VERIFIABLE MULTI-COPY DYNAMIC DATA CONTROL IN CLOUD COMPUTING SYSTEMS

[1]Nischitha G S, [2]Pushpalatha R
[1]Student, [2] Asst. Professor
[1] Department of Computer Science and Engineering
[1]Visvesvaraya Technological University Department of PG Studies, Regional Office
Mysuru, Karnataka, India

Abstract : **Gradually more associations are picking outsourcing information to remote cloud service providers (CSPs). Clients can lease the CSPs stockpiling framework to store and recover practically unlimited measure of information by paying expenses metered in gigabyte/month. For an expanded level of versatility, accessibility, and solidness, a few clients may need their information to be imitated on different servers over various server farms. The more duplicates the CSP is requested that store, the more expenses the clients are charged. Hence, clients require to have a solid certification that the CSP is putting away all information duplicates that are settled upon in the administration contract, and every one of these duplicates are predictable with the latest changes issued by the clients. In this paper, we propose a map-based provable multi-copy dynamic data possession (MB-PMDDP) conspire that has the accompanying components: 1) it gives a proof to the clients that the CSP is not deceiving by putting away less duplicates; 2) it underpins outsourcing of element information, i.e., it underpins piece level operations, for example, square adjustment, addition, erasure, and affix; what's more, 3) it permits approved clients to consistently get to the record duplicates put away by the CSP. We give a near examination of the proposed MB-PMDDP conspire with a reference show acquired by augmenting existing provable ownership of element single-duplicate plans. The hypothetical investigation is approved through test comes about on a business cloud stage. Furthermore, we appear the security against intriguing servers, and examine how to recognize defiled duplicates by marginally changing the proposed conspire.**

***Index Terms*- Provable data possession(PDP), storage security, Cloud Service Provider(CSP), Cloud Computing, Dynamic Data, Data Integrity, Multi-copy.**

I. INTRODUCTION

The cloud service provider has taken up many ventures to attract the users to land into their stage. In order to accomplish this, the data guarantee and reliability is the prime concern to the users. The main service models offered by the cloud computing are

- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)
- Software as a Service (SaaS)

All the services offered from the cloud and all the data that are stored in the cloud servers requires protection features in order to trust the Cloud Service Provider (CSP). However SaaS demands more security. The data owners are always are at their limits if their data is being used as private datasets or if it is being sent to the third party without approval. Thus there exists cyber-trust demands in cloud services. The cloud storage has gained popularity for its low maintenance cost and due to their on-demand services among both the individual users and the corporate users. There are both merits and demerits in the cloud storage systems. The cloud storage enables the user to reduce the cost and resource of local data storage. The demerits of the cloud storage can be

- Loss of physical control
- Possibility of data theft

Provable Multicopy Dynamic data Possession in cloud computing deals with stored data in Dynamic way to cloud server. Multicopy means, data to be copied in multiple server. When data is transferred to cloud service provider that may not at all truthful, data owner lose exact control through there sensible data. This absence of control raising new fearsome and difficult task associated to data privacy and reliability protection in cloud computing. The confidentiality problem can be treated by encrypting vulnerable data since outsourcing to remote server. It is essential need of customer to have strong assurance that the cloud server still possess their data and it is not being change with or partly deleted after a while. Therefore, numerous scientists have concentrated on the issue of provable data possession (PDP) and proposed distinctive plans to review the information put away on remote servers.PDP is a method for approving information reliability over remote servers.

PDP is a method for approving information honesty over remote servers. In a run of the mill PDP demonstrate, the information proprietor produces some metadata/data for an information record to be utilized later for check purposes through a test reaction convention with the remote/cloud server. The proprietor sends the record to be put away on a remote server which might be un-trusted, what's more, erases the nearby duplicate of the record. As a proof that the server is as yet having the information record in its unique frame, it needs to effectively figure a reaction to a test vector sent from a verifier — who can be the first information proprietor or a trusted substance that shares some data with the proprietor. Specialists have proposed distinctive varieties of PDP conspires under various cryptographic suppositions; for case, see [1]–[9].

