Ciaac: Ensuring Cia And Cp-Abe Based Access Control For Secured Storage In Cloud Computing

Mrs. K. Vidhya[1], Mr. R. Sivaramakrishnan[2]

[1]Assistant Professor
Department of Computer Science and Engineering
Sri Shakthi Institute of Engineering and Technology
Coimbatore.

[2]Assistant Professor
Department of Computer Science and Engineering
Sri Shakthi Institute of Engineering and Technology
Coimbatore.

Abstract

Cloud computing is basically a methodology of outsourcing the technology assets. The combination of technological advancements and changes in people mindset made cloud computing popular. But the perceived lack of security and privacy is an important factor for many industries have been reluctant to store their data in cloud. They don't want anyone including the cloud service provider to view their content and to know their list of customers. In order to address these problems in the areas as data Confidentiality, data Integrity and Availability(CIA), we propose the Cipertext Policy Attribute Based Encryption for Access Control(AC) and consumer confidentiality. Raptor code based storage system with TPA ensures the data Integrity. The authorized user who satisfies the access structure can access encoding and decoding and verify the cloud storage with linear cost for when compared with the existing methods.

1. Introduction

The various challenges like Globalization, Aging and rich maintenance of data centers ,Storage growth, Increased application Explosion, Ownership cost and various kind of Acquisitions makes the organizations to get into the concept of cloud computing .In order to meet the competition they must be able to respond quickly ,must be agile and also flexible. For meeting these basic requirements the cloud platform would be an incomparable boon. The services in the cloud are great for storing documents and for accessing from anywhere on any device. Through cloud computing people have been able to use someone else's software on their systems for long time. But there is a unavoidable fear about the security breaches of cloud computing. Normally the internet based services are vulnerable until and unless the specified security mechanisms are enforced.Basic framework of security must concentrate more on three major areas such as data Confidentiality, Integrity of data and the 24 ×7 ×365 data Availability (CIA). The data confidentiality is the assurance of not disclosing the information to un-authorized users or processes. Logical security of our data can be achieved by implementing the basic cryptographic technique called encryption. The Data integrity assures that our data is not modified .Through Availability the authorized users can have reliable and timely access the data. In order to achieve all the above goals the step stone activity is restricting the users or access control systems.

The proposed framework [Fig.1] meets all the above three primary goals of security. The ABE scheme ensures the data confidentiality by encrypting the data files using the components of public key corresponding to their most specific attributes. The secret key for an authorized user are defined by the access tree [13]. So the user will be recognized as an authorized user and he can decrypt the cipher text if and only if the data file attribute satisfy his

corresponding access structure. Thus this framework enforces the access control.

The proposed system uses Raptor code to address the problems of cloud data integrity. The erasure code is a file distribution scheme that is applied to the vectors of the data to be stored [1]. Then the erasure coded data is distributed across multiple servers, such a way that the original data file can be reconstructed from a few distributed erasure coded data available at different severs. This is helpful to achieve the data availability in a cloud environment.

Before such an erasure coded file distribution, the user calculates a set of token values for each data vector. Whenever, the user wants to verify the correctness of the cloud data, he requests the TPA to perform data integrity auditing. The TPA computes the token from the challenged data vectors which should match the token values already computed by the cloud users. This method helps to identify the corrupted data, localize the error and recover from the error. Further, the scheme discovers the faulty servers.

The proposed scheme is extended with Raptor Code. First, the data vectors are applied traditional erasure code and then appropriate raptor code is applied to the new set of symbols in a way that the traditional erasure code is capable of recovering the data even in a fixed fraction of erasure codes. The proposed system also supports third party auditing (TPA). The task of correctness verification may be delegated to a trusted third party auditor, when the CSP or user has no time to perform auditing. The scheme also supports the dynamic nature of the data. With these strategies, security and reliability are improved. Efficiency is improved in terms of encoding and decoding costs. The framework [Fig.1] contains three domains such as Data owner Domain [DOD], Storage Domain [SD] and Access Control Domain [ACD].The DOD is data owner's perspective. It deals the data owners willing based data encryption structure to make sure themselves that they haven't reveal their confidential data to some service provider as well as to some Third Party Auditors[TPA] primarily.SD facilitates an efficient and secured storage of data in distributed data servers. ACD ensures the authorized access.

