Dynamic Key Generation over Encrypted Cloud Data Using Genetic Functions

M.V.kishore[1], Ch.D.Naidu[2], Prof. Ch.Suresh[3]
[1]*Assistant Professor,* [2]*Sr.Assistant Professor,* [3]*Professor,*
Department of Information Technology,
[123]*Anil Neerukonda Institute of Technology & Sciences – (ANITS)*
Sangivalasa – 531162, Bheemunipatnam (Mandal), Vishakapatnam (District) Andhra Pradesh, India

Abstract- A general approach to protect the data confidentiality is to encrypt the data before outsourcing. However, this will cause a huge cost in terms of data usability. For example, the existing techniques on index-based retrieval cannot be directly applied on the encrypted data. Downloading all the data from the server and decrypt locally is obviously impractical. Hence, more practical special purpose solutions, such as encryption schemes have made specific contributions in terms of efficiency, functionality and security.
This system proposes a secure and index-based scheme over the encrypted cloud data, which supports dynamic key generation. Dynamic key and indexes, the proposed scheme can flexibly achieve in preserving privacy and deal with the deletion and insertion of documents in a more secured way. The secure AES algorithm is utilized to encrypt the data. To identify attacks while downloading, we provide the hackers information.

Keywords- Encryption, AES, confidentiality.

I. INTRODUCTION

CLOUD computing technology is a service-based, Internet-centric, safe, convenient data storage and network computing service. It is an internet-based model for enabling a convenient and on-demand network access to a shared pool of configurable computing resources. Cloud computing is the use of resources that are delivered as a service over a network. The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. The cloud providers manage the infrastructure and platforms on which the applications run. End users access cloud-based applications through web browser or a lightweight desktop or mobile app while the business software and user's data are stored on servers at a remote location. Cloud computing allows enterprises to get their applications run faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.

A general approach to protect the data confidentiality is to encrypt the data before outsourcing. However, this will cause a huge cost in terms of data usability. For example, the existing techniques on index-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical. In order to address the above problem, researchers have designed some general-purpose solutions with fully-homomorphic encryption or oblivious RAMs. However, these methods are not practical due to their high computational overhead for both the cloud sever and user. On the contrary, more practical special purpose solutions, such as searchable encryption schemes have made specific contributions in terms of efficiency, functionality and security. Searchable encryption schemes enable the client to store the encrypted data to the cloud and execute index search over ciphertext domain. Multi-indexed and dynamic key generation scheme achieves more and more attention for its practical applicability. Now some dynamic schemes have been proposed to support inserting and deleting operations on document collection. These are significant works as it is highly possible that the data owners need to update their data on the cloud server.
This paper proposes a secure-key and index-based scheme over the encrypted cloud data, which supports dynamic operation on the document collection. The secure AES algorithm is utilized to encrypt and decrypt the cloud data. To resist different attacks in different threat models, we construct two level security ie., by providing a dynamic key along with the indexes of the file.

Characteristics And Services Models:
The salient characteristics of cloud computing based on the definitions provided
National Institute of Standards and Terminology (NIST) are outlined below:

On-Demand Self-Service:A consumer can unilaterally provision computing capabilities,such as server time and network storage, as needed automatically without requiring human interaction with each service**'**s provider.

Broad Network Access: Capabilities are available over the network and accessed throughstandard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

Resource Pooling:The provider's computing resources are pooled to serve multipleconsumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center).

Rapid Elasticity: Capabilities can be rapidly and elastically provisioned, in some casesautomatically, to quickly scale out and rapidly released to quickly scale in.

Measured Service:Cloud systems automatically control and optimize resource use byleveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

Due to the increasing popularity of cloud computing, more and more data owners are motivated to outsource their data to cloud servers for great convenience and reduced cost in data management. However, sensitive data should be encrypted before outsourcing for privacy requirements..

Firstly, all the users usually keep the same secure key in symmetric schemes. Generally symmetric schemes assume that all the data users are trustworthy. But, a dishonest data user leads to many security issues.For each file request , user without pre-knowledge of the encrypted cloud data have to go through every retrieved file in order to find ones most matching their interest which demands possibly large amount of post processing overhead.

