A Fully Distributed Secure Approach for Database Security in Cloud Computing

Srinu Banothu, A. Govardhan, and Karnam Madhavi

Abstract Database-as-a-Service is one of the prime services provided by cloud computing. It is used to provide data storage and management services to the individuals, enterprises, and organizations on pay and use basis, in which any enterprise or organizations can outsource their database to the cloud service provider (CSP) and query the data whenever and wherever required through any devices connected to the Internet. The advantage of this service is that enterprises or organizations can reduce the cost of establishing and maintaining infrastructure locally. However, as cloud service providers are not trusted third parties, there exist some major challenges such as database security, privacy, and query performances to access data, and to overcome these issues, many authors contributed their work and developed various database security models. In this paper, we proposed a new model for cloud database security and we proved that our proposed model enhanced database security level and improved the query execution performance.

Keywords Cloud computing · Database-as-a-Service (DaaS) · Database security

1 Introduction

Cloud computing is a technology that provides various remote services on pay and use basis. The cloud services are broadly categorized into three types: (1) Software-as-a-Service(SaaS) (2) Platform as-a-Service(PaaS), and (3) Infrastructure as-a-Service(IaaS), the prime example of SaaS is Database-as-a-Service (DBaaS),

S. Banothu (✉)
JNTUH, VITS, Hyderabad, Telangana, India
e-mail: Srinub1307@gmai.com

A. Govardhan
CSE Department, JNTUH, Hyderabad, Telangana, India
e-mail: Govardhan_cse@jntuh.ac.in

K. Madhavi
CSE Department, GRIET, Hyderabad, Telangana, India

R. Buyya et al. (eds.), *Computational Intelligence and Data Analytics*,
Lecture Notes on Data Engineering and Communications Technologies 142,
https://doi.org/10.1007/978-981-19-3391-2_40

and it allows organizations and end users to easily outsource their databases and computations and access data whenever and wherever required through any device connected to Internet. DBaaS provides organizations with unlimited data storage services in a cost-effective way with higher availability and easy deployment. Nowadays, most of the organizations or individuals are outsourcing their databases to the cloud environment, and the amount of sensitive data stored in the cloud is increasing day by day; hence, it should be protected from malicious parties. It introduces new challenges regarding database security and privacy. The major threats to user data are (1) protecting data from external attackers and (2) protecting data from cloud service providers. Security and privacy to cloud database can be provided using 1. data distribution approaches and 2. data encryption techniques. Authors [1–17] contributed their work to protect cloud database from malicious attack, few of them used data encryption methods, and others used data distribution method. In this paper, we proposed a new model for cloud database security using combination of data encryption and data distribution approaches; the basic idea of our proposed method is initially all the tuples of a relation are encrypted using AES-CBC-256 algorithm [18] and a secret key; then, relation is partitioned vertically with selected columns into two or more fragments and store these fragments of tables into different databases of the same cloud server; to retrieve the tuple values, a query will be sent to all databases, processed on encrypted database tables; result returned to the user is in encrypted format; and user will decrypt the result using secret key. Our proposed method enhances security level of cloud database and reduces the query processing time to access the data from cloud database.

The structure of the paper discusses following concepts: Sect. 2 covers related work, Sect. 3 explains proposed model, Sect. 4 covers implementation, results, and performance tests, and Sect. 5 covers conclusion and future scope.

2 Related Work

In 2006, Evdokimov et al. [5] introduced a new security definition for database privacy homomorphism, and the idea behind this is construction of database privacy homomorphism based on searchable encryption scheme, in this scheme initially, create some words, those are strings of the same length and then identify the attributes of the relation. Then, bijectively convert the tuples of the given relation to the sets of words or documents. The number of words in each document is same as the number of the attributes in the relation. The globally fixed word length is equal to the length of an attribute identifier plus length of the longest attribute value. Then, documents are stored on a remote server by encrypting using a searchable encryption scheme. In order to apply exact select query on the encrypted relation, queries will be converted into the search operation and processed as a search operation, returns a set of encrypted strings. The strings are then decrypted and converted into the corresponding tuples. It is a generic construction for a database PH, and this can be

proved to be secure in a relaxed way, but still requires rigorous and plausible sense under widely accepted cryptographic assumptions.