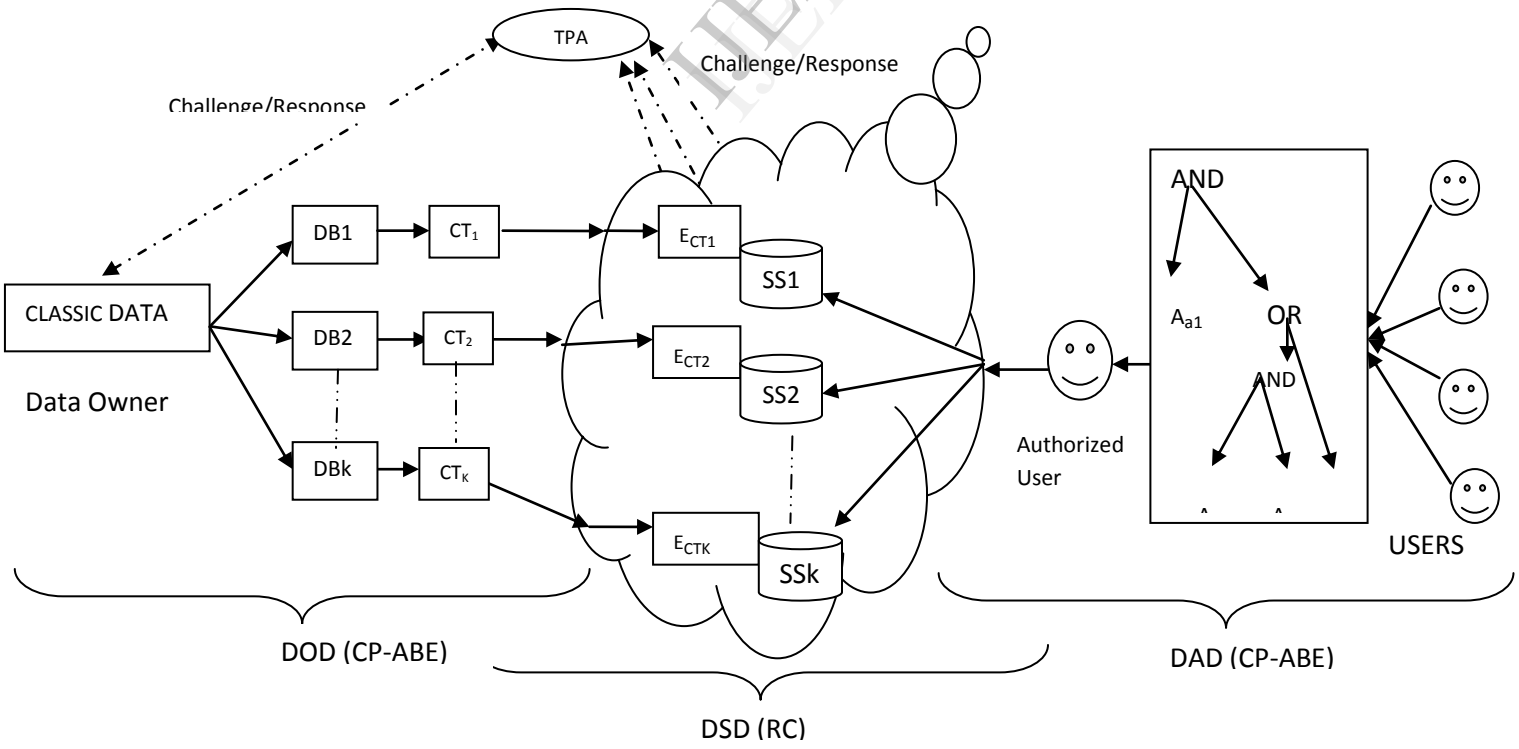

Fig. 1 CIAAC Framework
(DB – Data Block, CT – Cipher Text, E_{CT} – Encoded CT, SS – Storage Server)

2. Configuring a Secured Cloud Storage System

2.1. Data Confidentiality – ABE

Sahai and Waters [12][13] made introduced the concept of Attributed-Based Encryption (ABE).The data owner can enjoy the confidentiality of his data by providing an fine grained access control based on the attribute based encryption.In the Attribute Based Encryption scheme ,the keys and ciphertexts are labelled with sets of descriptive attributes and a particular key can decrypt a particular ciphertext only if there is a match between the attributes of the ciphertext and the user's key[11]. The cryptosystem of Sahai and Waters allowed for decryption when at least n attributes overlapped between a ciphertext and a private key. In the framework [Fig.1] the input data is split into number of blocks (DB) and each block is encrypted by CP-ABE with very essential attributes. We have constructed the access tree structure with AND and OR gates as interior nodes. The leaf node consists of various essential attributes. Any user's entry which meets the tree structure can get the decryption key and can decrypt the data. By this the ABE system provides a fine grained access control.