II. MATERIALS AND METHODS

For our system, we choose the B-tree as indexing data structure to identify the match between search query and data documents. Specially, we use inner data correspondence, i.e., the number of query keywords appearing in document, to evaluate the similarity of that document to the search query. Each document is uploaded to the cloud server along with the indexes and encrypted using AES. Whenever user wants to search, data user creates indexes for the files. Our aim is to design and analyse the performance of multi-indexed and dynamic key generation security scheme using AES algorithm for encrypting the cloud data. We designed a scheme based on secure multi-index over encrypted cloud data. Further, we analyzed its performance over false data users trying to access the secured data and are displayed in the hackers list. We have used DriveHQ platform to emulate the proposed system and to study its performance.

To design an efficient multi-indexed encryption scheme based on symmetric encryption, we included the following modules.

III. ENCRYPTION MODULE

By using AES, data in a file can be updated dynamically without affecting the overall performance. There is no need of re-encrypting the files in the database whenever the file is modified. This is a desirable feature as it reduces the computation time. Data owner first generates secret and public key pair (EK, DK) using a standard public-key encryption scheme i.e., RSA. Then owner makes the public key DK public and keeps the secret keys EK private.

Documents {D | D1, D2,…, Dn} are encrypted using EK resulting in a ciphertexts {C | C1,C2,….Cn}. The generated C is stored in cloud database.. This results in a set of

encryptions {e| e1,e2,..en} where each ej (for) is defined as E_Dj = AES_Enc (EK, Dj).

AES:

The Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S.National Institute of Standards and Technology (NIST) in 2001[8]

AES is a block cipher, but it does not use a Feistel structure. The block size of AES is 128-bit, but the key size may differ as 128, 192, or 256 bits [9].

Substitution: This method substitutes each byte of the block in the order of S-box. It provides an invertible transformation of blocks during encryption, with the reverse during decryption.

Shifting Rows: This operation performs left circular shifts of rows 1, 2, and 3 by 1, 2 and 3,

Mix Columns: This method multiplies each column of the input block with a matrix. The multiplication operation is just like matrix multiplication, except that it uses a Finite Field to multiply two elements and performs an XOR operation instead of addition.

Add Rounded Keys: This operation just applies an XOR operation to each byte of the input block and the current weight (key) matrix.

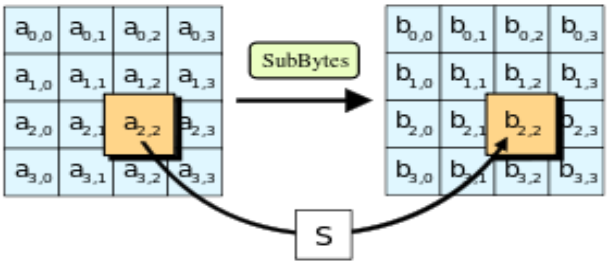


Fig.1: The Sub-Bytes step, one of four stages in a round of AES

The proposed algorithm consists of two steps i.e. random number generator and encryption

STEP-1 GENERATING RANDOM NUMBERS WITH THE HELP OF LINEAR METHOD AND GENETIC

FUNCTIONS, i.e., CROSSOVER AND MUTATION RESPECTIVELY:

Assumption about the generation of random numbers: -

1. Representations of the numbers are in Binary.

2. Population Size is fixed to 10.

3. Hence 5 generations are required to generate 50 numbers if the number of block of characters is 50.

LINEAR METHOD:
The first generation is created with the help of linear method and its equation is given below: - The sequence of random numbers is obtained via the following iterative equation.

Xn+1= (a*Xn+c) mod m

Where 1.Xn is the seed value or it is the value of the first Chromosome of the first generation.

2. m = modulus (m > 0).

3. a = multiplier (0 <= a < m).

4. c=increment (0 <= c < m).

Once the numbers of the first generation is created the next generation numbers are generated using the GA operators CROSSOVER and MUTATION.
After generation 1, the numbers of the next generation is obtained by CROSSOVER followed by MUTATION. The pairing up of numbers is done first, with the concept that for odd type generation pairing is done in one way and for even type generation in the opposite way. For example, after the first generation we got the following numbers:-
333, 6578, 8614, 5959, 7922, 8837, 4440, 903, 3693,2686. 2nd Generation: - Pairing up: - (333, 6578), (8614, 5959), (7922, 8837),(4440, 903), (3693, 2686). For this generation crossover and mutation will take place let at 6th locus of the gene of chromosome.

