International Journal of Advanced Scientific Technologies, Engineering and Management Sciences (IJASTEMS-ISSN: 2454-356X) Volume.3,Special Issue.1,March.2017

Incontestable Multicopy Dynamic Information Control in Cloud Computing Systems

A.Dhasaradhi, Associate Professor,
Dept.of CSE, SIETK, Puttur
dhasaradhi.a@gmail.com

P.Divyaja, Research Scholar,
SPMVV, Tirupati
pdivyajahime@gmail.com

Abstract:

Now a day's many Organizations are using Cloud computing for storing data. The organizations take rent from Cloud service providers (CSP) to store data in cloud storage. Small organizations are financially not capable to maintain servers, so they have to depend on cloud service providers for storing data in cloud. Customers pay the rent and collect data based on their usage on data size from the Cloud Service Providers. Always customers want their data for easy access to store multiple servers in the cloud computing system. Most of the Cloud Service Providers (CSP) aim is defrauds the customer's data from their storage on servers. Customers offer many number of copies to store but CSP is storing only partial no of copies. So customers need strong proof in the case of Cloud Service Providers (CSP) is storing all data copies that are agreed upon in the service contract. The Existing PDP (Provable Data Possession) scheme focus only on static data, once data was stored in cloud the customers won't change. The proposed Dynamic File Block (DFB) scheme dealing with Modification, Insertion, Deletion, and Append in Cloud Computing Systems with dynamic data. It allows customers to perform file block operations such as insertion, modification, deletion, and append and these scheme maintains verifier to verify the file in cloud. So the CSPs cannot cheat the customers because verifier check files after each operation.

Key Words: Cloud Computing, Dynamic Data, Dynamic File Block (DFB), storage security, Cloud Service Provider(CSP), Data Integrity, Multi-copy.

I.INTRODUCTION

Cloud Service Provider (CSP) is allows store more outsourcing data on private computer system. The data storage infrastructure to store and retrieve data and it store unlimited amount of data. Cloud computing provides virtualized computing resources over the internet. Many Geographic locations and Many Authorized users can access remotely stored data making it more flexible for them.

Data Owner loses the direct control over their sensitive data, once the data has been outsourced to a remote Cloud Service Provider (CSP). This lack of control raises new challenging tasks in cloud computing for integrity protection and data confidentiality. By using Encryption technique Confidentiality issue can be handled, As such, customers has strong evidence that cloud servers still possess their information and it is not being altered or tampered. Many researchers have focused on this problem.

The proposed Dynamic File Block Modification, Insertion, Deletion and Append in Cloud Computing Systems model focus on dynamic data. It supports file block operations such as modification, append delete. In this scheme preserve the verifier for checking data integrity over data files in remote servers. If data integrity test not succeed then the CSP did cheated the data owner regards of data files. In this scheme verifier maintains Map Version Table (MVT), this table contains information about which block was modified, inserted, deleted and appended in file block. The verifier inserts the new row in table when the new block was inserted, the verifier deletes the table row when the already exist block was deleted, the verifier increments the values in table column when appended the text in file block and the verifier also increments the values in table column when the owner modified the content in particular file block.

