Journal of Theoretical and Applied Information Technology
30th November 2016. Vol.93. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A CLASSIFICATION METHOD FOR IDENTIFYING CONFIDENTIAL DATA TO ENHANCE EFFICIENCY OF QUERY PROCESSING OVER CLOUD

[1] HUSSEIN ALBADRI, [2] ROSSILAWATI SULAIMAN

Center for Software Technology and Management, Faculty of Information Science & Technology,

Universiti Kebangsaan Malaysia, Selangor, Malaysia

E-mail: [1]h.albadry19@hotmail.com, [2]rossilawati@ukm.edu.my

ABSTRACT

With the increased use of Database-as-a-Service (DAAS), several issues also come in parallel, especially in translating and executing queries to and from the database securely and efficiently. These issues are in response towards potential attacks such as attempting to copy or eavesdrop the database via queries. Existing security mechanisms include securing the queries by using encryption. However, encrypting the queries significantly affects the efficiency of query processing because of the security overhead from the encrypting and decrypting processes. This study aims to address this problem by proposing a divide-and-conquer strategy in which partial encryptions is used on the queries. This is performed by classifying the data into sensitive and non-sensitive categories using a classification approach, so that only the sensitive data will be encrypted. The classification used in this study is based on the data classification policy from the Columbia University. Firstly, a manual annotation is conducted to label the data fields into sensitive and non-sensitive categories. Next, rules are generated in order to classify the queried data. If a query contains sensitive data, the encryption will specifically be applied to the sensitive data, whereas the non-sensitive data will remain unencrypted. Experiments have been conducted using real-time data from Baghdad University that is related to students' information consisting 35 tables and 362 fields. The evaluation is based on the comparison of security overhead of the fully encryption (without classification) and partial encryption (with the classification) using Advance Encryption Standard (AES). Results shown that the classification method has significantly reduced the time used to process the query. This implies that the partial encryption based on classifying the data into sensitive and non-sensitive categories has improves the efficiency of query processing.

Keywords: *Cloud Computing, Cloud Database, Secure Query Processing, Cloud Query Processing*

1. INTRODUCTION

With the arising of Database-as-a-Service (DAAS), several issue have been brought to attention such as securing query, translating query and executing query efficiently. These issues have been presented as a response toward potential attacks. According to [1], potential attacks could be represented by any attempt to copy or eavesdrop the database via queries. The challenging issue lies in the difficulty to detect or prevent such attempts due to the nature of cloud infrastructures which only provides access and management rather than protection [2].

In order to provide a secure query, the query itself has to be encrypted. Encryption has a significant impact on the failure execution for the query on the server. This is because the server cannot interpret encrypted queries. One approach to solve this problem is to use a Fully Homomorphic Encryption scheme such as one proposed by Gentry [3]. These schemes allow computation of any arbitrary function over encrypted data without the need to decrypt it. In this approach the client encrypts his data using a Fully Homomorphic Encryption scheme before storing it at the server. When the client wants to execute the query, it is transformed into an equivalent query over the encrypted data. This allows server to process any valid SQL queries on encrypted data and return the result set (still encrypted) to the client. The client decrypts this encrypted result set to get the final answer. However, this approach offloads all the database processing work to the server. The problem with this approach is that operations on current Fully Homomorphic Encryption schemes are orders of magnitude (10^9 times) slower than the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

operations on plain text data [4]. This makes this scheme impractical for use in current systems.

Another approach to solve this problem is to use service providers only as encrypted data storage. In this approach the client encrypts all the data before storing it at server. Whenever the client has to execute a query, the required part of the database is transferred to the client machine, and then decrypted to recreate plain text database. The query is executed on this plain text database using traditional query processing to get the result set. This approach has severe performance issues because for each query, huge network traffic (to transfer required relations) is generated. This approach also negates many benefits of DAAS model since client now has to maintain his own database servers.

An interesting question would be: "what is the appropriate way to secure the query processing in cloud without losing efficiency?". As a response to this question, researchers tend to use machine learning technique in order to classify the confidentiality of the data before migrating such data to the cloud provider. For instance For instance, Graepel et al. [5] proposed a linear classification method to classify whether the data is confidential or not to be encrypted. Such method is useful where the encryption will take place only on the confidential data.