CROSSOVER:
Binary Representation of the first pair:

333 = 000010**1001101**

6578 = 110011**0110010**

Crossover: 000010**0110010** 110011**1001101**

MUTATION:
Mutation: 00001**1**0110010 11001**0**1001101

= 434 =6577

Similarly, the other pairs can also be generated in the following way. Now after generating all the numbers by applying crossover and mutation on each pair we get;
434,6577,263,5798,8069,9202,4478,816,3646,2605

After the second generation we continue with the 3rd, 4th and 5th generation to generate 50 numbers (Each generation 10 populations) and get the final set of numbers.

STEP-2 ENCRYPTION :
1. Once all the numbers are generated then let this array of numbers be called SUB_ARRAY and select the first digit of each number from SUB_ARRAY and a new collection of numbers is generated and let this collection is called COLLECTION_ARRAY.
2. Use this numbers from COLLECTION_ARRAY sequentially for substituting on a one-to-one basis for the characters of the plain text
Use ASCII values of the plain text characters and subtract the numbers of COLLECTION_ARRAY from the ASCII values. For example the message "SOUMYA" the CIPHER TEXT will be calculated according to following method. LET SUB_ARRAY = {4167, 10117, 5602, 4867, 4307,2452}

Encryption:

Character	ASCII Value	Collection_Array Number Taken sequentially	Subtract	Result
S	83	4	83 - 4	79
O	79	1	79 - 1	78
U	85	5	85 - 5	80
M	77	4	77 - 4	73
Y	89	4	89 - 4	85
A	65	2	65 - 2	63

The enciphered message is "RESULT"

The Cipher text is: {79, 78, 80, 73, 85, 63}

Table1:encryption table

DECRYPTION:

Result (R)	For loop I =0 to 255 Do (I – R) & Compare With Collection_array & Choose I.	Charter
79	83	S
78	79	O
80	85	U
73	77	M
85	89	Y
63	65	A

Table.2:decryption

Hacker Module:
Searching for a file from the cloud and accessing the data has to be secured and preserved from fraudulent users. Hence a mechanism is developed to catch hold of all these users. These users are caught when entering wrong secret keys for accessing a file. The server uses this secret key and indexes to

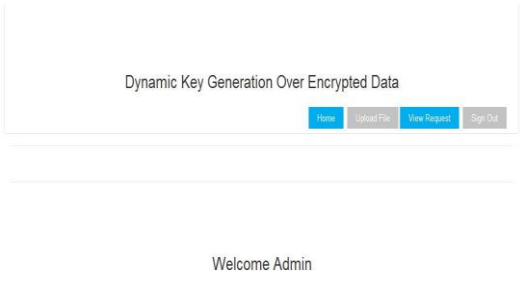
match the match with the entering indexes and keys for a file. If match found stores the pointer to that document in encrypted cloud else the data users are stored in the hackers list of the cloud server.

IV. EXPERIMENTAL RESULTS

1. Home Page:

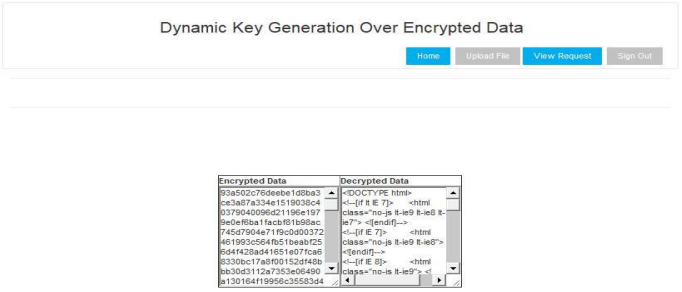
2.Data owner login:

3. Data owner or admin page after login:



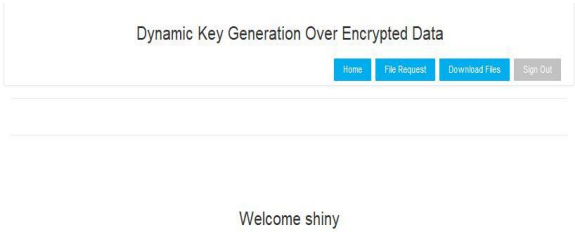
4. Uploading the files:

5. View encrypted and decrypted data after uploading a file:



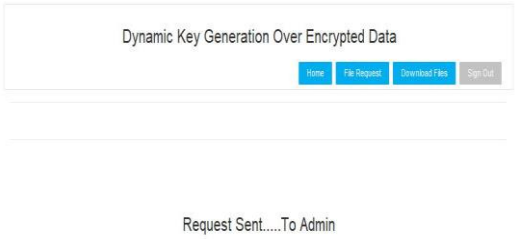
6. Data user login:

