ISSN 2395-1621

A Secure Re- Encryption with key Distribution Scheme for data sharing in Unreliable Cloud Environment

[#1]Vaibhav Mahode, [#2]Akshay Patil, [#3]Shubham Chaudhari, [#4]Rohan Vanga,

[1]vaibhavmahode@gmail.com
[2]akshaypatil9123@gmail.com
[3]shubhamchaudhari2017@gmail.com
[4]rohanvanga.rv@gmail.com

[#1234]Department of Computer Engineering,

JSPM's,
Imperial College of Engineering & Research, Wagholi, Pune.

ABSTRACT

The data of any organisation which is stored on cloud is very important and access by all authorised user of part of organisation. But some time's the situation happen like in organisation or private/public company sacks their employee or people revoke the organisation. But still the having authorised credential and they are the authorised revoked user. Though to provide a security by applying a traditional way it's not possible. So we discuss the double DES (re-encryption technic) and key sharing methodology. As a result of this the non-revoked authorised user will get the data and having the restriction on third party (revoked) users. Because of this revokes authorised used fails to get plain text data even though they have credentials.

Keywords: Private Data, Encryption, Decryption, key sharing, Data Security, Cloud Computing, double DES algorithm

ARTICLE INFO

Article History

Received: 28th April 2017

Received in revised form :

28th April 2017

Accepted: 3rd May 2017

Published online :

3rd May 2017

I. INTRODUCTION

In recent era, cloud services are used by many users as well as industries. Cloud provides large amount of space to store data as well as share data so that it can be available any time over network when user requires. Cloud provides such services in low cost. Compared to traditional technologies, the cloud has many specific features, such as its large scale and the fact that resources belonging to cloud providers are completely distributed, heterogeneous and totally virtualized [1]. Users can store as well as share pictures, videos or any file over cloud so that it can be accessed on demand. The data stored over cloud has security issues; it is vulnerable to security threats. User can store any sensitive information over cloud. If cloud server get direct access to all these users' data, it may try to analyse the documents to get private information. The initial purpose of this action may be kind. The server wants to provide better service by digging into these data and then displaying customer-oriented advertisement, which could be convenient but also annoying. Besides, when we consider sensitive data such as personal health records and secret chemical ingredients, the situation becomes even more serious [2].Theoretically, the server is not supposed to have access to sensitive data at all; therefore we should ensure the server has no access to leaking these data to an untrusted third party. Thus, sensitive data have to be encrypted before being outsourced to a commercial public cloud [3].However, encryption on sensitive data presents obstacles to the processing of the data. Information retrieval becomes difficult in the encrypted domain because the amount of outsourced files can be very large and traditional search patterns can not be deployed to cipher text retrieval directly. Users need to download all the data, decrypt it all, and then search keywords like plaintext retrieval. To overcome this, Searchable Encryption (SE) [4] Applying order preserving encryption

© 2017, IERJ All Rights Reserved

(OPE) [5] is one practical way of supporting fast ranked search.

II. PROPOSED SYSTEM

Cloud users store their data in encrypted form to maintain data privacy. Two approaches that are used to securely share data in cloud storage. Firstly, encrypt data using a symmetric key and share that key among the authorized users. Secondly, encrypt data using the individual public key of the authorized users. Authorized users can access plaintexts data by decrypting the corresponding ciphertexts using their respective private key.

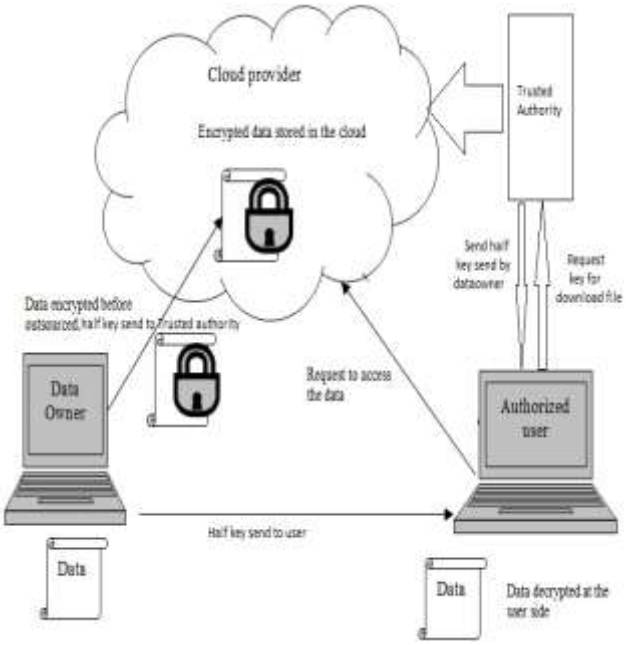

Fig 1. System Architecture

1) Data Owner:

A data owner can be an individual or a corporation, i.e., it is the entity that owns a collection of documents Dc = fD1;D2 : : :DNd g that it wants to share with trusted users. The keyword set is marked as W = fw1;w2 : : :wNw g.

2) Cloud Servers:

It is the place of hardware and software resources where a pool of data files and different applications can store. A cloud server conducts a secure search based on an encrypted index. In the search procedure, a user first generates a search request in a secret form a trapdoor T (w). In this example, the trapdoor is just the hash values of the keyword of interest. Once the cloud server receives the trapdoor T (w), it compares it with the hash values of all keywords in the index I, then the desired documents which are corresponding to keyword w are found.