However, there is still room for improvement regarding to the performance of the classification task. In fact, there are many classification methods that have different performance [6]. In such manner, identifying an appropriate classification method for categorizing the level of confidentiality for the data is a challenging task. Hence, this paper aims to propose an effective classification method for identifying the confidential data. As a result of the classification, the confidential data will be encrypted to be migrated in the cloud.

The paper has been organized as follows: Section 2 highlights the related work, Section 3 illustrates in detail the proposed method and its components, Section 4 discusses the results obtained by the proposed method, and finally Section 5 provides the final conclusion.

2. RELATED WORK

There are several cloud-based storage systems available, such as Dynamo [7], PNUTS [8], and Bigtable [9]. These systems have very high scalability, although it can only support very limited query languages. The restricted queries do not allow *join* queries and there are restrictions on how to specify the query conditions. In contrast, the BigIntegrator [9] pushes as much query processing as possible to the data sources and compensates the lacking query capability of a data source by doing post-query processing with its own query engine. Some cloud-based storage systems such as Cloudy [10] provides rather complete SQL capabilities. It offers key-value, SQL, and XQuery interfaces to manipulate its cloud data. Microsoft SQL Azure [11] offers full SQL language support for its cloud-based relational database. Unlike Cloudy and SQL Azure, the purpose of BigIntegrator is to allow joining of data from a restricted cloud-based data store such as Bigtable with relational databases, by generating execution plans that combine queries sent to the data sources. Unlike classical work on mediator/wrapper techniques over conventional databases such as [12], BigIntegrator provides data integration between cloud-based data repositories and relational DBMSs. Furthermore, a novel query plug-in mechanism based on absorbers and finalizers is developed to provide easy extensions for new kinds of data sources providing restricted query languages.

In particular, Wang et al. [13] have proposed a secure and efficient ranked keywords search on cloud data. Their assumption lies on the difficulty of searching over cloud data due to the encryption. In fact, there are several searchable encryption techniques enable users to securely search over encrypted data using keywords. However, such techniques only support Boolean search. This can directly affect the efficiency of searching. Therefore, the authors have concentrated on overcoming such problem by proposing a ranked search. They used the traditional statistical measures that have been used in information retrieval and text mining such as Term Frequency – Inverse Document Frequency (TF-IDF) and Mutual Information (MI). These statistical measures aim at ranking the data, based on relevancy.

Furthermore, Ren et al. [14] have concentrated on how to provide a confidential, private, efficient and low in-house processing cost query processing approach. In fact, they have proposed a Random Space Perturbation (RASP) approach using K-nearest neighbor (KNN) method. Their proposed method aims at providing random noise injection, dimensionality expansion and random projection to ensure optimum confidentiality. In addition, their proposed method preserves the topology of multidimensional range in secure transformation which enables indexing the queries. This indexing

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

will directly enhance the efficiency of processing the query. Furthermore, the proposed method has the ability to reduce the in-house processing workload. This is due to the low perturbation cost and high precision query results would be gained.

In order to propose an efficient retrieval method from cloud data, Graepel et al. [5] have proposed a machine learning technique for classifying conditional data before migrating it to the cloud provider. Their assumption lies on the high cost of encrypting the data and the impact of such encryption on the efficiency of the query processing. Therefore, there is an essential demand to accommodate a classification procedure in order to categorize the level of privacy for each record in the data.

Similarly, Zardari et al. [15] have presented a classification method for categorizing cloud data that are vulnerable to threats such as hacking. Such classification of the data can enhance efficiency of the cloud model where such classification will provide the user the required information (i.e. classified) based on the availability of this data or specific authentication adjusted by the database manager. Basically, the authors have encrypted the sensitive data and then divided the data into multiple classes. Meanwhile, the non-sensitive data will be classified without encryption.

3. PROPOSED METHOD

The architecture of the proposed method consists of three main components as shown in Fig. 1, which are the client side, the server side and the classification rules. The first component is the client side where the user initiates a query that may contain sensitive data such as:

*'Select * from Student where Student.Name = "Adam"'.*

Although not all the words in the query are sensitive, however the name *'Adam'* tend to be sensitive data. Hence, the query will undergo a classification process that aims to classify the sensitive and non-sensitive data.

The second component which is the classification rules aims to accommodate a processing task in order to determine which part of the query is sensitive. Several types of data security are provided by encryption, depending on the sensitivity of the data. Basically, if the query contains high sensitive data, a stronger encryption will be applied on the data. Otherwise, if the data contains less sensitive data, a weaker encryption will be applied on the data. On the other hand, if the data does not contain any private data, no encryption would take a place. The classification process will be further detailed in the next section.

