ORIGINAL RESEARCH PAPER
VOLUME-6 | ISSUE-5 | MAY - 2017 | ISSN No 2277 - 8179 | IF : 4.176 | IC Value : 78.46
INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH

DYNAMIC AND PUBLIC AUDITING WITH FAIR ARBITRATION FOR CLOUD DATA

Computer Science

KavithaPriya.C. J Assistant Professor, Department of Information Technology, Jeppiaar Institute of Technology

ABSTRACT

Cloud computing plays major role in different organization and it provides different types of services to its users thus providing services along with confidentiality and integrity is issue in cloud, thus to provide security and trust we are introducing the third party arbitrator in cloud to support the all security aspects for both data owners and cloud service providers ,it also ensures integrity and confidentiality for dynamic data in cloud the security is provided using blowfish algorithm while data transmission.

KEYWORDS:

Third party auditor, Intermediary arbitrator, Index switcher, Trust and Security.

INTRODUCTION

Cloud computing is the internet based technology which stores and retrieves the data based on the demand ,it is usually termed as " pay as per you go" model this provides service to its users based on the demand they provides services like IAS,PAS,SAS as services provided to the user should be secured as shown in fig.2.1.

The major challenge faced by cloud is security and the trust in our paper we are providing security for dynamic data in a cloud using TPA. In cloud computing data handling is burden to the data owner. Third party auditor provides provide security only to the data owners not to the CSP, the data owner can complain that CSP misused the data provided.

SYSTEM WORKING AND ARCHITECTURE

In cloud computing, the two major actors are data owner and the cloud service provider .data owner provides data to the cloud for access. The third party auditor in existing system verifies the cloud user is genuine or not and it also reduces the workload of a data owner. In this system there is no intermediate to support both data owner and CSP and it also difficult to maintain data in dynamic. Also it corroborates only invariable data. The gradual growth of the cloud computing technology is tabulated in Table-1.

TABLE – 1 GROWTH HISTORY OF CLOUD COMPUTING

Year	2008	2012	Growth
Cloud Spend on IT	16B Dollar	42B Dollar	27%
Total Spend on IT	383B Dollar	494B Dollar	7%
Total Spend on Cloud	367B Dollar	452B Dollar	4%
Cloud Total Spend	4%	9%	

THE ARCHITECTURE OF EXISTING SYSTEM

Proves support and avoids workload authenticates data owner. The existing system has the system architecture as represented in the fig 1. includes the Data owner which request the cloud service provider through the third party auditor which paves the path to the secure transfer of the data.

Fig.1 Represents the Architecture of Existing System

EXISTING SYSTEM

In existing system the protection to the data and providing trust for the client and CSP is major challenge. They also support only static data ,the data owner provides data to the CSP, the CSP authenticates the data owner is valid or not if the data owner authenticated user. The third party auditor in this system authenticates data owner and also avoid workload, burden of data storage. The data owner can claim that the CSP misuse my data and they are not providing service properly. There is no third party to support CSP, thus we are introducing the third party arbitrator to support both data owner and the CSP.

DISADVANTAGES OF EXISTING SYSTEM

1. They provides trust only to the data owner they are not favorer for CSP

2. Tag re computation is difficult

3. They provides services only for static data
4. Data update, insertion, deletion is impossible because the data are transformed in form of blocks thus identifying the particular block and tag of that block is impossible.

PROPOSED SYSTEM

In our proposed system as in fig.3 we introduces the third party arbitrator who acts as an inter-mediatory for data owner and the cloud service provider to establish trust and it also supports dynamic data. The third party arbitrator maintains the list of activities of both data owner and CSP in case of any disputes it provides the details about the work done by both the parties so that we can identity who is fraudulent. It also supports effective trust and good relationship between the data owners and the cloud service provider.

