioned data. Only symmetric encryption techniques ware used.

Revised Manuscript Received on October 15, 2019
***** Correspondence Author
Devendrasinh Vashi*, Computer Science Department, Nirma University, Ahmedabad, India. Email: devendra.vashi@gmail.com
Dr. H B Bhadka, Faculty of Computer Science, C U Shah University, Wadhwan, India, harshad.bhadka@yahoo.com
Dr. Kuntal Patel, School of Computer Studies, Ahmedabad University, Ahmedabad, India, kuntal.patel@ahduni.edu.in
Dr. Sanjay Garg, Computer Science Department, Nirma University, Ahmedabad, India. Email: devendra.vashi@gmail.com

II. ALGORITHM

A. Algorithm

The below cryptography based [9][13] hybrid algorithm [7] is implemented with symmetric encryption on a vertically partitioned database for different sensitive attributes.

Step-1:Start
Step-2:Create the data set in excel sheet
Step-3:Upload the data set in the system to convert it into SQL server data set
Step-4:Partitioned the SQL sever data set vertically into three data tables through attribute selection for every three tables (high, average and low sensitive).
Step-5:Select the attribute from each data set to encrypt
Step-6:Implement the symmetric encryption for the selected attribute on table-1
Step-7:Implement the asymmetric encryption for the selected attribute on table-2
Step-8:Implement the asymmetric encryption for the selected attribute on table-3
Step-9:Note down initial table size, encryption execution time and table size after encryption for Step-7 to Step-8.
Step-10:Merged all three encrypted tables then download the data set into the excel sheet to provide the third party for data mining.
Step-11: Stop

B. Database attributes

Following 23 attributes were considered during implementation of the proposed algorithm:

Timestamp	Nationality
Name	Education
Gender	Marital Status
Date of birth	Address 1(Street) [Home]
Blood Group	Address 2(City)[Home]
Disability	Address 3(State) [Home]
Pan card Number	Address 1(Street)[Work Space]
Diseases	Address 2(City)[Work Space]
Medication & allergies	Address 3(State)[Work Space]
Mobile No.	Doctor Name
E-mail ID	Date & Time of visit
Salary	

III. VERTICALLY PARTITIONED DATABASE

For vertically partitioning [11] of the database, excel sheet was converted first in the SQL server database. The attributes of the database table are displayed with check-boxes as per the FIg. 1. Then Name, Gender, Date of Birth, Mobile Number, Email id, and pan card is selected for the first table which is considered as highly sensitive attributes. Then blood group, disability, nationality, education, marital status, salary, disease, medication and allergies, doctor name and date-time of hospital visits are selected for creating the second table that is average or normal sensitive attributes. Then time-stamp, street, city, and state of home town and workplace respectively for creating the third table that is low sensitive attributes.

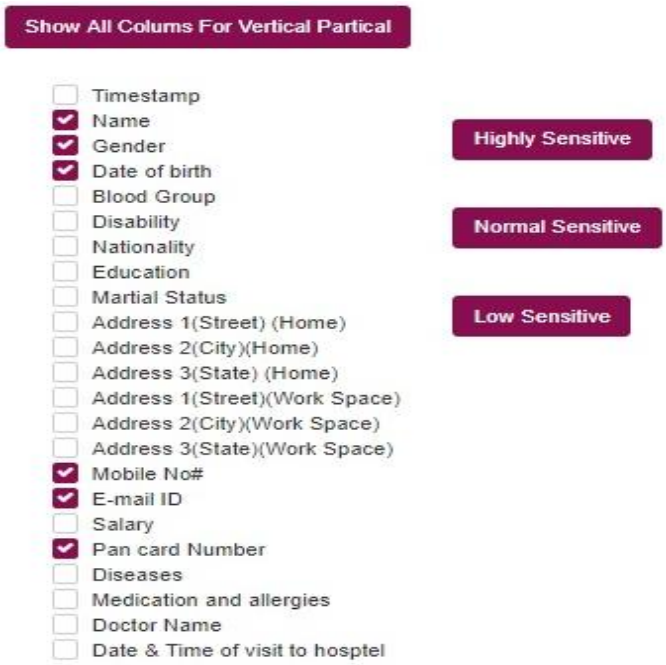

Fig. 1: Design of partitioning of table vertically

IV. SYMMETRIC CRPTOGRAPHIC TECHNIQUES

In symmetric encryption only one secret key is used. The secret key which is used for encryption, same secret key will be used for decryption. Based on survey or research article as compare to symmetric encryption asymmetrical encryption is costly to implement.