7. User page after login:

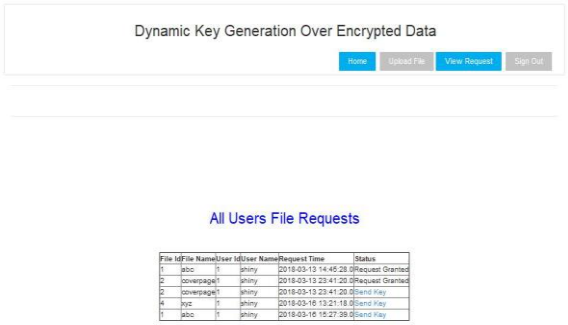


8. Send request to the files:

9. Request sent to the data owner for the requested file:

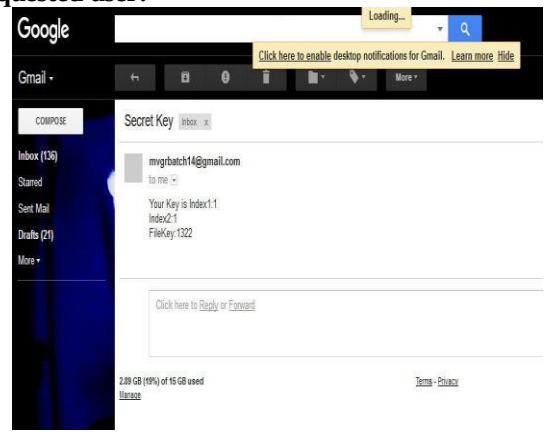


10.View all the requests sent in data owner page and send the secret key and file indexes to mail:

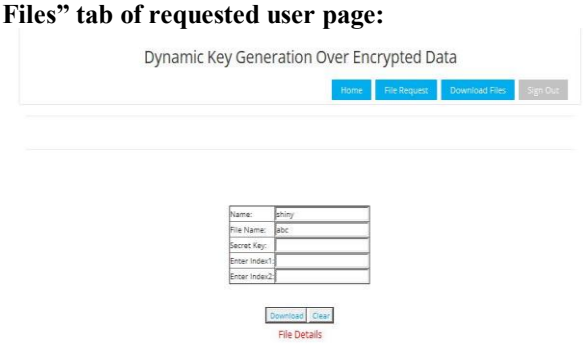


11. After sending the key, the request is granted for the user:

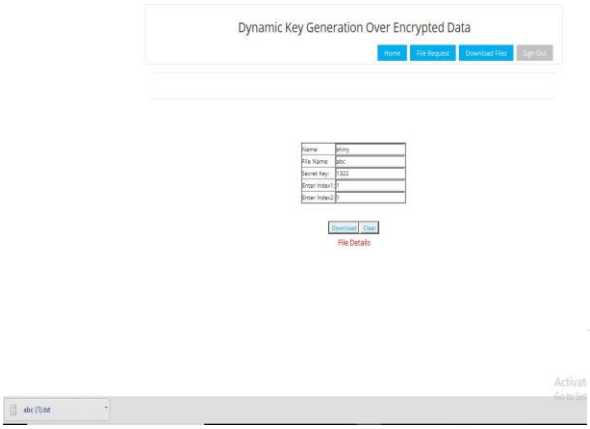
12. secret key and file indexes sent to email of the requested user:



13. **Enter the secret key and file indexes in the "Download Files" tab of requested user page:**