In 2012, Liu and Wang et al. [7] contributed a work for secure query processing over encrypted database, and it is named as programmable order preserving indexing scheme. This scheme is built over simple linear expression of the form $a*x + b$, the form of expression is public, 'x' is the input value, and coefficients 'a' and 'b' are kept secret (not known to attackers). By using linear expression, the indexing scheme maps input value 'x' to $a*x + b +$ noise, where noise is a random value. If noise is carefully selected, then order of input values is preserved. This indexing scheme allows the programmability of basic indexing expressions, in which users can select different linear expressions for different input values for indexing input values. Programmability improves robustness of the scheme against brute force attacks since there are more indexing expressions. This scheme is used to process range queries over encrypted database, and it only depends on linear expression, so that it is easy to understand by the users. The problem with this scheme is more processing overhead as different linear expressions are used to create indexing for different input values.

Authors in [13] proposed a model for cloud database security in Database-as-a-Services, and it provides data privacy and security using the data distribution techniques instead of data encryption. This technique is used by existing netDB2 service. It is based on the multiple service providers and secret sharing algorithm, and the basic idea of secret sharing method is to distribute data to multiple servers to ensure the privacy of user queries. If user wants to outsource data from data source (D) to database service providers (DBS1, DBS2, ……, DBSn), data is partitioned into n shares and n shares will be stored in n DBS. If user wants to retrieve the data from DBS, query will be sent to all DBS and data received from all DBS will be merged and result will be sent to the user. To reconstruct secret value Vs at data source D, the knowledge of any K can refer to Vs besides some secret information X that is known only to the data source. Therefore, with the full knowledge of (K-1), DBS will not have any knowledge of Vs, even if X is known to them. In this model, data source (D) selects a random polynomial equation $q(x)$ of degree (K-1), where the constant is Vs. Each DBS has constant Vs and X which is a set of n random points. The problem with this model is availability of all DBS. If any one of the service provider server is down, data cannot be retrieved and another problem is computational complexity of n random polynomial equations for different n values. Another issue identified in this model is authors only considered numeric data for encryption and does not talk about non-numeric data.

In 2017, authors in [14] recently proposed a model for cloud database security to improve security level, and this model provides security to cloud database using combination of data distribution and data encryption techniques. This model uses two types of clouds, one is master cloud and another is slave cloud, the master cloud stores the entire database encrypted using some encryption algorithms, and slave cloud stores the extended columns (i.e., vertically fragmented columns) of relation. Keys are not revealed to master cloud service providers. In this model, when a relation for master cloud is created, one additional column is added for storing the indexes of each tuple in that relation. The index column stores the indexes of tuple in plain

text format, so that the user can query the relation through that index to access the desired tuples. The index is replicated in each fragment stored in slave cloud. Here, master cloud is private cloud, because it is available within the enterprise limits, it acts as a proxy server, and the task of proxy server is to create relations, insert, delete, encrypt, decrypt, and process the queries. The problem with this model is that since it maintains a private cloud locally in enterprise environment, same infrastructure has to be established and maintained locally as public cloud, so there is no benefit of cloud service reflected. Another issue is all slave cloud servers must always be available to retrieve the data by users. If any one of slave cloud servers are down, data cannot be retrieved, so loosing on demand cloud service. And also it increases query processing time to access the data as query has to be split forwarded to all the slave clouds.

In 2020, authors in [19] proposed a model for database security in cloud known as Proficient Security over Distributed Storage: a method for data transmission in cloud. Authors focused on issues of data security on multiple clouds. Proposed model partitioned data into two major types such as normal data and sensitive data; again, sensitive data is further partitioned into two parts. Every individual part is enciphered and distributed over multiple clouds, and normal data is placed over a single cloud in cipher text format. While decryption, sensitive data is retrieved from multiple clouds and combined. The proposed model is also tested against various attacks and also proved that model is resistant to pollution attack, known plain text attack, and key attacks. But still, this method suffers from availability property of information security.