The third component which is the server side, aims to receive the query. In this manner, if the query contains encrypted information, a decryption task is carried out in order to interpret the query. Once the query has been interpreted, the query will be executed and the retrieved data will be sent to the client side. A classification process first takes place in order to identify whether the retrieved data contains private data or not.

3.1. Data Collection and Standard for Data Classification

One of the challenging task behind cloud database lies on figuring out a real-time data that could be used for the research's purposes. This is due to the difficulty of publishing private data whether from corporations or organizations. In this study, a database for students in University of Baghdad has been used in order to simulate the query operations. This database consists of 35 tables and 362 corresponding fields. The data is associated with student information such as biography, academic details, payment details and courses details.

There are many standards and policies available classifying data based on its sensitivity. ISO provides information classification guideline in ISO 17799 [16], which classify information as Top Secret, Highly Confidential, Proprietary, Internal Use Only, and Public Documents. Standard used in the US government [17] are Top Secret, Secret, and Confidential. Australia and New Zealand governments have an additional criterion known as Restricted [17, 18].

However, this study used a standard produced by Columbia University [19] for classifying data, as this policy is the closest match to our data sample related to the university environment. According to this policy, there are four classes of sensitivity for the data which can be illustrated as follows:

i. Sensitive Data: any information protected by federal, state or local laws and regulations. This would include information about an individual that (a) can be used to distinguish or trace an individual's identity, such as name, date and place of birth, mother's maiden name or biometric records, (b) is linked or linkable to an individual, such as medical, educational, financial and employment information, which if lost, compromised or disclosed without authorization,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

could result in harm to that individual and (c) is protected by federal, state or local laws and regulation or industry standards. We consider this type of data to be highly sensitive. Therefore, highest data security should be applied at this data.

ii. Confidential Data: any information that is contractually protected as confidential by law or by contract and any other information that is considered by the University appropriate for confidential treatment. This would include student information such as a student's name, address, degrees and awards, subject to certain requirements. In addition, it may include the human resource information such salary and employee benefits information. We consider this type of data to be sensitive. Therefore, high data security should be applied at this data.

iii. Internal Data: any information that is proprietary or produced only for use by members of the University community who have a legitimate purpose to access such data. This would include tactic operations such as reports and technical document. We consider this type of data to be low sensitive. Therefore, low data security should be applied at this data.

Based on the above classes, a manual annotation task has been performed in order to label all the fields in the database with their actual classes. This annotated data will be used in the classification method. Table 1 shows a sample of this annotated data

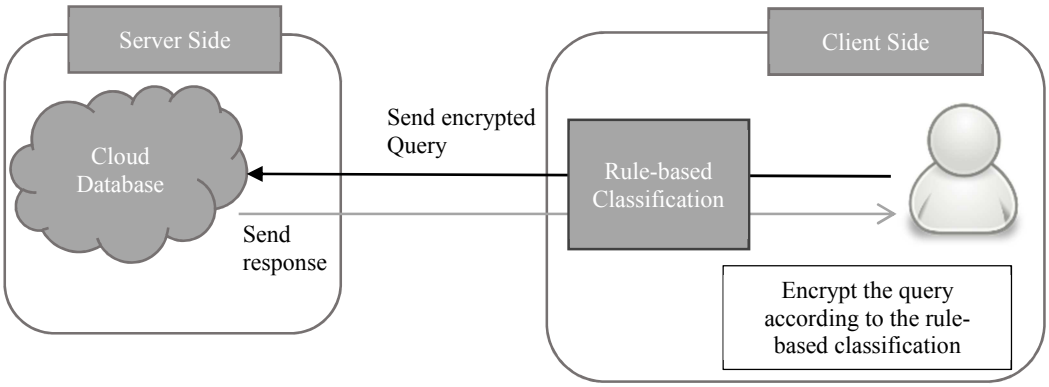


Figure.1. Architecture of the proposed method

Table 1. Sample of annotated data

Table	Field	Class
pass_word	Usernm	Sensitive
pass_word	Pass	Sensitive
pass_word	Priority	Sensitive
category	cat_id	Internal
category	cat_name	Internal
category	Acadmic_year	Internal
Religion	Rid	Internal
Religion	rel_name	Public

The annotated data will undergo partial encryption task in which only the sensitive information will be encrypted, and the non-sensitive information will be ignored

3.2. Classification

This phase aims to identify the classification method to categorize the data into sensitive and non-sensitive. We use a rule-based classification method for this purpose. This kind of classification method relies on a predefined list of rules [20]. As mentioned earlier, the rules have been generated based on the data classification policy produced by the Columbia University [19]. Fig. 2 shows a sample of generated rules.