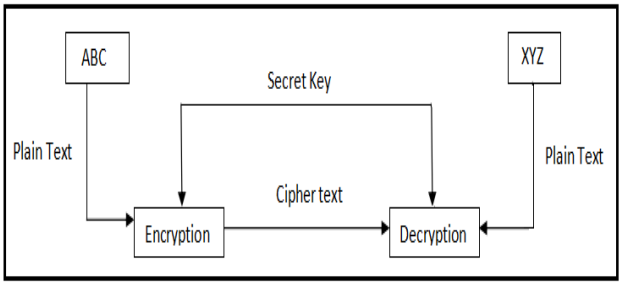

Fig. 2: Symmetric encryption

V. EXPERIMENT AND RESULT ANALYSIS

During the experiment total 10 excel sheet was prepared of 100 kb, 200 kb, 300, kb, 400 kb, 500 kb, 600 kb, 700 kb, 800 kb, 900 kb, and 1000 kb. Each sheet contains the data of different kinds of sensitive attributes. Initially, 100 kb file is used for the vertically partitioned [14] data into three tables of highly sensitive, average sensitive and low sensitive attributes as per Fig. 3. The selection of attributes was based on the survey from different age people. Then for every three tables, AES [1] encryption technique implemented on the selected attributes and observed the encryption time and memory occupied before and after encryption. Then gradually file size is increased every time and observed the encryption time and memory. The same process is repeated for DES [2], Rijndael [4], and RC2 respectively as per table-I to table-IV.

At the end as per the proposed hybrid algorithm for table1 DES encryption, for table2 Rijndael and for table3 RC2 encryption techniques is implemented and observed the encryption time and memory before and after execution time for 10 different size database as per table-V.

Fig 4 to Fig 7 depict that execution time increase with respect to file size increase. And Fig.8. shows the results of Performance of AES, DES, Rijndael & RC2 technique based on execution time. Fig.9. shows the results of Performance of AES, DES, Rijndael & RC2 [8] technique based on memory occupied after encryption which is almost symmetrical and found almost similar.

Fig.10. depicts the performance analysis using Rijndael technique based on execution time as per the proposed algorithm. Same was compared with the performance of AES, DES, Rijndael & RC2 in Fig.11 and found that the execution time is almost the same.