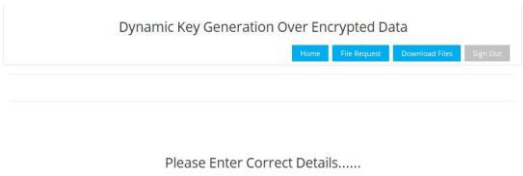


14. After clicking the Download button, the requested file is downloaded which is seen at the bottom left :

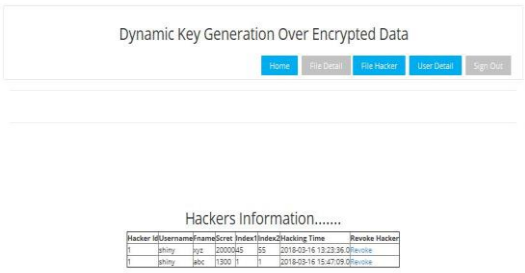


15. The user enters wrong secret key or indexes for the requested file:

16. Page displayed after entering incorrect details:

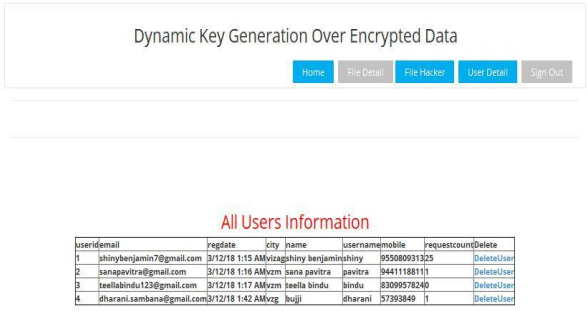


17. User entering wrong details are tracked and stored in the "File Hacker" tab of cloud server:



18. Details of all the files can be viewed in "File Detail" tab:

19.All the users information is stored in the cloud:



V. CONCLUSION

In this project, multi-indexed and secret key generation scheme with dynamic deletion and insertion of documents is proposed. By which privacy-preserving is possible and at the same time it is efficient. The AES algorithm is used to provide security against two threat models. Firstly, all the users usually keep the same secure key for trapdoor generation in a symmetric SE scheme wherein revocation of the user is big challenge. In order to revoke a user, we need to rebuild the index and distribute the new secure keys to all the authorized users. Secondly, generally symmetric schemes assume that all the data users are trustworthy. But, a dishonest data user leads to many security issues. For example, a dishonest data user may search the documents and distribute the decrypted documents to the unauthorized ones. Also, a dishonest data user may distribute his/her secure keys to the unauthorized ones.

VI. REFERENCES

[1]. Xia, Zhihua, et al. "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data." IEEE Transactions on Parallel and Distributed Systems 27.2 (2016): 340-352.

[2]. Prasanna B.T, C.B. Akki, 'A Survey on Homomorphic and Searchable Encryption Security Algorithms for Cloud Computing,' Communicated to International Journal of Information Technology and Computer Science, November, 2014.

[3]. Prasanna B.T, C.B. Akki, 'A Comparative Study of Homomorphic and Searchable Encryption Schemes for Cloud Computing,' Communicated to International Journal of Communication Networks and Distributed Systems, November, 2014.

[4]. Prasanna B.T, C.B. Akki, 'A Survey on Challenges and Security Issues in Cloud,' Presented in conference presented in Conference on Evolutionary Trends in Information Technology, May 20-22 2011, at Visvesvaraya Technological University, Belgaum, Karnataka.

[5]. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in IEEE INFOCOM, April 2011, pp. 829– 837.

[6]. S. Kamara and K. Lauter, "Cryptographic cloud storage," in Financial Cryptography and Data Security. Springer, 2010, pp. 136–149.

[7]. C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[8]. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, 'Public key encryption with keyword search,' in Proc. of EUROCRYPT, 2004.

[9]. W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, 'Secure knn computation on encrypted databases,' in Proc. of SIGMOD, 2009.

[10]. K. Ren, C. Wang, and Q. Wang, 'Security Challenges for the Public Cloud,' IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[11]. Zhangjie Fu et al, 'Multikeyword Ranked Search Supporting Synonym Query over Encrypted Data in Cloud Computing', IEEE Conference, 2013.

[12]. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, 'Searchable symmetric encryption: improved definitions and efficient constructions,' in ACM CCS, 2006.

[13]. P. Naresh, K. Pavan kumar, and D. K. Shareef, 'Implementation of Secure Ranked Keyword Search by Using RSSE,'

International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 2 Issue 3, March – 2013.

[14]. Buyrukbilen and S.Bairas, 'Privacy preserving ranked search on public key encrypted data,' in Proc. IEEE International Conference on High Performance Computing and Communications (HPCC), November 2013.

[15]. B. H. Bloom, 'Space/time trade-offs in hash coding with allowable errors,' Communications of the ACM, vol. 13, no. 7, 1970, pp. 422– 426.

[16]. C. Gentry and Z. Ramzan, 'Single-database private information retrieval with constant communication rate,' in ICALP, pp. 803–815.2005.

[17]. Sun, W., Wang, B., Cao, N., Li, M., Lou, W., Hou, Y.T., Li, H., 'Privacy-preserving multi keyword text search in the cloud supporting similarity-based ranking,' Proceedings of the 8th ACMSIGSAC symposium on Information, computer and communications security, ACM, pp