Journal of Theoretical and Applied Information Technology
30th November 2016. Vol.93. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
If Table.Name == 'Biography' &&
   Field.Name == 'Name'    → sensitive
If Table.Name == 'Contact' &&
   Field.Name == 'mobile' → sensitive
If Table.Name == 'Grades' &&
   Field.Name == 'GPA'     → sensitive
If Table.Name == 'Student' &&
   Field.Name == 'faculty' → non-sensitive
If Table.Name == 'Student' &&
   Field.Name == 'Id'      → non-sensitive
If Table.Name == 'Student' &&
   Field.Name == 'status'  → non-sensitive
```

Figure 2. Sample of generated rules

3.3. Encryption

In this phase, a partial encryption is applied on the sensitive information using the Advanced Encryption Standard (AES), which was introduced by [21]. AES is a cipher mechanism with different key lengths including 128, 192 and 256 bits [22]. The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the cipher-text. The numbers of cycles of repetition are as follows:

- 10 cycles repetitions for 128-bit keys.
- 12 cycles repetitions for 192-bit keys.
- 14 cycles repetitions for 256-bit keys.

In this research, we will use all types of AES key lengths to secure Sensitive, Confidential, and Internal types of data.

4. RESULTS

4.1.1 Experiment Setting

Table 2. Query set details

Attributes	Quantity
Add	50
Select	50
Delete	50

This section aims to identify the parameters that will be used in the experiments. The dataset is stored in MySQL, using Java JDK 8.1. In addition, an extension called 'Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy' is used in order to enable the encryption process. For the experiments, a set of queries is prepared. This set includes multiple types of queries including 'Add', 'Delete' and 'Select'. Table 2 shows the details of these queries.

4.1.2 Encryption

For encryption, we use AES algorithms with its three different sizes of key for encryption, to differentiate between Sensitive, Confidential, and Internal data types using 256-bit, 192-bit, and 128-bit respectively.

Let query Q which is an '*Add'* query that consists of several attributes including id '*Sid'* first name '*Fname'*, last name '*Lname'*, Date of Birth '*DOB'*, and mother tongue '*mother_tounge*'. This query can be expressed as in Fig. 3.

```
INSERT INTO `uni`.`addstudent`
(`Sid`,`Fname',`Lname`, `Dob`
,`mother_tongue`) VALES ('456',
'Adam', 'Smith', '12-6-80',
```

Figure 3. Sample Add query

This query will undergo a classification process in order to determine which part is sensitive and which part is non-sensitive.

4.1.3 Classification

Table 3 depicts the process of classifying the attributes using the proposed classification-based approach.

Table 3. Query attribute classification

Attribute	Value	Class	Encrypted records
Sid	456	P	456
Fname	Adam	S	E982CE5AA9AE85 432AA521A03
Lname	Smith	S	B7117A99F291A3 D16ABFABE43
Dob	12-6-80	I	C34EWSD
Mother-tongue	English	C	DA73CAF7F233B3 15A3FB70A2B826 F036

As shown in Table 3, four classes have been given for the attributes stated as; Public (P), Internal (I), Sensitive (S) and Confidential (C). The first class appears to be 'non-sensitive' attribute, while the latter three classes considered being 'sensitive' attributes. In this regard, only the sensitive attributes will be encrypted. Hence, the query in Fig. 3 will be migrated to the server cloud provider as in Fig. 4.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
INSERT INTO `uni`.`addstudent`
('Sid', 'Fname',`Lname`, `Dob` ,
'mother_tongue') VALES ('456',
'E982CE5AA9AE85432AA521A03',
'B7117A99F291A3D16ABFABE43',
'C34EWSD',
'DA73CAF7F233B315A3FB70A2B826F036')
```

Figure 4.Encrypted Add query

4.1.4 Decryption

In fact, this feature relies on the previous two features where identifying the type of affix with the word length will facilitate the process of determining its tense. The query will be stored in the database as in Fig. 4 (i.e. encrypted query). In this manner, if the user chooses to select a query, a decryption process should take place.

