An Efficient Data Outsourcing Scheme Using Non Intrusive Data Forecasting Algorithm

[1]Dr. V. Latha Jothi, [2]S. KalaiVani, [3]D. Keerthana, [4]K. Kiruthikayani

[1]Associate Professor, Department of Computer Science & Engineering, VCET, TN, India.

[2, 3, 4] BE, CSE, VCET, TN, India.

Received date: 2nd June, 2018, Revised Date: 20th June, 2018, Accepted Date: 24th June, 2018.

Abstract - Cloud computing allows end-users with limited resources to outsource their private data-sets to a third-party service provider. Database outsourcing is a noteworthy solution to improve quality of services while reducing data management costs. When data is stored and processed out of the territory of its owner, security becomes the first concern. Confidentiality of the outsourced data, correctness assurance of query results, and preserving user's access privacy are the primary requirements of secure data outsourcing. Nevertheless, most of research activities concentrate on confidentiality based on different encryption schemes. This paper proposes a secure data storage model in cloud computing environments that is based on the concept of data slicing and presents its prototype tool that supports the low-cost migration of existing applications. Our tool provides a structured query language (SQL) translation mechanism that provides transparent access to partitioned data without changing the original SQL queries. This concentrates on compelling strategy that can be utilized for giving better information utility and can deal with high dimensional information.

Keywords - Cloud Computing, Outsourcing Scheme, Encryption, Decryption, Sensitive Data Detector.

I. INTRODUCTION

Recent years have witnessed increasing need for large amounts of digital information to be collected and studied. To address the big data need, the database-as-a-service model was introduced, facilitated by the evolution of cloud computing. We consider the framework that contains three parties: the data owner, the users (clients), and the service provider (server). The data owner has a dataset that will be outsourced to the server. If the outsourcing service involves hosting information systems at a third party data center, an on-site visit to assess the security environment of the hosting company should be conducted before making any final decision to outsource.

Similarly, if customer data or other sensitive information is to be transferred to servers owned by a service provider, a security risk assessment covering the physical and logical security controls at the premises hosting the servers should be conducted before sensitive data is released to the service provider. The service provider should set up an isolated environment to segregate the organization's data from that of other clients. Communication paths used to transfer the data must be secure, and sensitive data should also be encrypted using strong encryption algorithms.

However, recent study has shown that deterministic encryption at a fine granularity may not be robust against adversary data dependency constraints. A typical type of data dependency constraints is the functional dependency (FD). Informally, a FD uniquely determines may over-protect the data and dramatically reduce the data usability. Only encrypting a portion of non-sensitive data values besides the sensitive values is sufficient to defend against the FD attack. The experimental results demonstrate that our greedy approach is ten times faster than the optimal solution, with as small as 1% of the overhead compared with the state-of-art solution.

II. RELATED WORK

Several micro data anonymization techniques have been proposed. The most popular ones are two popular anonymization techniques are generalization and bucketization. Generalization is one of the commonly anonymization approaches, which replaces quasi-identifier values with values that are less-specific but semantically consistent. Then, all quasi-identifier values in a group would be generalized to the entire group extent in the quasi-identifier possible items in the log.

The main problems with generalization are:

- It fails on high dimensional data due to the curse of dimensionality and
- It causes too much information loss due to the uniform-distribution assumption.

Bucketization first partitions tuples in the table into buckets and then separates the quasi identifiers with the sensitive attribute by randomly permuting the sensitive attribute values in each bucket. The generalized data consist of a set of buckets with permuted sensitive attribute values. In particular, bucketization has been used for high-dimensional data. However, their approach assumes a clear separation between quasi-identifiers and

sensitivity analysis. In addition, because the exact values of all quasi-identifiers are released, membership information is disclosed. In both generalization (b) and bucketization (c), one first moves identifiers from the data and then partitions tuples into buckets. The two techniques differ in the next step. Generalization transforms the quasi-identifier values in each bucket into "less specific but semantically consistent" values so that tuples in the Sensitivity Analysis me bucket cannot be distinguished by their quasi-identifier values. In bucketization, one separates the sensitivity analysis s from the quasi-identifiers by randomly permuting the sensitivity analysis values in each bucket.

A. Functional Procedure

Step 1: Extract the data set from the database.
Step 2: Anonymity process divides the records into two.
Step 3: Interchange the sensitive values.
Step 4: Multi set values generated and displayed.
Step 5: Attributes are combined and secure data displayed.