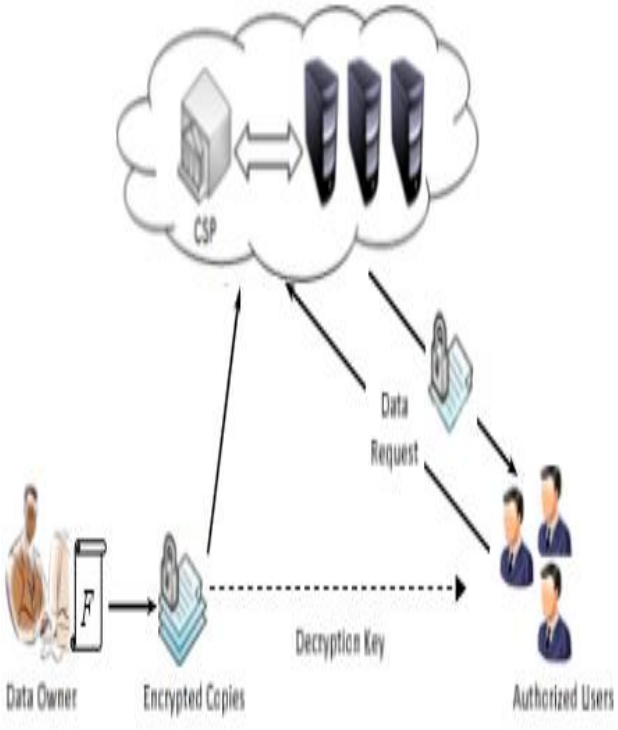
II.PROPOSED SYSTEM

Provable Data Possession at Untrusted Stores [1], schemes allows users to store files in untrusted servers in cloud computing system. The user needs to check whether the file is consistent in cloud servers without retrieving the file. In this scheme user maintains metadata (small information about file) about file that is used later for verification purpose. This scheme reduces the I/O cost. The challenge response protocol was used for verification of file consistency in cloud servers. Remote Integrity Check with Dishonest Storage Server [2]-[4], scheme maintains verifier to check the storage regularly because storage is deceitful. Jules introduces Proof of Retrievably (POR) [13]-[14], system to maintain security. The verifier uses Challenges-Responses protocol to check the files in deceitful storage. If file is large in size then the file deletes local copy of file. So we do not have the local file in your local system and we do not know if file is modified in cloud servers. The CSP may cheat when storing the file copies in remote cloud servers and tells the users that system crashed or hardware failed. So user need strong evidence the file must be safe in deceitful storage. Dynamic Provable Data Possession [8], the data owner stores more files in untrusted servers. This scheme focus on integrity of files stores on remote servers. The data

International Journal of Advanced Scientific Technologies, Engineering and Management Sciences (IJASTEMS-ISSN: 2454-356X) Volume.3,Special Issue.1,March.2017

owner wants the file copies to store multiple servers across the cloud computing systems. The integrity is main problem in cloud servers and CSP cheats the data owner to earn money. The verifier maintains metadata for each file to check the file consistency. But dynamic provable data possession scheme concentrate on consistent not allows insertion or deletion of file. Provable Possession and Replication of Data over Cloud Server [10]-[12], for improvement of scalability and availability many users want there to store on multiple servers. The data owner contact with server to store same file to replicate on multiple servers and this scheme also focus on multiple of static data. The scheme cannot allow the changes the file dynamically.

III.ARCHITECTURE



i) **Data Owner: -** Data owner generates keys that are required for sessions.

- It divides the files into blocks.
- These blocks are encrypted.
- These blocks are outsourced to the CSP.
- It receives location tags from the CSP and maintains the location details in it.
- It challenges the CSP to provide proof. It sends challenge to the CSP to verify whether the agreed number of copies are stored in the CSP. Proof is received by the data owner from different locations that are specified in the tags.
- After receiving the proof, proof is verified. If proof is correct then the exact copies of the files are maintained in the CSP. Then the data owner confirms reliability with the CSP. When the authorized users request to grant permission to access the file, data owner will share a key with the user and user will access the file with it.

ii) **Cloud Service Provider:-** CSP receives the file blocks outsourced to it.

- CSP creates multiple copies that agreed with the data owner.
- It sends the file copies to the location.
- After sending the file to location, tags are created with the details of the location.
- These created location tags are send to data owner.
- CSP receives a challenge from the data owner.
- When challenge is received, it is passed to the locations where the copies are stored.
- Each location computes a proof and these proofs are passed to the data owner with interactive zero knowledge protocol. Operations like insertion, deletion, modification, append are performed in the CSP on file blocks according to data owners' request. Insertion inserts a block anywhere in the file. Deletion deletes the block completely. Modification modifies the block content. Append operation adds a new block at the end of the blocks. After the operation change must be updated to all the copies present in the CSP.
- Request for accessing the file is received from the authorized users.
- After checking the authenticity encrypted blocks send to the authorized users.

iii) **Authorized Users: -** Authorized users request the data owner to grant permission to access the file from the CSP.