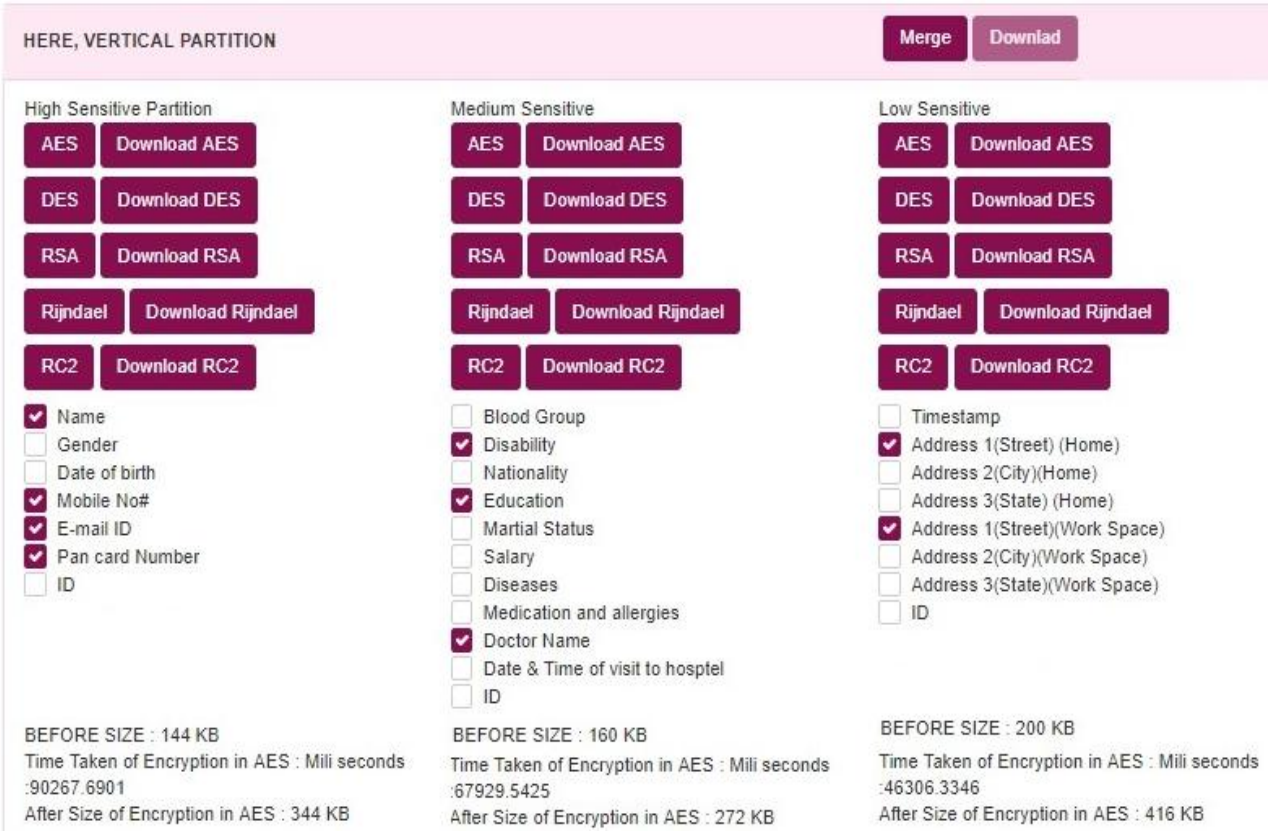

Fig. 3: Design of attribute selection for the symmetric encryption for each vertically partitioned table

Table- I: Execution time of different size dataset with AES technique

Partitioned table	Table1			Table2			Table3			Total size of all 3 table (kb)	Total memory occupied after encryption	Total execution time (milliseconds)
File name with data size (kb).	*Table-1 size before encryption (kb)*	*Execution time to table-1 (milliseconds)*	*Table-1 size after encryption (kb)*	*Table-2 size before encryption (kb))*	*Execution time to table-2* (milliseconds)	*Table-2 size after encryption (kb)*	*Table-3 size before encryption (kb)*	*Execution time to table-3* (milliseconds)	*Table-3 size after encryption (kb)*			
Sensitive data 1.xlsx (100)	144	90267.6901	344	160	67929.5425	272	200	46306.3346	416	504	1032	204503.5672
Sensitive data 2.xlsx (200)	280	183588.2397	704	304	129212.654	552	344	88987.4758	832	928	2088	401788.3695
Sensitive data 3.xlsx (300)	336	392387.1814	1072	368	333743.2281	864	408	213627.2396	1056	1112	2992	939757.6491
Sensitive data 4.xlsx (400)	544	463814.817	1424	592	319268.6467	1096	784	231906.4461	1752	1920	4272	1014989.91
Sensitive data 5.xlsx (500)	672	546453.8412	1808	736	393989.8274	1376	984	267975.3904	2176	2392	5360	1208419.059
Sensitive data 6.xlsx (600)	792	690905.2735	2144	872	523886.6581	1640	1160	345568.2388	2600	2824	6384	1560360.17
Sensitive data 7.xlsx (700)	912	949316.1172	2480	1000	698775.673	1928	1136	444246.7569	2760	3048	7168	2092338.547
Sensitive data 8.xlsx (800)	1024	1179222.182	2848	1120	811727.5981	2200	1496	535742.5915	3336	3640	8384	2526692.372
Sensitive data 9.xlsx (900)	1160	1199223.545	3192	1264	878406.5639	2456	1712	581185.5489	3808	4136	9456	2658815.658
Sensitive data 10.xlsx	1280	1452810.135	3520	1408	971090.5453	2728	1888	785263.2031	4208	4576	10456	3209163.883

