International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 4, Issue 3 (May-June, 2016), PP. 262-267

AN EFFICIENT ENCRYPTION AND SEARCHING TECHNIQUE FOR CLOUD USING RIJNDAEL ALGORITHM

[1]Sanidhya U, [2]Shrikanth N.G
[1]MTECH in Computer Science,
[2]Asst. Prof. , Department of Computer Science
Shree Devi Institute Of Technology,Kenjar,Mangalore,Karnataka
[1]Sanidhya468@gmail.com
[2]shrikanth109.ng@gmail.com

Abstract— **For an Internet based applications, a new service called as the cloud database is required. Data encryption, which is optimum solution for achieving the confidentiality. In the cloud database, all the information will be encrypted for the purpose of achieving security. This will improve the confidentiality of the information that stored in the cloud database. In this paper, we are presenting an architecture for the adaptive encryption of a public cloud database. In the paper confidentiality attained where the client will generate secret key and not stored anywhere on the cloud. In native method, encryption of the whole database through standard encryption algorithm which will not allow any of SQL operations directly on the cloud. This had affected the workload and the cost which made cloud computing services inconvenient. The novel cloud architecture uses the adaptive encryption technique not using intermediate servers or proxies. This scheme provides the cloud provider with the best level of confidentiality for any of the database workload.**

Index terms- **Adaptive Encryption Technique, Encrypted Metadata, Encrypted Data, Advanced Encryption Standard (AES), cloud,**

I. INTRODUCTION

The Cloud computing is limited by concern about the information confidentiality and over a medium and long term. There are many of research challenges in database service in term of security. Some of encryption schemes are applicable to database paradigm. Other encryption schemes which will allow execution of SQL operation over the encrypted data, either suffer from the performance limit or they requires the choice of which encryption scheme must become adaptive for each database column and the SQL operations.

In this paper, we are proposing an architecture for an adaptive encryption of a public cloud database which offers a proxy free alternative. Even when a set of SQL queries change, the architecture proposed here guarantees in an adaptive way, the best level of data confidentiality for any of database workload. Older adaptive encryption schemes which wasn't referring to the cloud, encrypts each plain column into the multiple encrypted column and each value is then encapsulated in to the different layers of the encryption, so that outer layer will guarantee higher confidentiality, but support only fewer computation capability with respect to its inner layers. The outer layer is dynamically adopted at its run time when new SQL operations is added to workload. In this particular architecture, it does not require to define which database operations are allowed on each of column , it poses the novel issue in terms of the feasibility in the cloud context and storage and network cost estimations.

The first we are implementing a proxy free architecture for an adaptive encryption of the cloud database. The concurrent and distributed client can now issue parallel operations without passing through the same centralized component as in a alternative architecture. So that there is no limit in the availability, elasticity and scalability of the plain cloud database.

1. The first we implement proxy free architecture for
2. adaptive
3. encryption of cloud database. The
4. concurrent clients can issue parallel operation
5. without passing through, same centralized component
6. as in alternative architecture. So that there is not limit
7. an availability elasticity and scalability of a plai

The first we implement proxy free architecture for adaptive
encryption of cloud database. The
concurrent clients can issue parallel operation
without passing through, same centralized component
as in alternative architecture. So that there is not limit
an availability elasticity and scalability of a plain

II. LITERATURE SURVEY

This[1] defines Cloud computing and provide the architecture for creating Clouds with market-oriented resource allocation by leveraging technologies such as Virtual Machines (VMs). It also provide insights on market-based resource management strategies that encompass both customer-driven

service management and computational risk management to sustain Service Level Agreement (SLA)-oriented resource allocation. In addition, thoughts on interconnecting Clouds for dynamically creating global Cloud exchanges and markets. Highlighting the difference between High Performance Computing (HPC) workload and Internet-based services workload. Also describe a meta negotiation infrastructure to establish global Cloud exchanges and markets, and illustrate a case study of harnessing `Storage Clouds' for high performance content delivery.

This[2] explores a new paradigm for data management in which a third party service provider hosts ”database as a service” providing its customers seamless mechanisms to create, store, and access their databases at the host site. Such a model alleviates the need for organizations to purchase expensive hardware and software, deal with software upgrades, and hire professionals for administrative and maintenance tasks which are taken over by the service provider. The authors of this paper have developed and deployed a database service on the Internet, called NetDB2, which is in constant use.

In a sense, data management model supported by NetDB2 provides an effective mechanism for organizations to purchase data management as a service, thereby freeing them to concentrate on their core businesses.