III. PROPOSED METHOD

Many algorithms like bucketization, generalization have tried to preserve privacy however they exhibit attribute disclosure. So to overcome this problem an algorithm called slicing is used. Initially, the data-set extraction can be used to extract the data-set and it will be stored in the database for future use. The dataset was selected, after that it will be split separate data and it can be stored in the table to the user database.

A. Selective Sensitive Data Detector Algorithm

- Every application will have its one type of sensitive and insensitive data and it will be analysed keenly.
- Multi valued attribute will be considered as single value and converted to atomic attribute.
- Quasi identifier will be considered for selection of sensitive attribute. It is used to select sensitive data dynamically.

In this paper, we introduce a novel data anonymization technique called slicing to improve the current state of the art. Slicing partitions the data set both vertically and horizontally. Vertical partitioning is done by grouping attributes into columns based on the correlations among the attributes. Each column contains a subset of attributes that are highly correlated. Horizontal partitioning is done by grouping tuples into buckets. Finally, within each bucket, values in each column are randomly per-mutated (or sorted) to break the linking between different co The basic idea of slicing is to break the association cross columns, but to preserve the association within each column. This reduces the dimensionality of the data and preserves better utility than generalization and bucketization. Slicing preserves utility because it groups highly correlated attributes together, and preserves the correlations between such attributes. Slicing protects privacy because it breaks the associations between uncorrelated attributes, which are infrequent and thus identifying. Note that when the data set contains quasi-identifier and one sensitivity analysis , bucketization has to break their correlation; slicing, on the other hand, can group some quasi-identifier attributes with the sensitivity analysis , preserving attribute correlations with the sensitive attribute. The key intuition that slicing provides privacy protection is that the slicing process ensures that for any tuple, there are generally multiple matching buckets.
The algorithm we have used for matching buckets:

B. Non Intrusive Data Forecasting Algorithm

- Grouping data with reference to the similar values of attribute
- Combine data with refer to more than one similar value of attribute

C. Advantages

- It is used to determine the sensitive and insensitive data in the dataset.
- It is developed using heuristic technique based on the intersection pattern of frequent item-sets to secure the set of sensitive association rules employing distortion method.
- It is used to secure the delicate association rules using multiple items in consequent and antecedent.
- The proposed algorithm is balanced in terms of accuracy, performance, and privacy protection.

D. Slicing Algorithm

- This algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning.

E. Attribute Partitioning

This algorithm partitions attributes so that highly correlated attributes are in the same column. This is good for both utility and privacy. In terms of data utility, grouping highly correlated attributes preserves the correlations among those attributes. In terms of privacy, the association of uncorrelated attributes presents

higher identification risks than the association of highly correlated attributes because the associations of uncorrelated attribute values is much less frequent and thus more identifiable.

F. Tuple partitioning

While column generalization may result in information loss, smaller bucket-sizes allow better data utility. Therefore, there is a trade-off between column generalization and tuple partitioning.

G. Column Generalization

Although column generalization is not a required phase, it can be useful in several aspects. First, column generalization may be required for identity/membership disclosure protection. If a column value is unique in a column a tuple with this unique column value can only have one matching bucket.

H. Multi Class Decomposition Algorithm

- Decompose the data into number of tables so that the integrity and normalization is maintained.
- It is used to fit the global values so that data prediction is impossible.
- It determines the threshold value for each attribute's privacy level so that it would be maintained to the entirety of the implementation.
- Data consistency will be maintained in this algorithm.

Tables 1 (a), (b) (c) and (d) show the original, generalized, Bucketized and sliced tables.

TABLE 1 (A) ORIGINAL TABLE

Age	Sex	Zip code	Disease
22	M	47906	dyspepsia
22	F	47906	flu
33	F	47905	flu
52	F	47905	bronchitis
54	M	47302	flu
60	M	47302	dyspepsia
60	M	47304	dyspepsia
64	F	47304	gastritis

TABLE 1 (B) GENERALIZED TABLE

Age	Sex	Zip code	Disease
[20-52]	*	4790*	dyspepsia
[20-52]	*	4790*	flu
[20-52]	*	4790*	flu
[20-52]	*	4790*	bronchitis
[54-64]	*	4730*	flu
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	gastritis

TABLE 1 (C) BUCKETIZED TABLE

Age	Sex	Zip code	Disease
22	M	47906	dyspepsia
22	F	47906	flu
33	F	47905	flu
52	F	47905	bronchitis
54	M	47302	flu
60	M	47302	dyspepsia
60	M	47304	dyspepsia
64	F	47304	gastritis

TABLE 1 (D) SLICED TABLE

(Age, Sex)	Zip Code, Disease
22,M	47906, dyspepsia
22,F	47906,flu
33,F	47905,flu
52,F	47905,bronchitis
54,M	47302,flu
60,M	47302,dyspepsia
60,M	47304,dyspepsia
64,F	47304,gastritis

IV. ARCHITECTURE

Figure 2 shows the architecture of the proposed method.