Table- II: Execution time of different size dataset with DES technique

Partitioned table	Table1			Table2			Table3			Total size of all 3 table (kb)	Total memory occupied after encryption	Total execution time (milliseconds)
File name with data size (kb).	*Table-1 size before encryption (kb)*	*Execution time to table-1 (milliseconds)*	*Table-1 size after encryption (kb)*	*Table-2 size before encryption (kb)*	*Execution time to table-2* (milliseconds)	*Table-2 size after encryption (kb)*	*Table-3 size before encryption (kb)*	*Execution time to table-3* (milliseconds)	*Table-3 size after encryption (kb)*			
Sensitive data 1.xlsx (100)	144	7419.4684	304	160	4419.9175	248	200	3459.4784	432	504	984	15298.8643
Sensitive data 2.xlsx (200)	280	23401.6023	624	304	14685.6821	496	400	13435.2242	856	984	1976	51522.5086
Sensitive data 3.xlsx (300)	336	85790.592	872	368	61703.1792	712	472	39305.7805	1056	1176	2640	186799.5517
Sensitive data 4.xlsx (400)	544	100706.7793	1272	592	65651.694	984	784	46001.0648	1680	1920	3936	212359.5381
Sensitive data 5.xlsx (500)	672	141062.8084	1576	736	89575.4847	1224	976	71319.6854	2088	2384	4888	301957.9785
Sensitive data 6.xlsx (600)	792	191739.577	1896	872	119723.7523	1464	1160	88549.5095	2480	2824	5840	400012.8388
Sensitive data 7.xlsx (700)	912	236152.7501	2184	1000	142879.6564	1704	1328	102811.9172	2848	3240	6736	481844.3237
Sensitive data 8.xlsx (800)	1024	309889.2111	2472	1120	189263.6741	1928	1496	151581.7154	3192	3640	7592	650734.6006
Sensitive data 9.xlsx (900)	1160	362130.6871	2808	1272	218216.2036	2168	1712	166186.7506	3656	4144	8632	746533.6413
Sensitive data 10.xlsx (1000)	1280	472875.2347	3080	1408	278923.602	2400	1888	203300.296	4016	4576	9496	955099.1327

Table- III: Execution time of different size dataset with Rijndael technique

Partitioned table	Table1			Table2			Table3			Total size of all 3 table (kb)	Total memory occupied after encryption	Total execution time (milliseconds)
File name with data size (kb).	*Table-1 size before encryption (kb)*	*Execution time to table-1 (milliseconds)*	*Table-1 size after encryption (kb)*	*Table-2 size before encryption (kb)*	*Execution time to table-2* (milliseconds)	*Table-2 size after encryption (kb)*	*Table-3 size before encryption (kb)*	*Execution time to table-3* (milliseconds)	*Table-3 size after encryption (kb)*			
Sensitive data 1.xlsx (100)	144	9264.195	344	160	4461.8692	272	200	2769.7335	416	504	1032	16495.7977
Sensitive data 2.xlsx (200)	280	29220.6266	704	304	15248.2485	552	400	12313.7182	888	984	2144	56782.5933
Sensitive data 3.xlsx (300)	336	105254.7383	1072	368	72217.8029	856	472	45819.9576	1136	1176	3064	223292.4988
Sensitive data 4.xlsx (400)	544	99057.7855	1464	592	64626.2652	1096	784	46782.2162	1752	1920	4312	210466.2669
Sensitive data 5.xlsx (500)	672	144475.2165	1808	736	88936.2767	1368	976	66912.4065	2176	2384	5352	300323.8997
Sensitive data 6.xlsx (600)	792	195080.4049	2128	872	125558.7778	1648	1160	83797.3951	2584	2824	6360	404436.5778
Sensitive data 7.xlsx (700)	912	266253.3033	2520	1000	193693.521	1928	1328	135680.0629	3000	3240	7448	595626.8872
Sensitive data 8.xlsx (800)	1024	351115.166	2840	1120	224623.1104	2200	1496	158960.6934	3328	3640	8368	734698.9698
Sensitive data 9.xlsx (900)	1160	428484.2833	3160	1272	247480.6719	2456	1712	178722.0331	3832	4144	9448	854686.9883
Sensitive data 10.xlsx (1000)	1280	522732.1352	3536	1408	406004.349	2736	1888	258095.8818	4208	4576	10480	1186832.366