This[3] referred says that, Utility grids such as the Amazon EC2 cloud and Amazon S3 offer computational and storage resources that can be used on-demand for a fee by compute and data-intensive applications. The cost of running an application on such a cloud depends on the compute, storage and communication resources it will provision and consume. Different execution plans of the same application may result in significantly different costs. Using the Amazon cloud fee structure and a real-life astronomy application, we study via simulation the cost performance tradeoffs of different execution and resource provisioning plans. We also study these trade-offs in the context of the storage and communication fees of Amazon S3 when used for long term application data archival. The results show that by provisioning the right amount of storage and compute resources, cost can be significantly reduced with no significant impact on application performance.

This[4] proposes a fully homomorphic encryption scheme – i.e., a scheme that allows one to evaluate circuits over encrypted data without being able to decrypt. Our solution comes in three steps. First, we provide a general result – that, to construct an encryption scheme that permits evaluation of arbitrary circuits, it suffices to construct an encryption scheme that can evaluate its own decryption circuit. Next, describing a public key encryption scheme using ideal lattices that is almost bootstrappable000000. Lattice-based cryptosystems typically have decryption algorithms with low circuit complexity, often dominated by an inner product computation that is in NC1. Also, ideal lattices provide both additive and multiplicative homomorphism (modulo a public-key ideal in a polynomial ring that is represented as a lattice), as needed to evaluate general circuits.

This[5] says that, Cloud computing, as an emerging computing paradigm, enables users to remotely store their data into a cloud so as to enjoy scalable services on-demand. However, allowing cloud service providers (CSPs), which are not in the same trusted domains as enterprise users, to take care of confidential data, may raise potential security and privacy issues. To keep the sensitive user data confidential against untrusted CSPs, a natural way is to apply cryptographic approaches, by disclosing decryption keys only to authorized users. In this paper, a scheme is proposed to help enterprises to efficiently share confidential data on cloud servers. This[6] says, Online applications are vulnerable to theft of sensitive information because adversaries can exploit software bugs to gain access to private data, and because curious or malicious administrators may capture and leak data. CryptDB is a system that provides practical and provable confidentiality in the face of these attacks for applications backed by SQL databases. It works by executing SQL queries over encrypted data using a collection of efficient SQL-aware encryption schemes. CryptDB can also chain encryption keys to user passwords, so that a data item can be decrypted only by using the password of one of the users with access to that data. As a result, a database administrator never gets access to decrypted data, and even if all servers are compromised, an adversary cannot decrypt the data of any user who is not logged in. This[7] says that rapid advances in networking and internet technologies have fueled the emergence of the “software as a service” model for enterprise computing. There are two main privacy issues. First, The owner of the data needs to be assured that the data stored on the service-provider sites is protected against the data thefts from outsiders. Second ,the data needs to be even protected from the service providers if the providers themselves can’t be trusted. This paper focus on the second challenge. Paper explores techniques to execute sql queries over the encrypted data. Strategy is to process as much query as possible at the service providers site without having to decrypt the data. Decryption and the remainder of the query processing are done at the client side. The paper explores algebriac framework to split the query to minimize the computation at the client side.

III. A PROPOSED SYSTEM ARCHITECTURE

In the system, the distributed and the concurrent clients can issue direct SQL operation .This proposed solution guarantees same level of availability and the scalability of a cloud service, by avoiding the architecture based on the one or more multiple proxies or intermediate server.

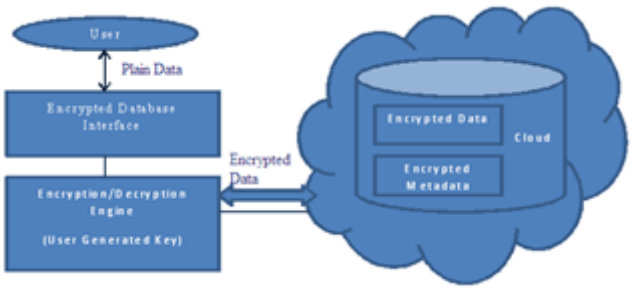

Fig:(3): Encrypted cloud architecture

Each of client will then execute encryption engine which manages the encryption operation. This particular software module accessed by the external user application via the encrypted database interface, five type of information are managed by the architecture.

1) Plain data is the tenants information.

2) Encrypted data are stored in a cloud database

3) Plain metadata is an additional information necessary to execute the SQL operation on the encrypted data.

4) Encrypted metadata is the encrypted version of a metadata, that are stored in cloud database.

5) Master key is an encryption key of a encrypted data that is generated by the client.