CP-ABE Procedures:

- **Setup(k):** The setup algorithm takes as input a security parameter sk and outputs the public parameters pk and a master key mk.
- **Keygen(*A*,mk):** The algorithm takes as input the master key mk and a set of attributes *A*. The algorithm outputs a secret key sk_A associated with *A*.
- **Encrypt(m,*T*,pk):** The encryption algorithm takes as input the public key pk, a message m, and an access tree *T* representing an access structure. The algorithm will return the ciphertext ct such that only users who have the secret key generated from the attributes that satisfy the access tree will be able to decrypt the message.
- **Decrypt(ct, sk):** The decryption algorithm takes as input a ciphertext ct, a secret key sk associated with *T*, and it outputs a message m or an error symbol. The process of obtaining the secret key is associated to set of attribute*[11]*.

2.2. Ensuring Cloud Data Integrity and Availability

2.2.1. Reed Solomon Erasure Code

An (M, K) Reed Solomon Erasure Correcting Code [1] is used to generate K redundant parity vectors from M data vectors of the encrypted data(ct) file E_D to be stored. The concept is that the original M data vectors can be reconstructed from any M out of the M + K data and parity vectors. All these M + K data vectors are distributed across N servers (N = M + K). The erasure code can tolerate up to K failures out of these M + K servers. Reed Solomon Erasure Correcting Code is based on the Vandermonde matrix formed from the data vectors of the file to be stored. By applying sequence of row transformations, a secret matrix P is generated from the Vandermonde matrix which is then used to generate M + K data vectors to be distributed across N servers. The secret matrix calculated here plays an important role in error localization and recovery. The encoded cipher text (ct) file is given as,

$$E_{CT} = E_D . A = (G^{(1)}, G^{(2)}, G^{(3)} \ldots G^{(m)}, G^{(m+1)} \ldots G^{(n)}) = (E_{D1}, E_{D2} \ldots E_{Dn}, G^{(m)}, G^{(m+1)} \ldots G^{(n)})$$

2.2.2. Computing Token Values from Data File

As discussed earlier, before distributing these data vectors across the several servers (N), a user computes a set of token values for each of the M + K data vectors based on the permutation key, in order to identify the integrity of the data and to localize the error. The user calculates a token value for each server with selected indices of the data vector, based on the challenge key and permutation key. These token values may be kept locally by the user or stored in an encoded format on the cloud.

After computing the token values and distributing the data file, whenever the user wants to guarantee the integrity of the data, he requests the cloud servers to perform auditing on a set of indices. The cloud server is aware of the permutation key and generates a signature for the requested indices of the data vector. If the integrity of the data is preserved, the value of this signature matches the token value

already computed by the user. If the signature value doesn't match the token value, the integrity of the data is lost because of malicious attack or server failure.

2.2.3. Performing an Audit to Verify Data Integrity and Localize the Error

To perform integrity auditing, a user gives the indices of the data vector and permutation key to each server (N). The server has to compute the token that is, generate signature on the data vector stored on it. Upon receiving this signature from all servers, the user verifies whether these values remain valid as determined by the secret matrix. That is, the product of the secret matrix and M responses from each server should match the remaining K responses (N = M + K). This audit determines which block of data is corrupted and thus identifies the integrity breach.

$$R^{(i)} =? T^{(i)}$$

Once if the user identifies that the integrity is lost, he uses the tokens already computed for each server. That is, the response received from the server should exactly match the token computed by the user. If such a condition fails, the corresponding server is identified to be faulty. The token based integrity auditing can discover up to M + K faulty servers. It doesn't tolerate failures more than M + K, which is the property of the erasure codes.

2.2.4. Ensuring Data Availability by Error Recovery

The integrity auditing not only reveals that the data is corrupted, but also indicates the faulty servers ($\leq$ M + K). The faulty servers are identified in the integrity audit and the corrupted data blocks are downloaded from those faulty servers. After retrieving them from the faulty servers, erasure correcting code is used to recover the original data. Once if the recovery of the original data is over, it sends back the corrected data to those faulty servers to maintain integrity.