Table- IV: Execution time of different size dataset with RC2 technique

Partitioned table	Table1			Table2			Table3			Total size of all 3 table (kb)	Total memory occupied after encryption	Total execution time (milliseconds)
File name with data size (kb).	*Table-1 size before encryption (kb)*	*Execution time to table-1 (milliseconds)*	*Table-1 size after encryption (kb)*	*Table-2 size before encryption (kb)*	*Execution time to table-2* (milliseconds)	*Table-2 size after encryption (kb)*	*Table-3 size before encryption (kb)*	*Execution time to table-3* (milliseconds)	*Table-3 size after encryption (kb)*			
Sensitive data 1.xlsx (100)	144	7165.003	304	160	5509.7986	248	200	3741.7822	392	504	944	16416.5838
Sensitive data 2.xlsx (200)	280	22822.654	624	304	14984.526	496	400	11026.9649	856	984	1976	48834.1449
Sensitive data 3.xlsx (300)	336	89817.72	880	368	61337.2469	712	472	39175.6097	1056	1176	2648	190330.5766
Sensitive data 4.xlsx (400)	544	80657.6471	1272	592	51676.9668	984	784	40656.7007	1680	1920	3936	172991.3146
Sensitive data 5.xlsx (500)	672	120777.8498	1568	736	80401.6261	1224	976	55965.7562	2088	2384	4880	257145.2321
Sensitive data 6.xlsx (600)	792	170473.9165	1896	872	134880.7295	1464	1160	92034.706	2480	2824	5840	397389.352
Sensitive data 7.xlsx (700)	912	268269.972	2176	1000	173321.4423	1704	1328	126189.5162	2864	3240	6744	567780.9305
Sensitive data 8.xlsx (800)	1024	356442.861	2496	1120	225062.8877	1928	1496	161376.5962	3184	3640	7608	742882.3449
Sensitive data 9.xlsx (900)	1160	400000.9189	2768	1272	232705.8421	2168	1712	160637.0658	3656	4144	8592	793343.8268
Sensitive data 10.xlsx (1000)	1280	423874.6214	3080	1408	264987.826	2408	1888	192623.0556	4032	4576	9520	881485.503

Table- V: Execution time of different size dataset with DES, Rijndael & RC2 technique [10]

Partitioned table	Table1			Table2			Table3			Total size of all 3 table (kb)	Total memory occupied after encryption	Total execution time (milliseconds)
File name with data size (kb).	*Table-1 size before encryption (kb)*	*Execution time to table-1 (milliseconds)*	*Table-1 size after encryption (kb)*	*Table-2 size before encryption (kb)*	*Execution time to table-2* (milliseconds)	*Table-2 size after encryption (kb)*	*Table-3 size before encryption (kb)*	*Execution time to table-3* (milliseconds)	*Table-3 size after encryption (kb)*			
Sensitive data 1.xlsx (100)	144	7419.4684	304	160	4461.8692	272	200	3741.7822	392	504	968	15623.1198
Sensitive data 2.xlsx (200)	280	23401.6023	624	304	15248.2485	552	400	11026.9649	856	984	2032	49676.8157
Sensitive data 3.xlsx (300)	336	85790.592	872	368	72217.8029	856	472	39175.6097	1056	1176	2784	197184.0046
Sensitive data 4.xlsx (400)	544	100706.7793	1272	592	64626.2652	1096	784	40656.7007	1680	1920	4048	205989.7452
Sensitive data 5.xlsx (500)	672	141062.8084	1576	736	88936.2767	1368	976	55965.7562	2088	2384	5032	285964.8413
Sensitive data 6.xlsx (600)	792	191739.577	1896	872	125558.7778	1648	1160	92034.706	2480	2824	6024	409333.0608
Sensitive data 7.xlsx (700)	912	236152.7501	2184	1000	193693.521	1928	1328	126189.5162	2864	3240	6976	556035.7873
Sensitive data 8.xlsx (800)	1024	309889.2111	2472	1120	224623.1104	2200	1496	161376.5962	3184	3640	7856	695888.9177
Sensitive data 9.xlsx (900)	1160	362130.6871	2808	1272	247480.6719	2456	1712	160637.0658	3656	4144	8920	770248.4248
Sensitive data 10.xlsx (1000)	1280	472875.2347	3080	1408	406004.349	2736	1888	192623.0556	4032	4576	9848	1071502.639