In the cloud database here , all the data and metadata are in encrypted form. Any of the application running on client will issue a SQL operation to the encrypted cloud database through an 'encrypted cloud database interface'. Data is transferred between a user application and an encryption engine that are in the plain format. Informations are always encrypted before sending it to the cloud database. When the application issues a new SQL operation, the encrypted database interface contacts the encryption engine which retrieves the encrypted metadata and then decrypted it using master key. The plain metadata are then cached locally by client as a volatile information (to increase the performance). After getting metadata, encryption engine is able to execute SQL operation over encrypted data. Then it decrypts the result .The results are then returned to the user application through an encrypted database interface .

A. An Adaptive Encryption Scheme

SQL-aware-encryption algorithm which guarantees the data confidentiality and allows the cloud database server for executing SQL operations over an encrypted data. Adaptive encryption scheme that preserves maximum level of the data confidentiality on a column that are not involved in any of the operation. Encryption algorithm are organized into a structure called as "onion". Each onion is composed by an ordered set of the encryption algorithms called 'encryption layer'. Outer layer quarantine is a higher level of data confidentiality and it allows the less type of operation on an encrypted data. Each of onion supports a specific set of operation. Each plaintext column is converted into one or more encrypted column, each one corresponds to the onion. Each plaintext value is then encrypted though all the layers of the onion. Actual layer is the most external layer of a onion and it correspond to the strongest encryption algorithm. Cloud database can see actual layer of onion and it has no access to a inner layer and the plain text data.

B. A Proposed Algorithm

It uses a symmetric key algorithm. AES uses a 128, 192, or 256-bit length of keys. Adopted by a National Institute of the Standards and Technology (NIST) on May26, 2002. AES uses the algorithm called Rijndael algorithm which is developed by the Joan Daemen and Vincent Rijmen of Belgium. AES is the simple design, high speed algorithm, with the low memory cost. AES is the symmetric block cipher. The same key will be used to encrypt and decrypt the messages. The plain text and cipher text are of same size. AES is restricted to the use of a block size of a 128 bits width .AES uses a permutation-substitution method which involves a series of substitution and the permutation steps to create the encrypted block. AES encryption on other hand is not breakable though there are few theoretical discussions about breaking AES .

IV. IMPLEMENTATION OF PROPOSED SYSTEM

A. Modules

- User applications
- Client-Side encryption engine
- Client Encrypted Database Interface
- Cloud database engine
- Cloud authentication and connection service.

B. Description Of Module

- User Application

First, all the clients have to select the data. The data useful for the client in the cloud. The original data not transferred directly to server. Before that a client have to encrypt data and then transfer to the server. Because of this ,the data is very secure. The metadata is also created. Using metadata the server can verify the data easily.

- Client-Side Encryption Engine

Sending an original data to server is vulnerable to the attacks. To avoid the problem, the original data should be encrypted. Client can't send original metadata to server and metadata also encrypted for the purpose of security. This process done in the client side encryption engine only.

- Client Encrypted Database Interface

This encrypted original data then are transferred to server. And then encrypted metadata are transferred to server. Using this metadata the client request is processed on the server side. These data are stored in the data base. The searching process is then done over the encrypted metadata. Metadata is used to get the encrypted original data for the client.

- Cloud Database Engine

The client request generated on the client side. And this request is transferred to server. The server search for related data by the query. This searching done by the encrypted metadata. Using the meta data, original data is referred on the server side. And then the transfer of the encrypted metadata to the client, who sent the request to the server. Using encrypted data the client will decrypt and will get the original data.

- Cloud Authentication and Connection Services

The client receive the data from the server. And uses the key for decrypting the data. The original data is visible to the client only.

client only. Server transfers the data to the client. But client does not know about the original data. Using the security and the metadata the data is transferred to the client.