3) Data Users:

The user can download all the encrypted documents based on the given IDs and decrypt them. A desirable system is supposed to return the documents in a ranked order by their relevance with the queried keyword, but using traditional encryption schemes will disorder relevance scores.

4) admin:

The data admin observe the key sharing and how data can transform from cipher text to plain text.

Algorithm Used:

Double DES
p->(E(k1, p))->(E(k2, (E(k1, p)))) =C

Double DES has a 112-bit key and enciphers blocks of 64 bits. DES is not a group; i.e., 21 is not equivalent to DES encryption using a single key. Recall that, for example, the Caesar cipher is a group. If a message were encrypted with the Caesar cipher with a key of 3 and then re-encrypted with the Caesar cipher with a key of 5, the result is equivalent to encrypting the message with the Caesar cipher with a key of 8. For the Caesar cipher, double encryption does not increase security. DES is not a group; double encryption is not equivalent to single encryption.
Security does increase by double encryption, but it does not increase much. The security of DES depends on its having a large key space; so large that (at least when it first began being used in the 1970's a brute force attack was not practical [that has now changed]). Recall that DES has a 56-bit key (the key is actually 64 bits, but every 8^{th} bit is a parity check; so, only 56 or the 64 bits ar e meaningful); therefore, the size of the key space is Recall that the algorithm that was originally proposed had a 128-bit key, but the size of the key space was reduced by the NSA (for some reason).
56
2
72, 057, 594, 037, 927, 936
=
Intuitively, double encryption should double the size of the key space. But, that is not the case with DES.

Algorithm Steps:

1) Encrypt the plaintext blocks using single DES with key K_1.

2) Now decrypt the output of step 1 using single DES with key K_2.

© 2017, IERJ All Rights Reserved

3) Finally, encrypt the output of step 2 using single DES with key K_3.

4) The output of step 3 is the cipher text.

5) Decryption of a cipher text is a reverse process

III. MATHEMATICAL MODEL

System Description:
Input:
Upload file ()
U : Upload file.
E : Encryption File.
F : file for security.
S : Store data base.

Output:
Stored Encrypted file to the Database.

Input
Function Encryption (id, request, file, key)
ID : unique id for each file.
Request : User request for particular file.
File : Check file on DB.
Key : Input key for decryption

Output:
File will recover to data user.

Success Conditions: Encryption will done for input file

Failure Conditions: Our system fails when no any security policy apply to the input file.

IV. RESULT ANALYSIS

Fig 2. File upload by owner

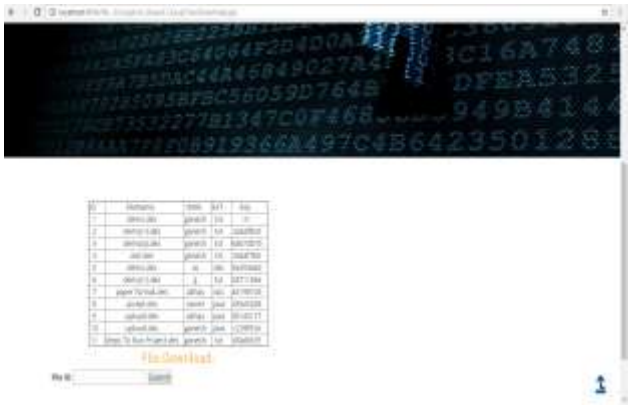

Fig 3. File download list

Fig 4. File access permission to the user

V. ACKNOWLEDGEMENT

I wish to express my profound thanks to all who helped us directly or indirectly in making this paper. Finally I wish to thank to all our friends and well-wishers who supported us in completing this paper successfully I am especially grateful to our guide for him time to time, very much needed, valuable guidance. Without the full support and cheerful encouragement of my guide, the paper would not have been completed on time.

VI. CONCLUSION

One-to-Many OPE is designed for encrypted data over the cloud and to preserve the order of relevance scores. cloud server can estimate the distribution of relevance scores by change point analysis on the differences of cipher texts of One-to-Many OPE. In this system we have described to improve One-to-Many OPE using this method. The system provides query privacy in search process under encrypted cloud data services. Search duration is reduced in the semantic relationship based encrypted keyword search process. Accuracy is improved with relevance score and semantic query model.

© 2017, IERJ All Rights Reserved

REFERENCES

[1] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, 34(1): 1-11, 2011.

[2] A. Boldyreva, N. Chenette and A. ONeill, "Order-preserving encryp-tion revisited: improved security analysis and alternative solutions," Advances in CryptologyCCRYPTO, 2011. Springer Berlin Heidelberg, pp. 578-595, 2011.

[3] L. Xiao, I.-L Yen, "Security analysis for order preserving encryption schemes," Proc. of 46th Annual Conference on Information Sciences and System, pp. 1-6, 2012.

[4] C. Wang, N. Cao and K. Ren, "Enabling secure and efficient ranked keyword search over outsourced cloud data," Parallel and Distributed Systems, IEEE Transactions 23(8), pp. 1467-1479, 2012.

[5] S. Yu, C. Wang and K. Ren, "Achieving secure, scalable, and fine-grained data access control in cloud computing," INFOCOM, 2010 Proceedings IEEE. IEEE, pp. 1-9, 2010.

© 2017, IERJ All Rights Reserved