II. RELATED WORK

Provable data possession at untrusted store [2] In this it proposed model for provable data possession that enable a client that has stored data at untrusted server to confirm that server possesses the actual data without retrieving. It concentrated on issue of verifying server stores client data. It presents a model for provable data possession in which it is useful to reduce the file block approach, computation on the server and client –server communication. Data possession and un cheatable data transfer [5] It defines a protocol depends on hash function which avoids cheating in a data transfer transaction while allocating little burden on trusted third party that controls the protocol. It also specify a cryptographic protocol based on this principle, along which prover can demonstrate possession of arbitrary set of data known to the prover. Efficient remote data possession checking in critical information infrastructures [6] It introduced a new remote data possession checking protocol it permit an unlimited number of file integrity verification and its maximal running time can be select at set-up time and traded off across storage at verifier. Remote data possession checking protocol allow examining that a remote server can approach uncorrupted file in such manner that the verifier does not want to realize before all file is being checked. Scalable and efficient provable data possession [11] this scheme is to provide integrity of outsourced data in multi-cloud environments. To accomplish this it building the use of ranking and confirmable responds. This is built on idea of zero knowledge interactive proof system that can avoid different attacks over cloud. Dynamic provable data possession [13] it represents definitional framework structure and effective construction for dynamic provable data possession, which expand the PDP model to keep provable update to stored data. It manage a new category of authenticated dictionaries establish on rank information. and not as an independent document. Please do not revise any of the current designations.

III. MOTIVATION

In this paper a map-based provable multi-copy dynamic data possession (MB-PMDDP) scheme is proposed . This scheme provides an adequate guarantee that the CSP stores all copies that are agreed upon in the service contract. Moreover, the scheme supports outsourcing of dynamic data, *i.e.*, it supports block-level operations such as block modification, insertion, deletion, and append. The authorized users, who have the right to access the owner's file, can seamlessly access the copies received from the CSP. We give a thorough comparison of MB-PMDDP with a reference scheme, which one can obtain by extending existing PDP models for dynamic single-copy data. We show the security of our scheme against colluding servers, and discuss a slight modification of the proposed scheme to identify corrupted copies.

IV. EXISTING SYSTEM

In Figure 4 existing system the uploaded data are stored in single copy way. Then authorized users send the file directly to the file owner, this is not the correct way and service provider may be access files illegally. So automatically data security loss

Existing System Disadvantages:

- It uses Single copy
- Service Provider may be Hack the data and authorized users not send file request to file owner