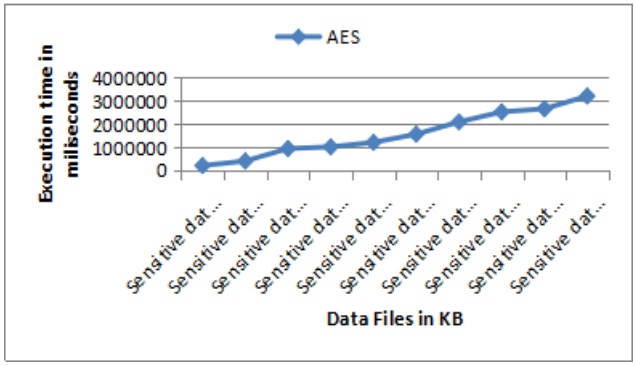

Fig.4. Performance analysis using AES technique based on execution time

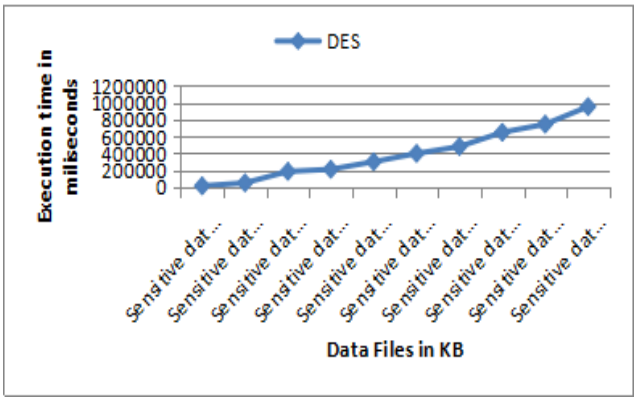

Fig.5. Performance analysis using DES technique based on execution time

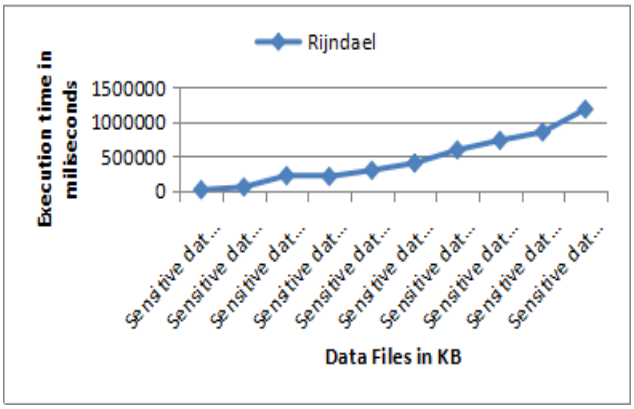

Fig.6. Performance analysis using Rijndael technique based on execution time

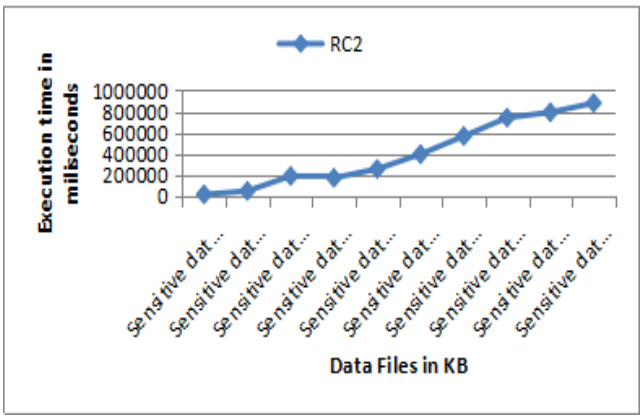

Fig.7. Performance analysis using RC2 technique based on execution time

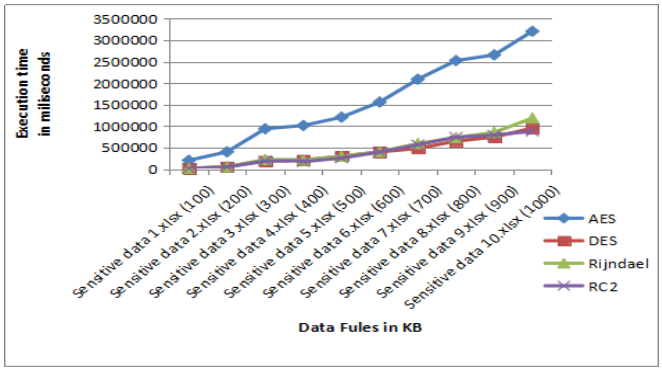

Fig.8. Performance of AES, DES, Rijndael & RC2 technique based on execution time

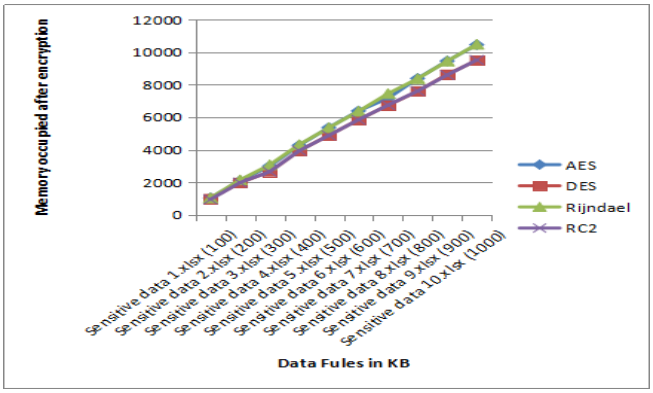

Fig.9. Performance of AES, DES, Rijndael & RC2 technique based on memory occupied after encryption