Let query Q which is an '*Select'* query that inquire about multiple attributes including id '*Sid'*, first name '*Fname'*, last name '*Lname'* and Date of Birth '*DOB'* with specific condition. This query can be expressed as in Fig. 5.

```
SELECT `Sid`,`Lname, `Dob`
,`mother_tongue`
FROM `uni`.`addstudent`
```

Figure 5. Sample Select query

In this case, the select query in Fig. 5 contains sensitive data (i.e. 'Adam').

```
SELECT `Sid`,`Lname, `Dob`
,`mother_tongue`
FROM `uni`.`addstudent`
WHERE 'Fname' =
'E982CE5AA9AE8543269358BBAA521A03
```

Figure 6. Encrypted Select query

Therefore, an encryption will be applied in order to hide the sensitive data, which is shown in Fig. 6.

Table 4. Retrieved record

Encrypted record (server-side)			
Sid	**Lname**	**Dob**	**Mother_tongue**
456	B7117A99F2 91A39BB4D9 3D16	C34EW SD	DA73CAF7F23 3B315A3FB70 A2
Decryption (client-side)			
Sid	**Lname**	**Dob**	**Mother_tongue**
456	Smith	12-6-80	English

As shown in Fig. 6, the encrypted query will be migrated into the server in the cloud. Since the data stored in the database is encrypted, thus a decryption process will be applied at the client side in order to interpret the retrieved record. Table 4 depicts an example of both encrypted and decrypted retrieved record.

4.1.5 Evaluation

In order to evaluate the proposed classification-based approach, three experiments will be performed. Firstly, the query set will be executed using full encryption (i.e. encrypting all the data), secondly, the query set will be executed using partial encryption (i.e. using the proposed classification-based), and lastly, the query set will be executed straight away without any encryption. Then, the evaluation will be performed based on the execution time. Every query will be analyzed in terms of the time and the encryption overhead will be calculated.

The efficiency of the proposed method can be observed in which the time consumed to process a fully encrypted query is compared against the time consumed to process a partial encrypted query (i.e. produced by the proposed method). Both types of queries will also be compared with the basic unencrypted query, so that we will get the actual overhead of both fully and partial encryptions method.

4.1.6 Results

The results of applying the approaches of encryption including full encryption (i.e. without classifying sensitive data), partial encryption (i.e. using the proposed classification method), and no encryption is discussed in this section. Fig. 8(a) to 8(c) shows such results for *Add*, *Select* and *Delete* queries. As shown in Fig. 7 (a), the application of partial encryption using the proposed classification-based approach has outperformed the application of full encryption for the fifty 'Add' queries. As an average, the full encryption has consumed 2.97 seconds compared to 1.49 seconds consumed by partial encryption.

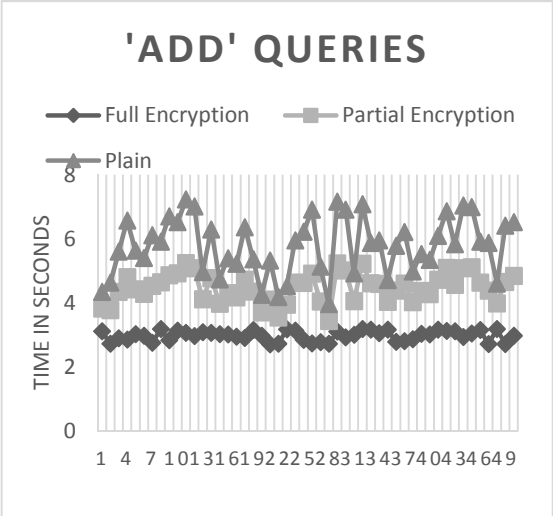
Also, in Fig. 7 (b), the application of partial encryption using the proposed classification-based approach has outperformed the application of full encryption for the fifty 'Select' queries. As an average, the full encryption has consumed 2.04 seconds compared to 0.79 seconds consumed by partial encryption.