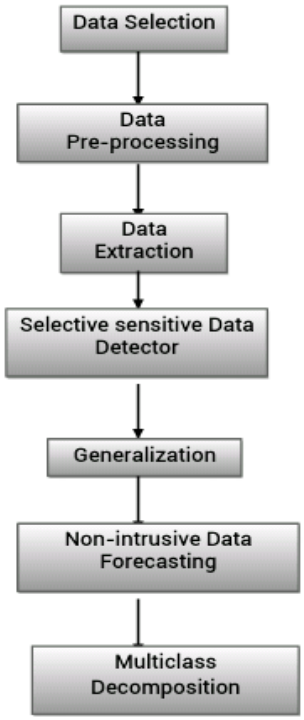

Fig. 2 Architecture of the proposed model

V. EXPERIMENTAL RESULTS

Graph generation can be used to find the classification accuracy between Original data, Generalization, Bucketization and Slicing. Slicing shows better accuracy than generalization. When the target attribute is the sensitive attribute, slicing even performs better than bucketization. Figure 3 shows the comparison chart.

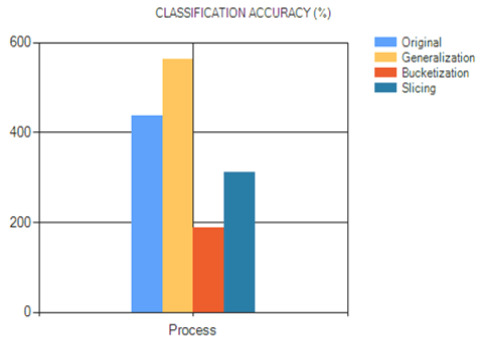

Fig. 2 Comparison Chart

VI. CONCLUSION

In this paper, Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. Slicing prevents attribute disclosure and membership disclosure. Slicing preserves better data utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute. We consider slicing where each attribute is in exactly one column. An extension is the notion of overlapping slicing, which duplicates an attribute in more than one column.

We plan to design more effective tuple grouping algorithms. Another direction is to design data mining tasks using the anonymized data computed by various Anonymization techniques. Slicing protect privacy by breaking the association of uncorrelated attributes and preserve data utility by preserving the association between highly correlated attributes. Another important advantage of slicing is that it can handle high dimensional data. Our experiments show that slicing the original data is very effective.

VII. FUTURE WORK

An extension is the notion of overlapping slicing, which duplicates an attribute in more than one column. This release more attributes correlations. Future it could provide better data utility, but the privacy implications need to be carefully studied and understood. It is interesting to study the trade-off between privacy and utility. The trade-off between column generalization and tuple partitioning is the subject of future work. The design of tuple grouping algorithms is left to future work.

REFERENCES

[1] Boxiang Dong and Wendy Wang, "Secure Data Outsourcing with Adversarial Data Dependency Constraints", IEEE 2nd International Conference on Big Data Security on Cloud, DOI: 10.1109/BigDataSecurity-HPSC-IDS.2016.17, 2016.

[2] Trang and Maruyama, "Secure Data Storage Architecture on Cloud Environments", Proceedings of 11th International Joint Conference on Software Technologies, Vol. 1, pp. 39-47, 2016.

[3] Konda Ta disetti, P.Madhuri, J.Bheemeswara Shastri, "Implementation of Slicing Technique for Privacy Preserving Data Publishing", International Journal of Research in Computer and Communication Technology, Vol. 3, No. 12, pp. 1736-1739, 2014.

[4] H. Wang and R. Liu. Privacy-preserving publishing micro data with full functional dependencies. Data & Knowledge Engineering, Vol. 70, No. 3, pp. 249-268, 2011.

[5] Y. Li, W. Dai, Z. Ming, and M. Qiu, "Privacy protection for preventing data over-collection in smart city", IEEE Transactions on Computers, Vol. 65, No. 5, pp. 1339-1350, 2015.

[6] M. Qiu, M. Zhong, J. Li, K. Gai, and Z. Zong, "Phase-change memory optimization for green cloud with genetic algorithm", IEEE Transactions on Computers, Vol. 64, No. 12, pp. 3528-3540, 2015.

[7] N. P. Smart and F. Vercauteren, "Fully homomorphism encryption with relatively small key and cipher text sizes", Public Key Cryptography (PKC), pp. 420-443. 2010.