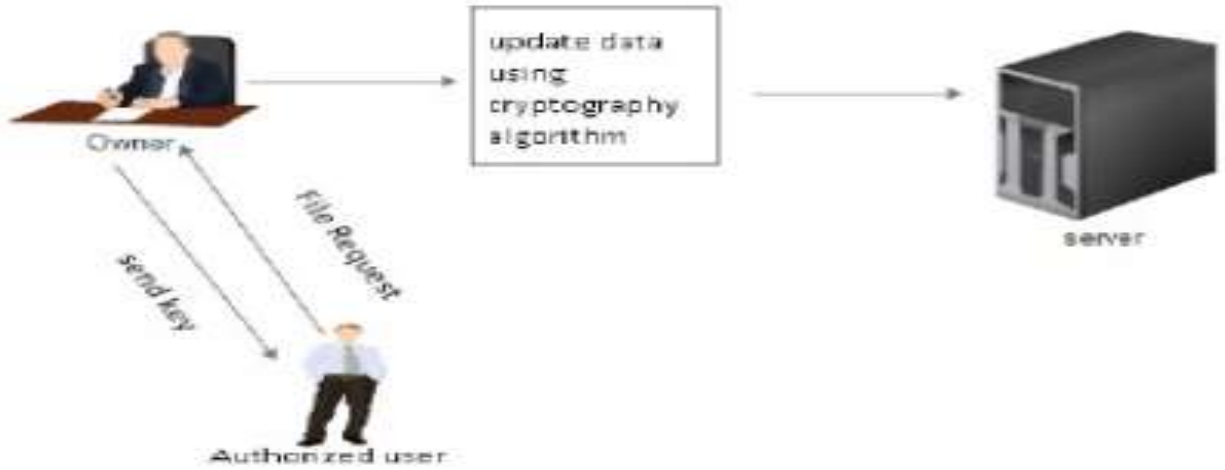

Figure 4 : Existing system architecture

V. PROPOSED SYSTEM

The proposed MB-PMDDP scheme allowing the data owner to update and scale the blocks of files copies outsourced to cloud servers which may be untrusted Validating such copies of dynamic data requires the knowledge of the block versions to ensure that the data blocks in all copies are consistent with the most recent modifications issued by the owner.The proposed scheme incorporates Blowfish ciphering for encryption and Secure Hash algorithm based SHA1 for authentication.

Proposed System Advantages:

- Multi-copy Data reduce access time and communication cost for user.
- If one copy is corrupted it will be redirected to another server and the file can be downloaded

VI. SYSTEM ARCHITECTURE

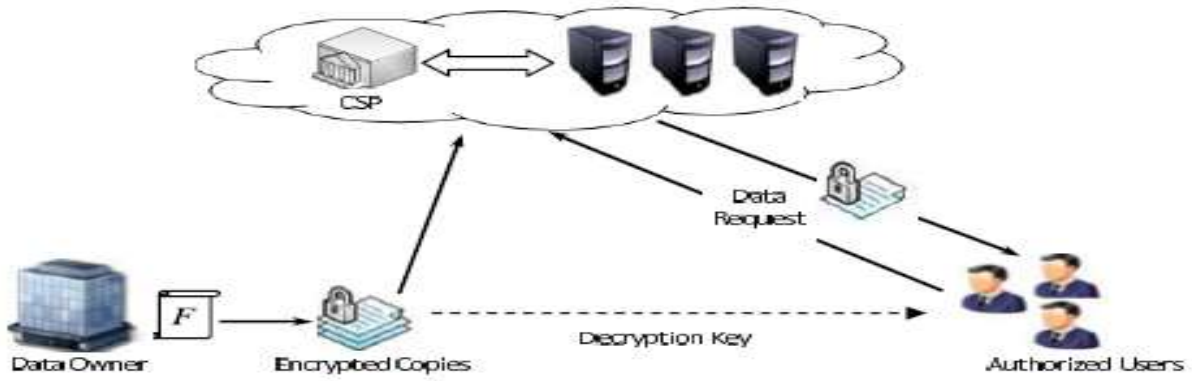

Figure 5 :System Architecture

System Components

Data Owner:

That can be an organization or an individual originally possessing sensitive data to be store in the cloud.

CSP:

Who manages Cloud Servers (CSs) and provide paid storage space on its infrastructure to stores files.

Authorized Users:

Authorized users, these users only to download the file from the others users. These users to send the file request to admin and file owner.

The system consist of data owner, cloud service provider, authorized user. Data owner can be organization basically exhibit sensitive data to be store in cloud. Cloud service provider which control cloud server and distribute paid storing area on its framework to store owner's file. Authorize user these are group of owner's client which is desirable to obtain information. This system framework accepted by many experimental implementations. To give an example such as e-Health application can be considered by this model where patient database that involve huge and sensitive data can be stored on cloud server. In this e-Health organization examined as data owner and physician as authorized user which have right to acquire patient medical record. It depends on handling small data structure called map-version table. Map-version table consists of three columns serial number(SN), block number (BN) and the block version (BV).The SN indicate the physical location of block in data file. The BN indicate logical indexing/numbering. The relationship between serial number and block number can be considered as mapping among logical number and physical position. The BV indicates current version of file blocks. Although data file is at beginning created block version of all block is 1. When certain block is updated block version is increased by 1.