3 Proposed Security Model

In our proposed model, three entities are involved: 1. data owner, 2. data user, and 3. cloud service provider (CSP) or cloud vendor. In this model, we used data encryption in combination with data distribution approach for secure data storage in cloud, first database relations are encrypted using symmetric encryption algorithms, and then, relations are vertically fragmented with selected columns of relations (i.e., column selection for table partition is based on data sensitivity level) and stored in cloud databases. Data owner creates two types of databases in cloud, one is master cloud database and another is two or more slave cloud databases, master cloud database is used to store the metadata information such as fragmented table data (i.e., name of the fragmented table, column names in fragmented table, slave database name where table fragment is stored), and cloud slave databases are used to store the fragmented tables. The metadata in master database is required for data owner for easy retrieval of data from cloud databases. For data encryption, we found AES-256-CBC [18] is the best algorithm for data confidentiality, it provides high security to the data with optimal database encryption time, and we used Order Preserving Encryption (OPE) scheme for executing range queries over encrypted cloud database and also used the hashing encryption scheme for equality condition checking; this model is

called as fully secure distributed approach (FSDA). The processing mechanism of my proposed model consists of three major modules known as end user module, application interface (API) module, and cloud storage:

- First, end users (data owner) select the database table to be outsourced from local system, select a secret key to encrypt the database tables, and send to API module to be outsourced to cloud.
- API module encrypts all the column values of a relation in a database using AES-256-CBC encryption algorithm and secret encryption key, this key is known only to the data owner, and it should be securely shared to the data users.
- API module also splits the encrypted database tables or relations vertically into two or more parts with selected columns by considering data sensitivity criteria and also adds index value in index column for each vertically fragmented relations, and this index value must be replicated in all fragments for the tuple of unpartitioned relation, so that the user can retrieve the tuple data easily and reduces the query execution time.
- Then uploads the vertically fragmented table in multiple slave database instances of cloud database servers of cloud service provider environment.
- API module also uploads the metadata information to the master database in cloud to know the locations of fragments stored in slave databases.
- In end user module, data owner must authenticate users to perform operations on cloud database; for authentication, users must register with data owner with their details.
- In end user module, data owner will share the user credentials with cloud service providers (CSP), so that in CSP verify the user credentials with credentials already shared by data owner to CSP, if user is valid, then accept the user query and returns the resultant data to the user.
- Data owner securely shares the secret key to users for data decryption, and the data user after retrieving the secret key from data owner performs some operations on databases like selection, insertion, deletion, and updation.
- Cloud storage environment stores the fragmented data uploaded by data owner in multiple database instances.
- To retrieve the data from cloud, data is retrieved from multiple slave database instances in encrypted format and merged and then decrypted using the secreted key.

As database relations are partitioned vertically and distributed into multiple databases that if attacker compromises the data in one fragment, cannot get the complete information. So our model provides more security to the data stored in cloud environment from trusted third party attack (i.e., insider attack) than method used in [14]. The cloud service providers will be unaware about the data stored in database because all attribute values of records in database relations are stored in cipher text format.

4 Implementation

A. *Cloud Computing Tools*

First, we created a public cloud computing account at cloud clusters.io, which provides open-source cloud computing service. We have created a MySQL server managed with phpMyAdmin for experimental purposes and then created one master database to store metadata and two slave databases for storing fragments of database relation. The configurations of servers created on cloud are as follows: 3(core) processors, 4 GB RAM, and 100 GB SSD. XAMPP server was installed on my local system to run my application on system configured with Intel core i5 processor, 10 GB RAM, and 360 GB hard disk space. Network speed is 150 mbps.

B. *Performance Evaluation*

We evaluated the SQL SELECT query execution performance in terms of time taken to access the data stored in cloud database and return the results to the user. Performance of our proposed method is compared with methods used in [14].

For experimental work, we have taken student's data and stored in cloud database, we have created one master database and two slave databases in cloud server and created two tables in master database, one table is to store student's records with field names id, username, password, first_name, last_name, email, and phone_no, and another table is to store metadata; we also created fragmented table in slave database1 with fields username, first_name, and lastname and also created another fragmented table in slave database2 with fields password, email, and phone_no. Then, 600 records of students are inserted into the cloud databases and evaluated the performances of SELECT query execution on four methods to retrieve the records with predicates and without predicates.

Table 1 shows the performance of SELECT query without predicates to retrieve all the records from cloud database, this query performance is measured on four methods, and proposed model's performance is compared with performances of models in [13, 14, 19], so our proposed model average query processing time is less than methods used in [13, 14, 19].