The another issue in the existing system is that it does not support the dynamic data because the data are transformed in form of blocks each block have tag number or segmentation number, if in case consider their 455 block of data the user have to delete the data in 56 block it is impossible to identity that particular block and to delete that block. So we introduce the index switcher who maintains the mapping of data block and tag number, so that particular bock has been identified and data can be updated, deleted or inserted as

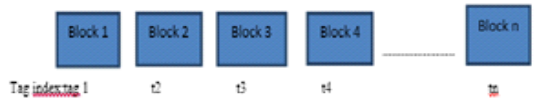

Fig. 2 Shows the Mapping of Data Block and Tag Number for Updating of Data.

ARCHITECTURE OF PROPOSED SYSTEM

The authenticated data owner provides data to the cloud service provider through the third party auditor. If any disputes occur in between them the third party arbitrator acts as an intermediate to solve the dispute, it maintains the list of activities done by both the parties as shown in fig 3.2. If data owner updates any data the notification goes to trusted third party arbitrator. The work of arbitrator is to maintain the list of activities performed by both parties they does not have any provision to read or view the data of cloud and owner to ensure security here we are using blowfish algorithm to encrypt and decrypt the data in transaction.

ALGORITHM

The blowfish algorithm is fast and effective algorithm used for encrypts and decrypt the data during transformation

TERMS

Encryption : It is the process of converting the plain text into cipher text using keys on the sender side

Decryption : It is the process of converting the cipher text into plain text using the key on the receiver side.

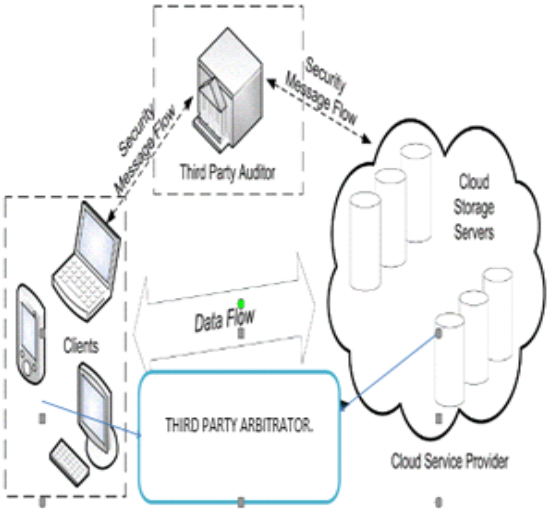

Fig. 3 Shows the System Architecture of the Proposed System

PRINCIPLE OF BLOWFISH ALGORITHM

Blowfish algorithm is the symmetric key block cipher algorithm which provides efficient encryption and decryption for the data during transformation. blowfish algorithm is 64 bit block cipher and length ranges from 64 bit to 448 bit, it performs encryption and decryption in 16 rounds. The two task performed by this algorithm is key expansion and data encryption.

WORKING OF ALGORITHM

Key management: It converts key of 448 bits to sub keys of total 4168 bytes. In each round it performs permutation and substitution. It performs operation like XOR and addition. Blowfish algorithm comprises of many sub keys and are pre- computed before encryption and decryption. In the algorithm, array comprises of 18 32- bit sub keysa1,a2,a3,.....a18 in that four 32-bit s-box with each 256 entries.

S(1,0) S(1,1) ..... S(1,255)

S(2,0) S(2,1) ..... S(2,255)

S(3,0) S(3,1) .....S(3,255)

S(4,0) S(4,1) .....S(4,255)

Data encryption: blowfish algorithm performs 16 rounds of operation. It takes 64-bit as input and divide into two halves each of 32-bitsas left (L) and right(R) so that for 16 rounds of iteration ,it performs operation as follows

For i=1 to 16

$$X_l = x_L \oplus P_i \quad (1)$$

$$X_R = X_R \oplus P_{17} \text{ and } X_L = X_L \oplus P_{18} \quad (2)$$

Xl and Xr are swapped in the above equation (2).

$$x_r = F(X_L) \oplus X_r \quad (3)$$

After 16 th iteration, swap x_L and $x_r$once again to undo last swap now, XL and XR are combined as in (2) to get cipher text as shown in fig.4. The generation of cipher text undergoes various process and number of rounds by using the keys as the input. It includes the input plaintext of about 64 bits and 16 keys for the rounds of 15.