VI. METHODLOGY

The data owner has a file F consisting of m blocks and the CSP offers to store n copies { F1,F2, ……..,Fn} of the Owner's file on different servers — to prevent simultaneous failure of all copies — in exchange of pre-specified fees in the form of GB/month. For data privacy, the owner encrypts their data before outsourcing to CSP. After outsourcing all n copies of the file, the owner may work together with the CSP to carry out block-level functions on all copies. These functions contains alter, insert, append, and remove specific blocks of the outsourced data copies. An authorized user of the outsourced data throws a data-access request to the CSP and accepts a file copy in an encrypted form that can be decrypted using a secret key shared with the owner. Validating such copies of dynamic data requires the knowledge of the block versions to ensure that the data blocks in all copies are consistent with the most recent modifications issued by the owner.
Furthermore, the verifier should be aware of the block indices to guarantee that the CSP has inserted or added the new blocks at the requested positions in all copies.

we propose a MB-PMDDP scheme allowing the data owner to update and scale the blocks of file copies outsourced to cloud servers which may be untrusted. Validating such copies of dynamic data requires the knowledge of the block versions to ensure that the data blocks in all copies are consistent with the most recent modifications issued by the owner. Furthermore, the verifier should be aware of the block indices to guarantee that the CSP has inserted or added the new blocks at the requested positions in all copies. To this end, the proposed scheme is based on using a small data structure (metadata), which we call a map-version table.

The integrity of customers' data in the cloud may be at risk due to the following reasons. First, the CSP — whose goal is likely to make a profit and maintain a reputation — has an incentive to hide data loss (due to hardware failure, management errors, various attacks) or reclaim storage by discarding data that has not been or is rarely accessed. Second, a dishonest CSP may store fewer copies than what has been agreed upon in the service contact with the data owner, and try to convince the owner that all copies are correctly stored intact. Third, to save the computational resources, the CSP may totally ignore the data-update requests issued by the owner, or not execute them on all copies leading to inconsistency between the file copies. The goal of the proposed scheme is to detect (with *high probability*) the CSP misbehavior by validating the number and integrity of file copies.
Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Modules:

The system is proposed to have the following modules along with functional requirements.

1. Owner registration
2. User registration
3. Data encryption and uploading
4. Data downloading and decryption
5. Encryption evolution management

Data encryption and uploading

The Owner first encrypts the data based on the Owner's sub ACPs in order to hide the content from the Cloud and then uploads them along with the public information generated by the KeyGen algorithm and The Cloud in turn encrypts the data based on the keys generated using its own KeyGen algorithm. KeyGen at the Cloud takes the secrets issued to Users and the sub CSPs given by the Owner into consideration to generate keys.

Data downloading and Decryption

Users download encrypted data from the Cloud and decrypt twice to access the data. First, the Cloud generated public information tuple is used to derive the OLE key and then the Owner generated public information tuple is used to derive the ILE key using the AB-GKM::KeyDer algorithm. These two keys allow a User to decrypt a data item only if the User satisfies the original ACP applied to the data protected.

Encryption Evolution Management

Over time, either ACPs or user credentials may change. Further, already encrypted data may go through frequent updates. In such situations, data already encrypted must be re-encrypted with a new key. As the Cloud performs the access control enforcing encryption, it simply re-encrypts the affected data without the intervention of the Owner.