- It will receive a key from the data owner.
- After receiving the key, it will request for the file to the CSP.
- User will receive the encrypted blocks of the file in an unordered manner.
- Blocks are decrypted using the Shared secret key. These blocks are rearranged to get a complete file. Every file can be decrypted with the same key. Users can seamlessly access the file from the CSP.

IV.ALGORITHM

The proposed model consists of seven algorithms; they are

1. Key Generation
2. Copy Generation
3. Tag Generation
4. Update Preparation
5. Update Execution
6. Proof
7. Verify

1. KeyGeneration( ) → (pk, sk): The key Generation algorithm is used by the origination who is data owner. The data owner run any one of encryption algorithm to

International Journal of Advanced Scientific Technologies, Engineering and Management Sciences (IJASTEMS-ISSN: 2454-356X) Volume.3,Special Issue.1,March.2017

generate public key and private key (secrete key). The public key is denoted by pk and private key is denoted by sk. The data owner remains a private key sk by secret. The data owner sends public key pk is send to their clients.

RSA Algorithm: The RSA algorithm is used to generate public key pk and private key (secrete key) sk. The data owner maintains organization and the clients are working in organizations who are authorized users. The authorized users only have rights to access the owner file. The data owner generates public pk and private key sk before send the data file to Cloud Service Provider. The data owner kept private key sk secret and public key pk sends to clients Steps:

1. Choose two distinct prime number x and y. The x and y numbers both are integers, random and similar bit length.
2. Calculate z=x*y.
3. Calculate Euler's totient function Φ(z)=(x-1)*(y-1).
4. Choose an integer pk, such that 1<sk<Φ(z) and GCD of pk, Φ(z) is 1. pk is public key exponent.
5. Calculate sk=e-1(mod Φ(z)). sk is multiple inverse of e mod Φ(z).
6. sk is private key (secrete key) , sk*e=1 mod Φ(z).
7. Public key consists of modules n and public key exponent pk ie,(pk,n).
8. The private key consists of modules n private key exponent sk ie, (sk,n).

2. CopyGeneration(CNi, F)1≤k≤n → F : The copy generation algorithm is used by data owner. The data owner makes the file copy into many number of copies depend on important of file. The CopyGeneration algorithm has two inputs file copy number CNi and file F. The data owner generate output n file copies F = {Fi}1≤k≤n and send file copies to CSP, the CSP stores file copies on remote cloud server.

3. TagGeneartion(sk, F) → Φ: The TagGeneration algorithm is used by data owner. The data owner takes the private key sk, file copy number CNi and file F as input. The data owner generates the output tags set Φ. The file is divided into number blocks. The tags are assigning to file blocks. The data owner sends tags set Φ to CSP. The CSP keeps the tags set Φ along with the file copies F .

4.PrepareUpdation
(D, UpdateInformation) → (D', Updaterequest):

The PrepareUpdation algorthim is used by data owner, the data owner maintains metadata for each and every file. The data owner uses this metadata later for verification of the files. The data owner consists of two inputs metadata D and UpdateInformation.

The UpdateInformation input contains update (insert, delete and append) information to particular file block. The PrepareUpdation algorithm generate output D' modified metadata about file and UpdateRequest output may contains file block operation such as insert, delete and append to particular file and it may contain new tags for file. The data owner sends UpdateRequest to CSP for perform data owner request updation.

5. **ExecuteUpdation(F , Φ, UpdateRequest) → (F , Φ'):** The ExecuteUpdation algorithm is used by CSP. This algorithm contains three inputs file copies F , tag set Φ and UpdateRequest. The output contains updated version of file copy F with updated tag set Φ'. The new private key generation is not required but new tag insertion or deletion is performs on tag set Φ.

6. Proof(F , Φ, Challenge) → P:
The proof algorithm is used by CSP. The Proof algorithm have three inputs file copies F , tag set Φ and Challenge (it was sent from verifier). The Proof algorithm generates proof P, the proof P contains information about whether CSP maintains all files consist with respect to number of file copies and intact with inserted, deleted and appended file operation.