3. Enhancements to the Cloud Storage Security

3.1. Support for Third Party Auditing

The proposed scheme also supports Third Party Auditor [2] (TPA) to perform integrity auditing and error recovery. If the user or the cloud server doesn't have the time or enough resources to perform integrity auditing, a trusted third party auditor may be delegated this task. The responsibility of the TPA is to identify the potential risks and threats to data integrity. Also, the user must ensure that the content of the data is not revealed to the TPA during the integrity audit. In order to assure that the TPA doesn't learn the data, the scheme may follow privacy preserving third party auditing [3] or zero knowledge third party auditing [4][5]. To delegate integrity auditing, and to preserve the contents of the data, the secret matrix is kept confidential by the user. The audit may be conducted without downloading the data file [7] from the servers which increases the burden on the Internet. However, with TPA the communication and processing overhead is same as the task carried out by a cloud server.

3.2. Support for the Dynamic Nature of Data

The data stored on the cloud may be static or dynamic. The concepts discussed so far should also be extended for active data. That is the user may wish to update, insert, append and delete data already stored on the cloud. In such a situation, the cloud server and the user should be able to provide the same level of security to the data.

Hence, to support the changing nature of the data, the user should ensure that the variations in the data are correctly reflected on all servers storing the data file and accordingly recalculate the tokens. Whenever the user makes changes to the data stored on cloud, he must recalculate token values to the changed data and inform all the servers about the updated data. After the token is recalculated, the data file is distributed as already addressed. During the integrity auditing, the recalculated token values are matched against the signatures generated by the cloud server or trusted third party auditor.

4. Raptor Code (RC) Based Cloud Storage Services

4.1. Applying Raptor Code for Cloud Storage Services

Raptor code [6] is an extension of the Luby Technique (LT) code mainly designed for

distributing the data file across the servers and identifying the errors with the property of linear time encoding and decoding. The concept of raptor code is, given any integer of J symbols, and any real E > 0, raptor code produces a stream of output symbols such that any symbols of size J (1 + E) is enough to recover the original J symbols with increased likelihood.

First, the cipher text blocks are encoded using the erasure codes and then LT-Codes are applied to the set of new input symbols in a way that the original input symbols can be recovered from the fixed portion of erasure codes. Raptor code has the parameters (J, S, σ(x)) where J is the number of input symbols, S is the codeword (intermediate symbols) generated with *'n'* symbols called the pre-code and σ(x) is the degree of distribution of the data file. The output symbols of the raptor code are generated from the *'n'* intermediate symbols by applying LT-Codes to them. The encoding function of the raptor code takes J input symbols and an encoding algorithm for S in order to generate the intermediate symbols of S. Then, the encoding algorithm for LT-Code is used to generate output symbols from the intermediate symbols with the degree of distribution σ(x).

The decoding function of the raptor code takes *'m'* output symbols of the raptor code and recovers J input symbols from any set of *'m'* output symbols with the probability $1/J^c$ for some constant *'c'*. The raptor code assures constant encoding and decoding costs with respect to the input symbols J. By integrating raptor code with erasure code, the proposed scheme achieves more security and efficiency.

5. Performance Analysis

According to the above discussion our proposed scheme will meet the security goals i.e., data confidentiality, user access privilege confidentiality and data availability by error correction with raptor code and also the access control. The doubly secured data with its access structure ensures the secrecy of data which is not revealed even with the auditor and cloud service provider or storage cloud servers. Thus the performance of our scheme is noticeable one comparing to the previous methodologies.

Here the performances of Reed Solomon erasure code and Raptor code are also analyzed. As discussed earlier, Reed Solomon erasure code requires more processing power for its encoding and decoding functions than Raptor code. The encoding and decoding functions in a Reed Solomon erasure code are quadratic according to the number of input symbols, where as it is linear to that of a Raptor code. The performance of Reed Solomon erasure code and Raptor code are analyzed based on their encoding and decoding functions.

The performance of Reed-Solomon and Raptor codes are assessed based on the number of XOR operations [8] required for encoding and decoding of input and output symbols. The workload of encoding function is the total number of XOR operations divided by *n*, which is the number of output encoded symbols; and the workload of decoding function is the total number of XOR operations divided by *k*, which is the number of source (input) symbols recovered from output symbols.

The performance analysis considers Reed Solomon erasure codes with 8 – bit field elements. The workload of encoding function is considered as (k/n) × (n-k) × 4 XOR operations for each output symbol encoded. And, the workload of decoding functions is considered as (n-k) × 4 XOR operations for each input symbol recovered from the output symbols.

In case of a Raptor code, the workload of encoding function is calculated as 4.5 + 7.5 × (*k*/*n*) XOR operations for each output symbol encoded. The workload of decoding function is calculated as 10 + 4.5× min (*n* − *k*, *k*)/*k* XOR operations for each input symbol recovered from output symbols. From the Fig. 2, it is clear that raptor code performs better than the Reed – Solomon erasure code in case of encoding and decoding costs with respect to the number of input symbols.