For the Delete queries (Fig 7 (c)), the application of partial encryption using the proposed

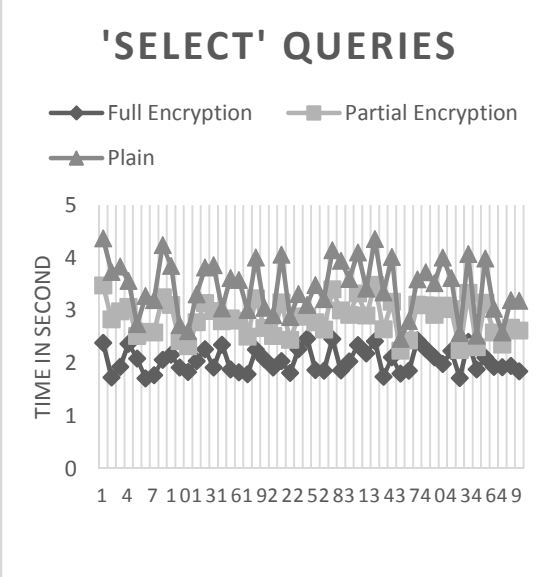
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

classification-based approach has outperformed the application of full encryption for the fifty 'Delete' queries. As an average, the full encryption has consumed 2.21 seconds compared to 0.64 seconds consumed by partial encryption.

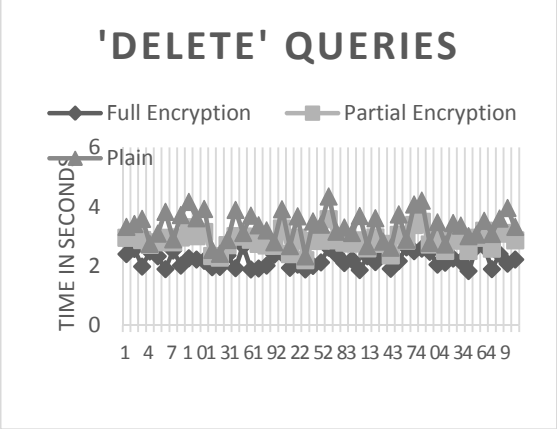
Such outperformance for the partial encryption implies the significance enhancement accomplished by the proposed method, in terms of processing time. This means that the proposed method can provide secure query processing in cloud without losing the efficiency criteria.



(a)



(b)



(c)

Figure 7(a) to (c). Experimental results

For all three settings, the execution of queries without encryptions has given the fastest execution times. Table 5 describes the execution time of the three queries set on the Full, Partial, and Plain (without encryption) in average. In addition, the percentage of encryption overhead is also calculated for full and partial encryption like the following:

$$Overhead = ((encryption - plain)/plain) * 100$$

The security overheads for each encryption scheme are obtained by calculating the additional time consumed by each scheme, as compared with the 'Plain' setting.

Table 5. Execution Time for Add, Select, and Delete

Queries	Average execution time (in seconds)			Encryption overhead (%)
	Add	Select	Delete	
Full encryption	2.97	2.04	2.21	65%
Partial encryption	1.49	0.79	0.64	18%
Plain	1.30	0.73	0.45	-

As shown in Table 5, we can see that in general, the proposed partial encryption has significant difference, when compared to the full encryption. For the overall time taken to finish the execution, we found out that the Add queries have taken the longest time to complete. This is because, the Add process requires inserting new records which consumes more time, compared to displaying existing record as in the Select query, or deleting existing record as in the Delete query. Basically, the proposed classification method has successfully

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

reduced the overall encryption overhead, three times lower compared to the full encryption method.

In general, the results have demonstrated a significant enhancement achieved by the proposed partial encryption. Apparently, the divide-and-conquer approach of classifying the sensitive data has improved the execution time of the queries. Comparing the results of the proposed method with other related work such as Zardari et al. [15] and Graepel et al. [5], the performance of the proposed method tend to be competitive in terms of efficiency.

5. CONCLUSION

This paper proposed a classification-based approach that aims to classify the sensitive data and perform encryptions before sending the data to the server. One of the limitations behind this study lies in the process of generating the rules. In this case, exploiting machine learning techniques, which depend on statistical model rather than predefined rules will be a great opportunity for future researches. One of the limitations behind this study lies on the process of generating the rules. In this case, exploiting machine learning techniques which depend on statistical model rather than predefined rules, would be a great opportunity for future researches.

ACKNOWLEDGEMENT

This paper has been funded by Universiti Kebangsaan Malaysia under research grant (DPP-2015-FTSM).

REFERENCES

[1] Huiqi Xu, Shumin Guo, and Keke Chen, "Building confidential and efficient query services in the cloud with rasp data perturbation," IEEE Transactions on Knowledge and Data Engineering, vol. 26, pp. 322-335, 2014.