V. EXPERIMENTAL RESULTS

Fig. (5a) Selecting client file

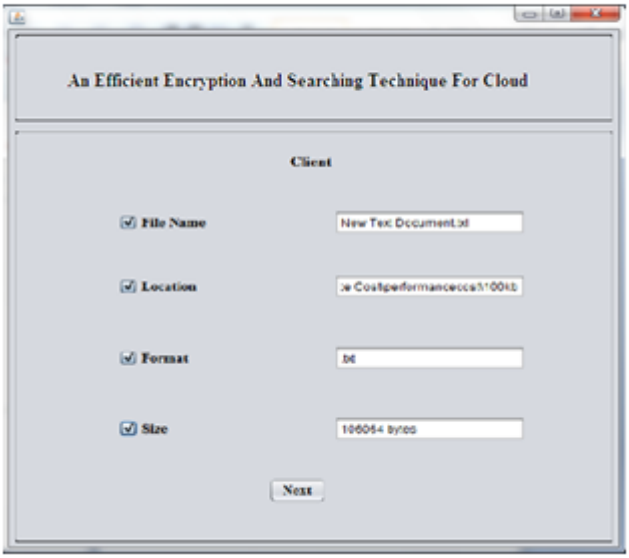

Fig. (5b). Details of file

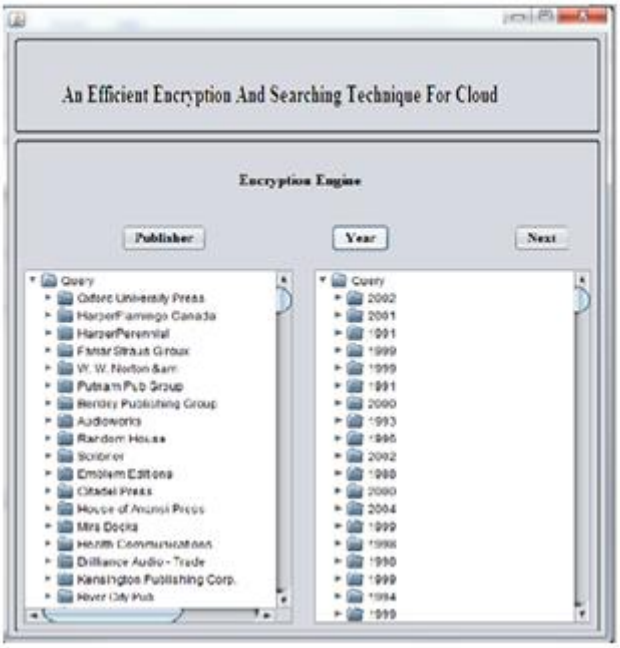

Fig. (5c) Metadata Created for Publisher, Year

Fig. (5e) Ciphering of Meta Data

Fig. (5g) Query to the cloud

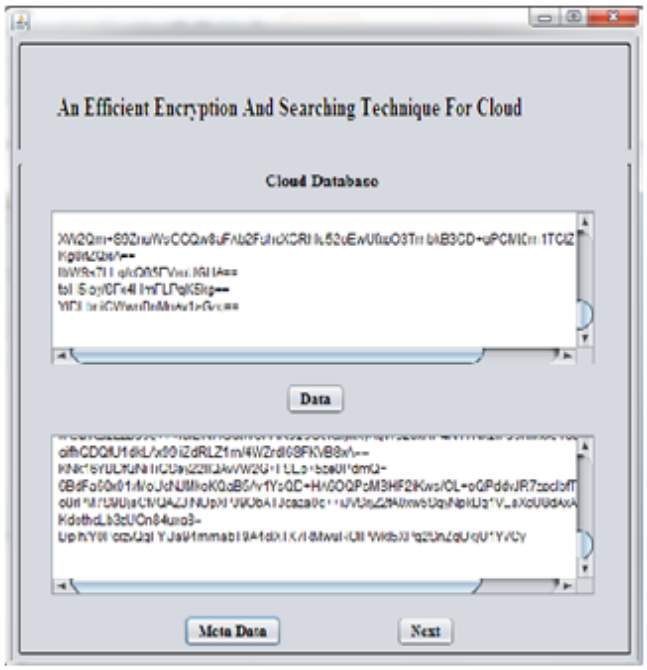

Fig. (5f) Upload to cloud

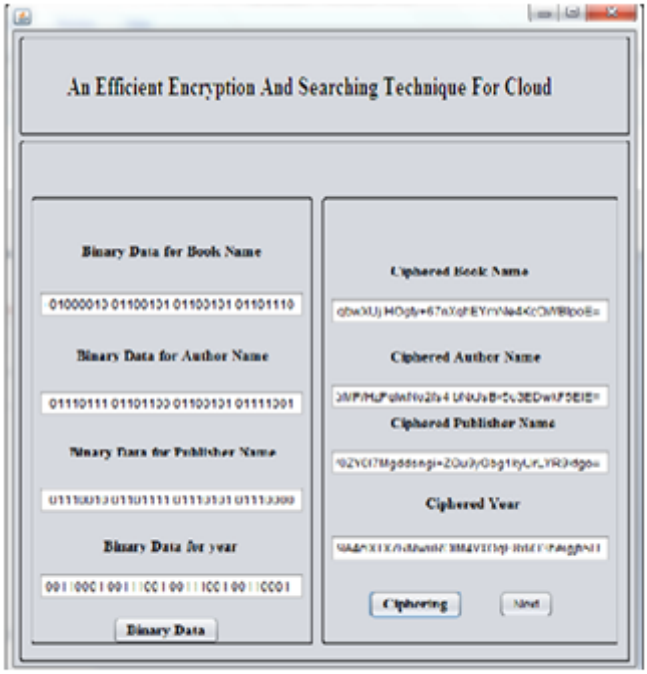

Fig (5h) Ciphering Query

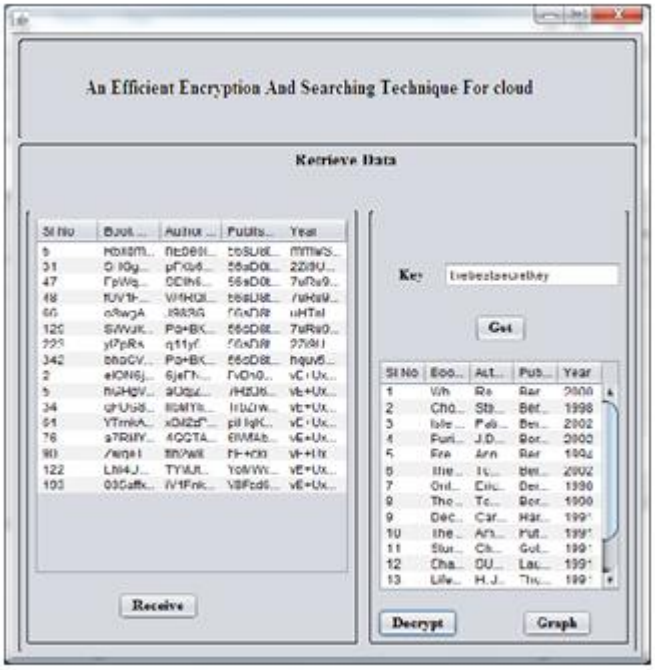

Fig.(5i) Receiving and decrypting data

VI. CONCLUSION AND FUTURE WORK

Data confidentiality is very important thing when comes to sharing. The data must be encrypted on a client side using the user generated key and transferred the encrypted data to the server. Then the client sends request to the server. Server will process the request and transfers it to the client. Then client uses the same key and decrypts the data**.**

Improvements in the algorithms can be done to improve the performance of encryption and decryption process. Private key management server can be added to make key management process easier.

VII. ACKNOWLEDGEMENT

This work has been supported in the part by Shrikanth n.g, Asst. Prof. and Guide of the Computer science and engineering Department, for his guidance through out my paper work by innumerable act of timely advice and encouragement. I am very much thankful to all of the teaching and the non-teaching staffs for allowing me to successfully carry out with amy project work.

REFERENCES