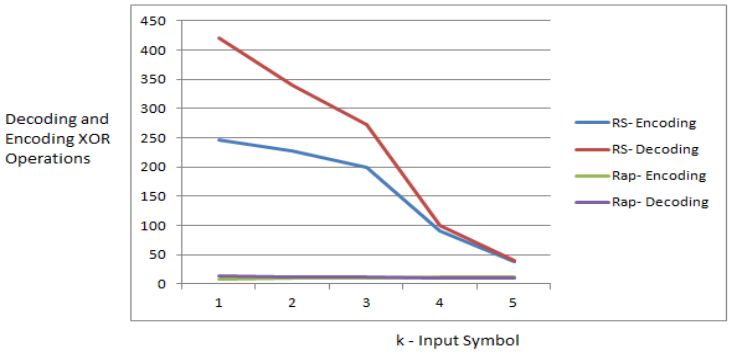

Fig. 2 A graph showing the performance of Reed Solomon and Raptor code

6. Conclusion

The proposed framework meets all the security goals. Here the data owners themselves ensures the confidentiality and integrity of outsourced data by CP-ABE scheme and challenge/response activity of auditing. Similarly the service provider ensures that the data stored and maintained by them are secured and there is only an authorized access due to the restriction in access tree structure. Since the CP-ABE system manipulates only the very essential attributes, the computation time and cost is low and tolerable. Our system analyses the erasure and raptor code based integrity auditing, which determines the potential corruption in a data file and identify the faulty servers. This scheme outperforms the previous auditing mechanisms which give only binary results about the status of the data file in the cloud. The scheme is further extended to support third party auditor and changing nature of the data. The usage of raptor code reduces the cost, overhead and space required to perform such integrity auditing. The scheme is strong against servers colluding attacks and malicious data modification.

7. References

[1] Cong Wang, Qian Wang, "Toward Secure and Dependable Storage Services in Cloud Computing", IEEE Transactions on Services Computing, No. 2, Vol. 5, June. 2012.

[2] J. Li, C. Wang, K. Ren, and W. Lou, "Towards Publicly Auditable Secure Cloud Data Services", IEEE Network Magazine, Vol. 24, No. 4, pp. 19-24, July/August. 2010.

[3] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy – Preserving Public Auditing for Storage Security in Cloud Computing", *Proc.* IEEE INFOCOM, Mar. 2010.

[4] Yan Zhu, Hongxin Hu, Gail John Ahn, S. Yau, "Efficient Audit Service Outsourcing for Data Integrity in Clouds", IEEE 2012 Transactions on Cloud Computing, Vol. 85, Issue 5.

[5] Yan Zhu, Hongxin Hu, Gail John Ahn, S. Yau, "Dynamic Audit Services for Outsourced Storages in Clouds", IEEE Transactions on Services Computing.

[6] Amin Shokrollahi,"Raptor Codes", IEEE Transactions on Information Theory, Vol. 52, No. 6, June. 2006.

[7] C. Erway, A. Kupcu, C. Papamanothu and R. Tamassia, "Dynamic Provable Data Possession", *Proc. 16th ACM Conf. Computer and Comm. Security (CCS'09)*, pp. 213-222, 2009.

[8] "Why DF Raptor is better than Reed Solomon for streaming applications", Digital Fountain Incorporated, A Qualcomm Company, 2010.

[9] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," *Proc. 11th USENIX Workshop Hot Topics in Operating Systems (HotOS '07),* pp. 1-6, 2007.

[10] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," ACM Trans. Computer Systems, vol. 20, no. 4, pp. 398-461, 2002.

[11] Efficient and Provable Secure Ciphertext-Policy Attribute-Based Encryption Schemes, Luan Ibraimi1, Qiang Tang1, Pieter Hartel1, Willem Jonker.

[12] Sahai, A., Waters, B.: Fuzzy Identity Based Encryption. In: Advances in Cryptology – Eurocrypt. Volume 3494 of LNCS, Springer (2005) 457–473.

[13] Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute Based Encryption for Fine - Grained Access Conrol of Encrypted Data. In: ACM conference on Computer and Communications Security (ACM CCS). (2006)

[14] Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute Based Encryption. Ming Li, Member, IEEE, Shucheng Yu, Member, IEEE, Yao Zheng, Student Member, IEEE,Kui Ren, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE,Transaction on parallel and distributed systems.Vol 24 ,No 1, January 2013.