Table 1 Select query execution time in milliseconds to access all the records from cloud databases

Cloud database security methods	Fully distributed secure approach (FDSA)	[13, 14, 19]	Secure centralized approach (SCA)	Unsecure centralized approach (UCA)
1	3721.92	3060.72	2272.83	1235.77
2	3802.85	3169.87	2249.55	1257.1
3	3729.75	3666.42	2267.89	1273.61
4	3020.1	4470.15	2220.98	1244.06
5	3060.97	3969.29	2253.72	1235.31
Avg. query execution time	3467.118	3667.29	2252.994	1249.17

Table 2 Select query execution time in milliseconds to access selected records with two predicates connected with AND operator

Cloud database security methods	Fully distributed secure approach (FDSA)	[13, 14, 19]	Secure centralized approach (SCA)	Unsecure centralized approach (UCA)
1	258.93	262.94	265.58	251.36
2	256.23	249.45	249.71	261.84
3	256.66	252.25	258.46	254.80
4	257.47	275.48	252.90	250.95
5	256.36	275.71	250.23	255.54
Avg. query execution time (milliseconds)	257.13	263.17	255.376	254.898

Table 2 shows the performance of SELECT query with predicates to retrieve selected records from cloud database, this query performance is measured on four methods, and proposed model's performance is compared with performances of models in [13, 14, 19], so our proposed model average query processing time is less than the methods used in [13, 14, 19].

Figure 1 shows the graphical representation of query processing time to select the all records from cloud databases and return the results to the user; here, the experiment is performed on four methods and collected the data; and it shows the average query processing time of all four methods.

Figure 2 shows the query processing time to select the single record from cloud databases and return the results to the user; here, the experiment is performed on four methods and collected the data; and it shows the average query processing time of all four methods and proposed method average query processing time less than distributed secure approach [13, 14, 19].

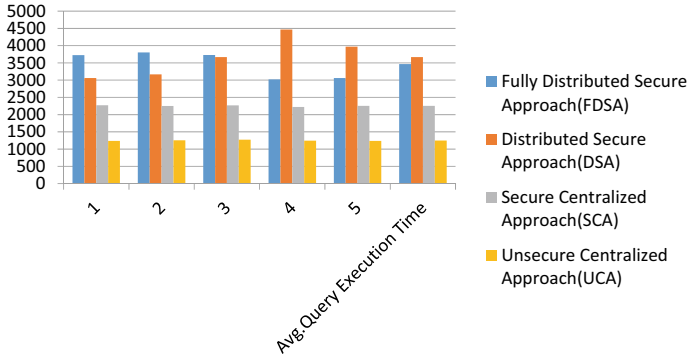

Fig. 1 Select query execution time in milliseconds to access all the records from cloud databases

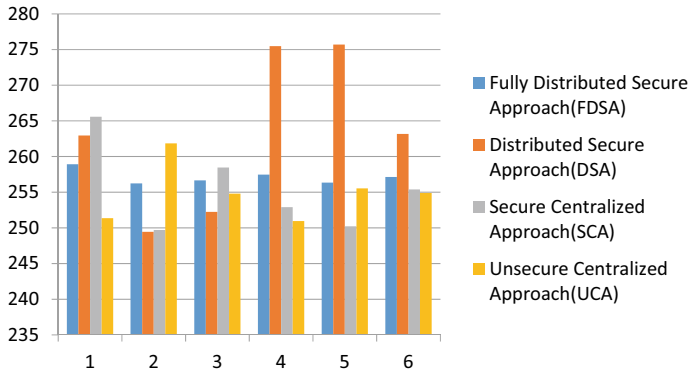

Fig. 2 Select query execution time in milliseconds to access selected records with two predicates connected with AND operator

5 Conclusion and Future Scope

Database-as-a-Service is one of the cloud computing services providing to individuals or organizations on pay and use basis. It provides data storage and management services through Internet. The aim of the data storage and management service is any individual or organizations can outsource their data to the cloud environment and access the data whenever and wherever required through Internet-connected devices. As the cloud service providers are not trusted parties, they may leak the user's or organization's sensitive data or data may be stolen by attackers. To overcome these issues, we have proposed a new model named as fully distributed secure approach (FDSA) for cloud database security. In this paper, we have discussed about related works and implemented our proposed model using PHP and run on XAMPP tool on local machine, created a database server on open-source cloud service provider cloud cluster, and evaluated the performance of SELECT query execution with simple predicates and complex predicates on encrypted cloud databases. In our research, we have compared the performance of our model with methods used in [13, 14, 19] with experimental work and found our model is more secure with less query execution time. And our future research is to enhance model for executing range queries on encrypted cloud databases and evaluate the performances.

