Bulletin of Environment, Pharmacology and Life Sciences
Bull. Env. Pharmacol. Life Sci., Vol 4 [10] September 2015: 94-97

Online ISSN 2277-1808
Journal's URL:http://www.bepls.com
CODEN: BEPLAD
Global Impact Factor 0.533
Universal Impact Factor 0.9804

ORIGINAL ARTICLE **OPEN ACCESS**

Propose a new Architecture based on Data Masking to reduce memory overhead in big data Warehouses

Jalil Gazalan Toosi[1], Hamid Tabatabaee[2,*], Hamid Reza Ghaffary[3]
[2]Young Researchers and Elite Club, Mashhad Branch, Islamic Azad University, Mashhad, Iran
[2,*]Young Researchers and Elite Club, Quchan Branch, Islamic Azad University, Quchan, Iran
[3]Depatment of Computer Engineering, Ferdos Branch, Islamic Azad University, Ferdos, Iran

ABSTRACT
Data warehouse is a base of huge data which includes great information such as financial information, organization secrets, credit card Number and other personal information. Based on the important and expansion of information inside data warehouse, to mask and protect this information against prohibited people is a critical issue. Security in big data warehouse and in identification of invasive people is very essential. The suggested methods to protect data privacy besides its effects, has a lot of costs for the data warehouse. Therefore to decrease this cost (that usually is memory and time overhead) and increase Data securing, date masking method has been recommended. Using random solutions causes increase general securing, but the overhead generated by these methods are still one of the challenges in big data warehouse. In this paper, with adding an intelligent layer based on Log file analysis as a main source to recognize attack that analyze the behavior of user in log file, we can overcome user attacks, and with improving data masking method we can overcome memory overhead in big data warehouse.
Our Results show that
Key words: Big Data warehouse, distributed data base, data security, data warehouse, data masking,

Received 12.02.2015 Revised 09.05.2015 Accepted 11.07.2015

INTRODUCTION

Data warehouse is generally a date base which in charge of collecting and storing historical and present business information[1] data warehouse would include critical information such as financial information, organization secrets, credit card Number and other individual information in which causes a required motivation to attack this critical informative sources in date store, it should be assured that critical information is not to be trapped in unknown people, especially when date contains an essential information of data warehouse users. The published statistics show that the number of attack in data has been highly increased[2].
So, effectiveness of providing security of stored information in big data warehouse is very important. Also, memory and time overhead created to secure data is very essential, in huge size of data warehouse it would cost a lot.
In this paper with analyzing Log file we can overcome user attacks and with proposing an active data masking method try to overcome memory overhead.
In following text in section 2 we introduce the related works and in section 3 introduce the system architecture, in section 4 shows the Results and Evaluation and in section 5 is Conclusion.

System Architecture

The purpose system in this paper is usable in all database especially distributed database would have its specific users with various access levels. In suggested architecture having several distributed database in the level of an organization and or through the Internet which they communicate altogether, we design. In figure 1, suggested architecture diagram block has been shown for a database. In suggested system architecture, there are several main nodes which are as follows: masked database, security software, proxy which we illustrate them furthered.

Masked Database

This database includes our main information which we try to protect them. In this date base in each table based on the security level for any records, we have a key k3 which we use in masking data operation. To choose key k3, we consider various policies. Depend on policy Key k3 can be created for any records uniquely or create in limit count. This selection would have a direct influence on created memory overhead in the suggested architecture. DBMS is as the manager of the database which manages the database.

Proxy

Proxy is another security layer in our suggested system all requests are first sent to proxy. Proxy has a security key called k1 which we use to mask data.

Security Software

Instead any date base, we have a security software which its task is to monitor sent queries to database and assess their value and if valuable, the requests are they care be executed. Log file: all accumulations are registered in data store in file.

Black box: the relevant information to user's access levels of date base are stored in this table.

Table TA^i: this table contains the accuracy of users operation with a series of additional information such as the permission of any requests, transaction number, transaction time, table name; operation type and etc. are registered.

Key K2: is a security key in which we use in masking data operation.

Node G: this node would assess the existing requests in Log file and register accuracy of operation with some additional information in table TA.

Any database is connected through a communication line to security software and then to proxy. All the requests by the users to be done on any database are sent to database in this way.

Execution process

The user A begins to query one of the existing databases in system and sent its own request to proxy. The proxy receives the request and sends that request to the security software of database and the security software locks at its TA table and finds the last same user's request (meaning, similarity in accessible date) to figure out its access level. When it finds such a request and the time of its execution had not lasted a lot (the time can be set by the user) that request can be monitored. If it has not execution permission, it would reject the request.