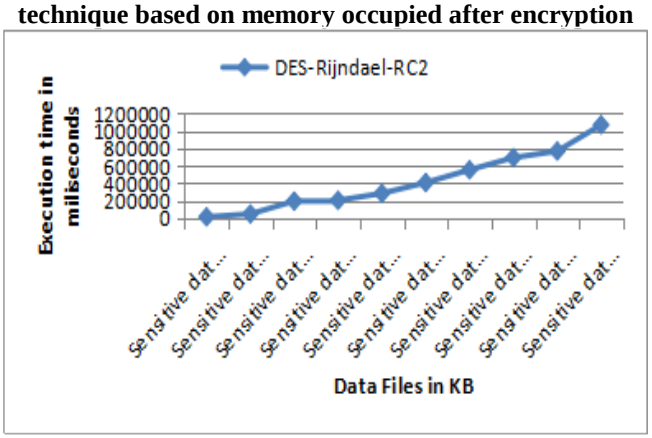

Fig.10. Performance analysis using Rijndael technique based on execution time

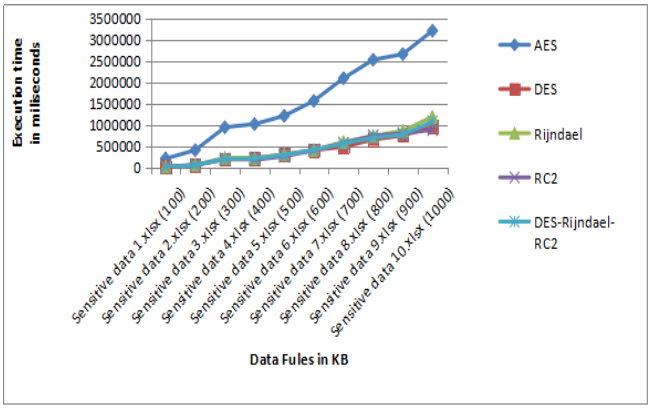

Fig.11. Performance of AES, DES, Rijndael, RC2 & DES-Rijndael-RC2 technique based on execution time

VI. CONCLUSION

The experimental data and graphs indicate that the performance with respect to encryption execution time and memory occupied after encryption of a Hybrid of algorithm (DES-Rijndael-RC2-RSA) algorithm is almost same as compare to AES, DES, Rijndael and RC2 algorithm as an individual. Fig.8. indicates the performance of AES, DES, Rijndael & RC2 technique based on memory occupied after encryption which is almost symmetrical for each algorithm so the hybrid algorithm is better option occupying memory too. So instead of using only one symmetric encryption techniques for all partitioned, it is better to use different symmetric encryption for each table. So the recommended algorithm will be more secure in this hybrid algorithm. As per proposed hybrid algorithm is advisable to enhance the overall performance of the system. so this attributed based encryption with vertically partitioned is more suitable in privacy-preserving data mining as in privacy-preserving main concern is to protect privacy for sensitive attributes of an individual.