As given in the equations 1,2,3, the swapping of the elements are done with 15 iterations and ends up in the 16th iteration. Finally from the input of 64 bits plain text, 64 bits cipher text is generated. This cipher text provides secured data to the user. The proposed algorithm converts the key of 448 bits to the sub keys of a count of 4168 bytes. Hence by performing this conversion, the major need of the system for accessing the data securely can be achieved.

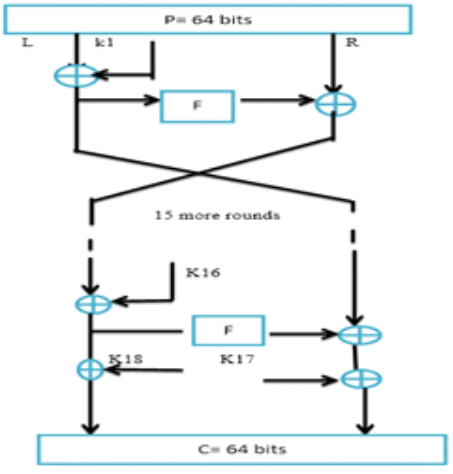

Fig. 4 Shows the Working of the Algorithm of the Proposed System.

CONCLUSIONS

The aim of this paper to provide an integrity, confidentiality, public verifiability, fair arbitration and dynamic data support to eliminate the limitation of index usage in data dynamics and to support fair data dynamic with efficient update, deletion, insertion in cloud. We differentiates between the block index and tag index .meanwhile both data owner and CSP can misbehave so, we extended existing threat model to identify the conflict and to solve disputes between the cloud service provider and the data owner.

REFERENCES

[1] Y. Deswarte, J.-J. Quisquater, and A. Sa¨ıdane, "Remote integrity checking," in Proc. 5th Working Conf. Integrity and Intl Control in Information Systems, 2004, pp. 1–11.
[2] D. L. GazzoniFilho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfer." IACR Cryptology ePrint Archive, Report 2006/150, 2006.
[3] A. Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Computer and Comm. Security (CCS07), 2007, pp. 584–597.
[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Computer and Comm. Security (CCS07), 2007, pp. 598–609.

[5] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. 14th Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 08), 2008, pp. 90–107.
[6] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing,"
in Proc. 14th European Conf. Research in Computer Security (ESORICS 08), 2009, pp. 355–370.
[7] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents." IACR Cryptology ePrint Archive, Report 2008/186, 2008.
[8] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," Network, IEEE, vol. 24, no. 4, pp. 19–24, 2010.
[9] C. Erway, A. Kupc¸ ¨ u, C. Papamanthou, and R. Tamassia, "Dynamic¨ provable data possession," in Proc. 16th ACM Conf. Computer and Comm. Security (CCS 09), 2009, pp. 213–222.
[10] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Dynamic audit services for integrity verification of outsourced storages in clouds," in Proc. ACM Symp. Applied Computing (SAC 11), 2011, pp. 1550–1557.
[11] Q. Zheng and S. Xu, "Fair and dynamic proofs of retrievability," in Proc. 1st ACM Conf. Data and Application Security and Privacy (CODASPY 11), 2011, pp. 237–248.
[12] A. Kupc¸ ¨ u, "Official arbitration with secure cloud storage applica-¨ tion," The Computer Journal, pp. 138–169, 2013.
[13] N. Asokan, V. Shoup, and M. Waidner, "Optimistic fair exchange of digital signatures," in Proc. 17th Intl Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT98), 1998, pp. 591–606.
[14] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. IEEE INFOCOM, 2010, pp. 1–9.
[15] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for secure cloud storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362–375, 2013.
[16] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," IEEE Trans. Cloud Computing, vol. 2, no. 1, pp. 43–56, 2014.
[17] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in Proc. 22nd Intl Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT03), 2003, pp. 416–432.
[18] P. A. Bernstein and N. Goodman, "An algorithm for concurrency control and recovery in replicated distributed databases," ACM Trans. Database Systems, vol. 9, no. 4, pp. 596–615, 1984.