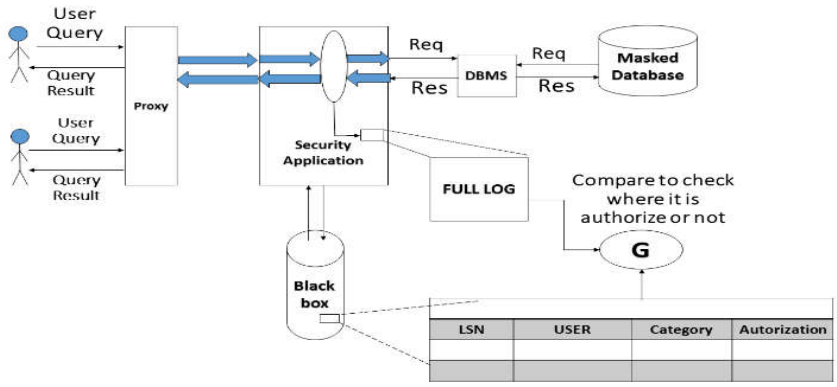

Figure 1 block diagram of proposed method

If it has execution permission or couldn't find a permission (this is the first time of executing the request) it would send user's present request with key K2 to database manager and update Log and its TA.

The database manager would unmask the result with K2 and k3 with an algorithm aided after receiving the request and send security software and security software out send it to proxy after receiving. Proxy would also unmask it with the aid of its own key k after receiving response from security software and give the result to the relate user. It is good to mention that a standard as a maximum of the number of permission to have an access to any users, we should limit the number its requests.

Masking And Unmasking

To mask date we apply this formula:

$$(R_j, C_i)' = (R_j, C_i) - \left((K3_j \, MOD \, K2) \, MOD \, (K2 * 2)\right) + K_1$$

This formula looks like the aforementioned method in[7] with a little change.

In the above formula $C_i$is the column and R_j is the row which we mask its data. To unmask we apply this formula:

$$(R_j, C_i) = (R_j, C_i)' + \left((K3_j \, MOD \, K2) \, MOD \, (K2 * 2)\right) - K_1$$

K_1 and $K2$ are two private key with 128 bytes length. K_1 is maintaining in proxy and $K2$ is stored in security software and $K3_i$ is a general key with 128 bytes length which based on our policy we can use in encryption for any records, we can have this value To increase the speed of data masking in the beginning of data masking operation and saving in space instead applying previous methods[11] which $K2$ and $K3$ would place for any row and column of a various random number, we calculate K_1 and $K2$ as fixed once and $K3$ randomly but in a limited number.

Selection Method For k3

We consider M which represents the number of k3 keys which we create for records and produce the keys randomly and maintain these keys in a table. To find key k3 for masking new records, if the quotient of the new record number (row number or etc.) on to M is zero, we create k3 randomly (this means, these data are initial records) and if the quotient is one, we read k3 from the record number one (result above division plus 1) of k3 table. For example if the result is zero we read it from the first record and if the result is f, we read the value of k3 from f+1 record. For greater M the security gets higher. In this method we should watch the k3 records not to be omitted physically.

RESULTS

Evaluation Method

In this article for evaluation results, calculate the memory overhead create by keys that use for masking, in[11] for each record use 2 key with 128 bit length, so if the number of records equals to 1,000,000 then memory overhead calculate as below :

Memory overhead = (len (K3) +len (K2))*record counts+len (K1)

For 1,000,000 record memories overhead equals to 256,000,000 bit or 30.5 megabyte.

In propose method memory overhead calculate as below:

Memory overhead= (len (K1) +len (K2))+M*len (K3)

For 1,000,000 record with M=1000 memory overhead equals to 128,256 bit or 0.015 megabyte.

DISCUSSION

The mentioned method in this paper with adding another security level as time goes by tries to maintain database security and also with purpose data masking method in this article data masking overhead has been reduced a lot .but the problem is that proxy creates a single point of failure. Proxy out of service causes the whole date base be out of service. In data masking method which we has applied in this article, while calculating k3 for any records separately The security is much higher, but memory overhead is a little higher, although the purpose method decrease the of memory overhead. In table 1 the extent of memory overhead in purpose method and the old method[11] have been compared. The overhead of memory due to megabyte has been calculated.