[1] R. Buyya, C, S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, 'Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility,' Future Generation Computer Systems, vol. 25, no. 6, pp. 599–616, 2009.

[2] T. Mather, S. Kumaraswamy, and S. Latif, 'Cloud security and privacy: an enterprise perspective on risks and compliance.' O'ReillyMedia, Incorporated, 2009.

[3] H.-L. Truong and S. Dustdar, 'Composable cost estimation and monitoring for computational applications in cloud computing environments,' Procedia Computer Science, vol. 1, no. 1, pp. 2175 – 2184, 2010, iCCS 2010.

[4] E.Deelman, G.Singh, M.Livny, B.Berriman, and J. Good, 'The cost of doing science on the cloud: the montage example,' in Proc.2008 ACM/IEEE Conf. Supercomputing, ser. SC '08. Piscataway, NJ,USA: IEEE Press, 2008, pp. 50:1–50:12.

[5] H.Hacig¨um¨us¸, B.Iyer, and S.Mehrotra, "Providing database as a service,' in Proc. 18th IEEE Int'l Conf. Data Engineering, Feb. 2002.

[6] G. Wang, Q. Liu, and J. Wu, 'Hierarchical attribute based encryption for fine-grained access control in cloud storage services,' inProc. 17th ACM Conf. Computer and communications security. ACM,2010, pp. 735–737.

[7] H.Hacig¨um¨us¸, B.Iyer, C.Li, and S.Mehrotra, 'Executing sql over encrypted data in the database service-provider model,'in Proc.ACM SIGMOD Int'l Conf. Management of data, June 2002.

[8] L. Ferretti, M.Colajanni, and M.Marchetti, 'Distributed, concurrent, and independent access to encrypted cloud databases,' IEEE Trans. Parallel and Distributed Systems, vol. 25, no. 2, Feb. 2014.