7. Verify(pk,P,D) → {1,0}: The Verify algorithm is used by verifier. The verifier may be data owner or owner trusted person. The verify algorithm has three inputs public key pk, proof p from the CSP and the recent metadata D for file. The algorithm returns 1 if all the files are consist with recent modification and return 0 other wise.

Map Version Table (MVT): The verifier maintains MVT to verify whether the all files in cloud servers is consistent or not. The MVT is dynamic data structure and the table contains mainly three columns Index Number (IN), Block Number (BN) and Block Version (BV). IN is physical position of file block and BN is logical position of file block. There map between IN and BN displayed in the manner of physical position of IN is logical position of BN. The BV is Block Version the value of BV is incremented by 1 when insert, delete and append of file was take place. The Map Version table was maintained by verifier and below tables displays the clear explanation, the table 1 is initial only insertion is done. The table 2 contains the information after file block modified at positio3. The table 3 contains the information after the new file block inserted in the position 4. The table 4 contains the information after file block position 1 deleted.

International Journal of Advanced Scientific Technologies, Engineering and Management Sciences (IJASTEMS-ISSN: 2454-356X) *Volume.3,Special Issue.1,March.2017*

Table 1 Initial

IN	BN	BV
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1

Table 2 Block Modidifed At Position 3

IN	BN	BV
1	1	1
2	2	1
3	3	2
4	4	1
5	5	1

Table 3 Block Insertion At Position 4

IN	BN	BV
1	1	1
2	2	1
3	3	2
4	4	1
5	6	1
6	5	1

Table 4 Block Deletion At Position 1

IN	BN	BV
1	2	1
2	3	2
3	4	1
4	6	1
5	5	1

V. CONCLUSION

The many organization stores the data or file across multiple servers in cloud computing systems. The multiple copies stored on untrusted remote server, this models concentrate on file copy consistent. The proposed Dynamic File Block Modification, Insertion, Deletion and Append in Cloud Computing Systems models supports the data owner to store multiple file copies to remote servers with the help of CSPs. The main aim of CSP is to earn more money, so they can cheat the data owner. This model maintains verifier to check the file copy consistency. The verifier with the help of metadata (the small information about file) checks the file consistency. The model is dynamic allows users to insert, delete, modify and append blocks in particular file.

REFERENCES

[1] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY,USA, 2007, pp. 598–609.

[2] K. Zeng, "Publicly verifiable remote data integrity," in Proc. 10th Int. Conf. Inf. Commun. Secur. (ICICS), 2008, pp. 419–434.

[3] Y. Deswarte, J.-J. Quisquater, and A. Saïdane, "Remote integrity checking," in Proc. 6th Working Conf. Integr. Internal Control Inf. Syst. (IICIS), 2003, pp. 1–11. [4] D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfer," IACR (International Association for Cryptologic Research) ePrint Archive, Tech. Rep. 2006/150, 2006.

[5] F. Sebé, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. Knowl. Data Eng., vol. 20, no. 8, pp. 1034–1038, Aug. 2008.

[6] P. Golle, S. Jarecki, and I. Mironov, "Cryptographic primitives enforcing communication and storage complexity," in Proc. 6th Int.

[7] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proc. 11th USENIX Workshop Hot Topics Oper. Syst. (HOTOS), Berkeley, CA, USA, 2007, pp. 1–6.

[8] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," IACR Cryptology ePrint Archive, Tech. Rep. 2008/186, 2008.

[9] E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases," ACM Trans. Storage, vol. 2, no. 2, pp. 107–138, 2006. [10] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. 4th Int. Conf. Secur. Privacy Commun. Netw. (SecureComm), New York, NY, USA, 2008, Art. ID 9.

[11] C. Wang, Q. Wang, K. Ren, and W. Lou. (2009). "Ensuring data storage security in cloud computing," IACR Cryptology ePrint Archive, Tech. Rep. 2009/081. [Online]. Available: http://eprint.iacr.org/

[12] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2009, pp.213–222.