Table 1 compare memory overhead in purposed method and the old one

Record count	1000000	10000000	100000000	1000000000
Old method	30.517578	305.1758	3051.7578	30517.5781
purpose method M = 1000	0.015	0.015	0.015	0.015
purpose method M= 10000	0.1525879	0.152588	0.1525879	0.15258789
purpose method M = 100000	1.5258789	1.525879	1.5258789	1.52587891

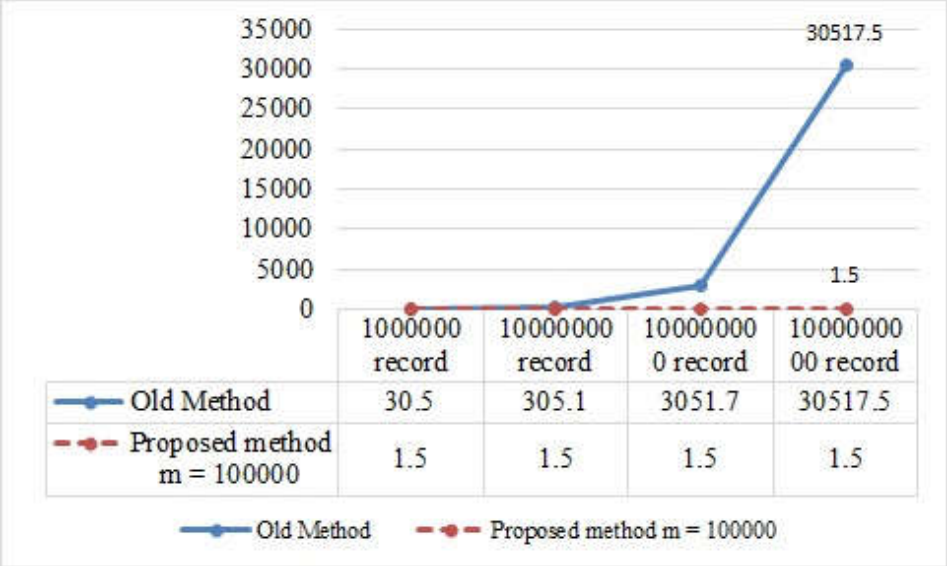

Figure 2 Compare old method And Proposed Method with m = 100000

In figure 2 show the compare between old method and proposed method with m = 100000, X-Axis is number of record and Y-Axis is memory overhead in Megabyte for masking database records.

As shown Figure 2, Old method in database with 1000000000 record has more memory overhead than the proposed method.

Cause of this disparity is difference between used algorithms for masking.

CONCLUSION

With the increasing size of big Data warehouses containing very important information, security technique are more important and reducing memory overhead in these techniques are an important challenge. The purpose method in this article while adding a security layer to overcome users attacks use an dynamic data masking method to reducing the memory overhead. Our method can improve the memory overhead dynamically, According to distributing the keys in three different parts; access to them for an intruder would be very difficult.Intelligent analysis of queries that sent to the data warehouse by node G in software, considering uncertainty and the send time queries are some of the feature works.

REFERENCES

1. Baer, H., (2004). On-Time Data Warehousing with Oracle Database 10g–Information at the Speed of Your Business. Oracle Whitepaper, Oracle Corporation.
2. Yuhanna, N., (2010). Your Enterprise Database Security Strategy. Forrester Research.
3. Agrawal, R., R. Srikant, and D. Thomas. (2005). Privacy preserving OLAP. in Proceedings of the ACM SIGMOD international conference on Management of data. ACM.
4. Wang, L., D. Wijesekera, and S. Jajodia, Cardinality-based inference control in sum-only data cubes, in Computer Security—ESORICS 2002. 2002, Springer. p. 55-71.
5. Gupta, S., M. Sonali, and M. Palak, Data Warehouse Vulnerability and Security. International Journal of Scientific & Engineering Research, 2012. **3**(5).
6. Ventrapragada, V.S., A Review on Data Security in Data Warehousing.
7. Santos, R.J., J. Bernardino, and M. Vieira. A survey on data security in data warehousing: Issues, challenges and opportunities. in EUROCON-International Conference on Computer as a Tool (EUROCON), 2011 IEEE. 2011. IEEE.
8. Lee, S.Y., W.L. Low, and P.Y. Wong, (2002).Learning fingerprints for a database intrusion detection system, in Computer Security—ESORICS Springer. p. 264-279.
9. Weippl, E.R., (2010). Security in Data Warehouses. IGI Global, Data Ware-housing Design and Advanced Engineering Applications.
10. Corporation, O., (2010).Oracle Advanced Security Transparent Data Encryption Best Practices. Oracle White Paper.
11. Singh, A. and N. Umesh, (2013).Implementing Log Based Security in Data Warehouse. International Journal of Advanced Computer Research (IJACR) Volume-3 Number-1 Issue-8 23-29.

CITATION OF THIS ARTICLE

Jalil Gazalan T, Hamid T, Hamid Reza G .Propose a new Architecture based on Data Masking to reduce memory overhead in big data Warehouses. Bull. Env. Pharmacol. Life Sci., Vol 4 [10] September 2015: 94-97