[2] Yi Chen, Wei Wang, and Ziyang Liu, "Keyword-based search and exploration on databases," in 2011 IEEE 27th International Conference on Data Engineering, 2011, pp. 1380-1383.doi.

[3] Craig Gentry, "Fully homomorphic encryption using ideal lattices," in STOC, 2009, pp. 169-178.doi.

[4] Nuno Santos, Krishna P Gummadi, and Rodrigo Rodrigues, "Towards trusted cloud computing," in Proceedings of the 2009 conference on Hot topics in cloud computing, 2009, pp. 3-3.doi.

[5] Thore Graepel, Kristin Lauter, and Michael Naehrig, "ML confidential: Machine learning on encrypted data," in International Conference on Information Security and Cryptology, 2012, pp. 1-21.doi.

[6] Xiaojin Zhu and Andrew B Goldberg, "Introduction to semi-supervised learning," Synthesis lectures on artificial intelligence and machine learning, vol. 3, pp. 1-130, 2009.

[7] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels, "Dynamo: amazon's highly available key-value store," in ACM SIGOPS Operating Systems Review, 2007, pp. 205-220.doi.

[8] Brian F Cooper, Raghu Ramakrishnan, Utkarsh Srivastava, Adam Silberstein, Philip Bohannon, Hans-Arno Jacobsen, Nick Puz, Daniel Weaver, and Ramana Yerneni, "PNUTS: Yahoo!'s hosted data serving platform," Proceedings of the VLDB Endowment, vol. 1, pp. 1277-1288, 2008.

[9] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E Gruber, "Bigtable: A distributed storage system for structured data," ACM Transactions on Computer Systems (TOCS), vol. 26, p. 4, 2008.

[10] Donald Kossmann, Tim Kraska, Simon Loesing, Stephan Merkli, Raman Mittal, and Flavio Pfaffhauser, "Cloudy: A modular cloud storage system," Proceedings of the VLDB Endowment, vol. 3, pp. 1533-1536, 2010.

[11] David G Campbell, Gopal Kakivaya, and Nigel Ellis, "Extreme scale with full sql language support in microsoft sql azure," in Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, 2010, pp. 1021-1024.doi.

[12] Vanja Josifovski, Peter Schwarz, Laura Haas, and Eileen Lin, "Garlic: a new flavor of federated query processing for DB2," in Proceedings of the 2002 ACM SIGMOD international conference on Management of data, 2002, pp. 524-532.doi.

[13] Cong Wang, Ning Cao, Kui Ren, and Wenjing Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," IEEE Transactions on parallel and distributed systems, vol. 23, pp. 1467-1479, 2012.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[14] Yongjun Ren, Jiang Xu, Jin Wang, and Jeong-Uk Kim, "Designated-verifier provable data possession in public cloud storage," International Journal of Security and Its Applications, vol. 7, pp. 11-20, 2013.
[15] Munwar Ali Zardari, Low Tang Jung, and Muhamed Nording B Zakaria, "Hybrid Multi-cloud Data Security (HMCDS) Model and Data Classification," in Advanced Computer Science Applications and Technologies (ACSAT), 2013 International Conference on, 2013, pp. 166-171.doi.
[16] ISO17799. (2000). Establishing Information Classification Criteria. Newsletter. Available: http://17799-news.the-hamster.com/issue09-news1.htm
[17] EO12958. (1995). Classified National Security Information Available: http://www.fas.org/sgp/clinton/eo12958.html.
[18] SIGS. (2001). Security in the Government Sector. Available: http://www.gcsb.govt.nz/assets/GCSB-Documents/Security-in-the-Government-Sector-2002.pdf
[19] Columbia University. (2016). Data Classification Policy. Available: http://policylibrary.columbia.edu/information-security-charter
[20] Georgios Petasis, Frantz Vichot, Francis Wolinski, Georgios Paliouras, Vangelis Karkaletsis, and Constantine D Spyropoulos, "Using machine learning to maintain rule-based named-entity recognition and classification systems," in Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, 2001, pp. 426-433.doi.
[21] Joan Daemen and Vincent Rijmen, "AES proposal: Rijndael," 1999.
[22] Joan Daemen and Vincent Rijmen, The design of Rijndael: AES-the advanced encryption standard: Springer Science & Business Media, 2013.