References

1. Rivest RL, Adleman L, Dertouzos ML, On data banks and privacy homomorphisms. Massachusetts Institute of Technology Cambridge, Massachusetts Copyright © 1978 by Academic Press, Inc
2. Davida GI, Wells DL (1981) A database encryption system with subkeys. ACM Trans Database Syst 6(2):312–328

3. Hacigümüş H, Iyer B, Li C, Mehrotra S (2002) Executing SQL over encrypted data in the database-service-provider model, ACM SIGMOD 2002 June 4–6, Madison, Wisconsin, USA Copyright, ACM 1-58113-497-5/02/06 ...}5.00
4. Bertino E, Sandhu R, (Jan–March 2005) Database security—concepts, approaches, and challenges. IEEE Trans Dependable Secure Comput 2(1)
5. Evdokimov S, Fischmann M, Gunther O. Provable security for outsourcing database operations. Proceedings of the 22nd international conference on data engineering (ICDE'06) 8-7695-2570-9/06 $20.00 © 2006 IEEE
6. Evdokimov S, Günther O (2007) Encryption techniques for secure database outsourcing, published. In: Biskup J, Lopez J (eds) ESORICS 2007. LNCS, vol 4734. Springer, Heidelberg. http://www.springerlink.com/content/978-3-540-74834-2/
7. Liu D, Wang S∗. Programmable order-preserving secure index for encrypted database query, 2012 IEEE fifth international conference on cloud computing, 978-0-7695-4755-8/12 $26.00 © 2012 IEEE
8. Liu D, Wang S∗. DEMO: query encrypted databases practically, CCS'12, October 16–18, 2012, Raleigh, North Carolina, USA. ACM 978-1-4503-1651-4/12/10
9. Xu L, Wu X (2013) Hub: Heterogeneous bucketization for database outsourcing, cloud computing' 13, May 8, 2013, Hangzhou, China. Copyright ACM 978-1-4503-2067-2/13/05 ...$15.0
10. Ferretti L, Colajanni M, Marchett M (Feb 2014) Distributed, concurrent, and independent access to encrypted cloud databases. IEEE Trans Parallel Distrib Syst 25(2)
11. Li J, Yao W, Zhang Y, Qian H, Han J (Sept–Oct 2017) Flexible and fine-grained attribute-based data storage in cloud computing. IEEE Trans Serv Comput 10(5)
12. Guo C, Zhuang R, Jie Y, Ren Y, Wu T, Choo K-KR (2016) Fine-grained database field search using attribute-based encryption for e-healthcare clouds. J Med Syst 40:235. https://doi.org/10.1007/s10916-016-0588-0
13. ALzain MA, Pardede E (2011) Using multi shares for ensuring privacy in database-as-a-service. Proceedings of the 44th Hawaii international conference on system sciences, 1530–1605/11 $26.00 © 2011 IEEE,pgno:1–9
14. Alsirhani A, Bodorik P, Sampalli S, Improving database security in cloud computing by fragmentation of data, 2017 International conference on computer and applications (ICCA), 978-1-5386-2752-5/17/$31.00 2017 IEEE
15. Gahia Y*, El Alaoui I (2019) A secure multi-user database-as-a-service approach for cloud computing privacy. Procedia Computer Science 160:811–818. International workshop on emerging networks and communications (IWENC 2019) November 4–7, 2019, Coimbra, Portugal, ScienceDirect Available online at www.sciencedirect.com
16. Madhavi K, Ramesh G, Sowmya K (2019) CICIT 630–636
17. Banothu S, Govardhan A, Madhavi K. Performance evaluation of cloud database security algorithms, W3S web of conference (ICMED 2021)
18. Banothu S, Govardhan A, Madhavi K (2021) Performance comparison of cryptographic algorithms for data security in cloud computing. J Inf Comput Sci 11(9):1–8. ISSN: 1548-7741
19. Shahid F, Ashraf H, Ghani A, Ghayyur SAK, Shamshirband S, Salwana E (2020) PSDS–proficient security over distributed storage: a method for data transmission in cloud. IEEE Access 8:118285–118295