REFERENCES

1. Dani, Virendra, Shubham Kothari, and Himanshu Panadiwal. "PPARM: Privacy Preserving Association Rule Mining Technique for Vertical Partitioning Database." International Conference on Innovations in Bio-Inspired Computing and Applications. Springer, Cham, 2018.
2. Upadhyay, Somya, et al. "Privacy preserving data mining with 3-D rotation transformation." Journal of King Saud University-Computer and Information Sciences 30.4 (2018): 524-530.
3. Abouelmehdi, Karim, Abderrahim Beni-Hessane, and Hayat Khaloufi. "Big healthcare data: preserving security and privacy." Journal of Big Data 5.1 (2018): 1.
4. Sankar, Athira, and Soumya Murali. "A Survey on Efficient Privacy-Preserving Ranked Keyword Search Method." (2017).
5. Ansar, Mehreen, et al. "Security of Information in Cloud Computing: A Systematic Review." American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS) 48.1 (2018): 90-103.
6. Devendra I Vashi et al., Critical Study and Analysis for Deciding Sensitive and Non-Sensitive Attributes of Medical Healthcare dataset Through Survey and Using Association Rule Mining. Int J Recent Sci Res. 8(5), pp. 17218-17222.
7. Devendra I Vashi et al (2017), "Challenges & Opportunities in Privacy Preserving Data Mining for Healthcare Dataset", International Conference on "Research and Innovations in Science, Engineering & Technology", ICRISET-2017
8. Jain, Amit, and Divya Bhatnagar. "A Comparative Study of Symmetric Key Encryption Algorithms." IJCSN Int. J. Comput. Sci. Netw 3.5 (2014): 2277-5420.
9. Chhinkaniwala, Hitesh, and Sanjay Garg. "Privacy preserving data mining techniques: Challenges and issues." Proceedings of International Conference on Computer Science & Information Technology, CSlT. 2011.
10. Kuntal Patel, "Performance analysis of AES, DES and Blowfish cryptographic algorithms on small and large data files", International Journal of Information Technology, ISSN: 2511-2104
11. Demir, Salih, and Bulent Tugrul. "Privacy-preserving trend surface analysis on partitioned data." Knowledge-Based Systems 144 (2018): 16-20.
12. Zhang, Lili, et al. "A Survey on Privacy Preserving Association Rule Mining." 2018 5th International Conference on Information Science and Control Engineering (ICISCE). IEEE, 2018.
13. Narwaria, Mamta, and Suchita Arya. "Privacy preserving data mining—'A state of the art'." 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2016.
14. Sharma, Surbhi, and Deepak Shukla. "Efficient multi-party privacy preserving data mining for vertically partitioned data." 2016 International Conference on Inventive Computation Technologies (ICICT). Vol. 2. IEEE, 2016.

AUTHORS PROFILE

Prof Devendrasinh Vashi is working as an Assistant Professor in the Computer Science and Engineering Department. He has more than 13 years of teaching experience. Prof Vashi received his MCA degree from the Visvesvaraya Technological University, Belgaum. He has published two research article in reputed journal. He has also presented two research papers in international conference. He has organized one Shor Term Training Programme successfully. He is currently pursuing his doctoral studies from C U Shah University, Surendranagar in the field of Privacy Preserving Data Mining.

Dr. H B Bhadka is working as Dean at Faculty of Computer Science, C. U. Shah University, Wadhwan City. Gujarat, India. He has completed his Ph. D. in 2009, from Saurashtra University. Currently working as Dean of Faculty of Computer Science, C. U. Shah University, Surendranagar. Working as Head of Institute since 2006 at CCMCA. Published 70+ Research Papers, attended many conferences, attended many workshops. He has successfully supervised two PhD dissertations and is currently guiding two PhD students in the field of Data Mining, privacy preserving in data mining. He is also a reviewing the PhD thesis of his research area.

Dr. Kuntal Patel is currently working as an Assistant Professor at School of Computer Studies, Ahmedabad University. Being a researcher on "IT Standards", he has completed his Ph. D. from North Gujarat University in 2006. He has published more than 25 research papers at peer-reviewed Journals and Conferences. He is certified Cyber Security Professional. He is actively involved in Google, ACM-India and Govt. of Gujarat supported Activity Based Learning Project called CS-Pathshala. He is an editorial board member of Computer Science textbook of Gujarat Board. He is the Secretary and Professional member of Association of Computer Machinery (ACM) – Ahmedabad chapter and life time member of CSI and ISTE.

Dr Sanjay Garg is working as Professor in Computer Science and Engineering Department. He has more than 31 years of teaching experience. Dr Garg has done his BE in Computer Technology from SATI, Vidisha (Barkattullah University, Bhopal) in 1991, ME in Computer Engineering and Automation from SGISTS, Indore in 2001 and PhD in Computer Science from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal in 2009. He has completed three funded research projects as Principal Investigator, two of them were sponsored by ISRO, under RESPOND scheme, other one by GUJCOST. Additionally two research projects sponsored by ISRO are in progress currently. He has successfully supervised six PhD dissertations and is currently guiding two PhD students in the field of Data Mining, Pattern Recognition and Image Processing.