VII. CONCLUSION

We have proposed a new PDP scheme (referred to as MB-PMDDP), which supports outsourcing of multi-copy dynamic data, where the data owner is capable of not only archiving and accessing the data copies stored by the CSP, but also updating and scaling these copies on the remote servers. To the best of our knowledge, the proposed scheme is the first to address *multiple* copies of *dynamic* data. The interaction between the authorized users and the CSP is considered in our scheme, where the authorized users can seamlessly access a data copy received from the CSP using a single secret key shared with the data owner. Moreover, the proposed scheme supports public verifiability, enables arbitrary number of auditing, and allows *possession-free* verification where the verifier has the ability to verify the data integrity even though he neither possesses nor retrieves the file blocks from the server.
Through performance analysis and experimental results, we have demonstrated that the proposed MB-PMDDP scheme outperforms the TB-PMDDP approach derived from a class of dynamic single-copy PDP models. The TB-PMDDP leads to high storage overhead on the remote servers and high computations on both the CSP and the verifier sides. A slight modification can be done on the proposed scheme to support the feature of identifying the indices of corrupted copies. The corrupted data copy can be reconstructed even from a complete damage using duplicated copies on other servers. Through security analysis, we have shown that the proposed scheme is provably secure.

VIII. ACKNOWLEDGMENT

I would like to thank Dr.Thippeswamy Ph.d., Professor & Chairman, Dept of studies in computer science &engineering and Ms.Pushpalatha R M.tech., Assistant professor, Dept of studies in computer science &engineering, vtu regional office, mysuru and anonymous reviewers encouragement and constructive piece of advice that have prompted us for new round of rethinking of our research, additional experiments and clear presentation of technical content.

REFERENCES

[1] F. Barsoum and M. A. Hasan.(2015) ―Provable Multicopy Dynamic Data Possession in Cloud Computing Systems‖ IEEE transaction.
[2] A. F. Barsoum and M. A. Hasan. (2010). ―Provable possession and replication of data over cloud servers,‖ Centre Appl. Cryptograph. Res., Univ. Waterloo, Waterloo, ON, USA, Tech. Rep. 2010/32.
[3] A. F. Barsoum and M. A. Hasan. (2011). ―On verifying dynamic multiple data copies over cloud servers,‖ IACR Cryptology ePrint Archive, Tech. Rep. 2011/447.
[4] 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
[5] 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].
[6] 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.
[7] 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.
[8] K. Zeng, ―Publicly verifiable remote data integrity,‖ in Proc. 10th Int. Conf. Inf. Commun. Secur. (ICICS), 2008, pp. 419–434.
[9] 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.

[10] 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.
[11] Amazon Elastic Compute Cloud (Amazon EC2). Available:http://aws.amazon.com/ec2/, accessed Aug. 2013.
[12] Amazon Simple Storage Service (Amazon S3).Available: http://aws.amazon.com/s3/, accessed Aug. 2013.
[13] Amazon EC2 Instance Types. [Online]. Available:http://aws.amazon.com/ec2/, accessed Aug. 2013.
[14] P. S. L. M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in Proc. 12th Int. Workshop SAC, 2005, pp. 319–331.
[15] A. L. Ferrara, M. Green, S. Hohenberger, and M. Ø. Pedersen, "Practical short signature batch verification," in Proc. Cryptograph. Track RSA Conf., 2009, pp. 309–324.
[16] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Efficient provable data possession for hybrid clouds," in Proc. 17th ACM Conf. Comput. Commun. Secur. (CCS), 2010, pp. 756–758.
[17] Kochumol Abraham, Win Mathew John, "Proving Possession and Retrievability within a Cloud Environment: A Comparative Survey", 2014.
[18] Yihua Zhang and Marina Blanton," Efficient Dynamic Provable Possession of Remote Data via Balanced Update Trees", 2013.
[19] Kevin D. Bowers, Ari Juels, Alina Oprea, Proofs of Retrievability: Theory and Implementation, CCSW'09, Journal of Systems and Software, v.85 n.5, p.1083-1095, May, 2012.
[20] Malathi.M, Murugesan. T, A Scheme for Checking Data Correctness in the Cloud, 2012 International Conference on Information and Network Technology (ICINT 2012) IPCSIT vol. 37 (2012).
[21] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR- PDP: Multiple-Replica Provable DataPossession," in 28th IEEE ICDCS, 2008, pp. 411–420.
[22] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in ESORICS'09: Proceedings of the 14th European Conference on Research in Computer Security, Berlin, Heidelberg, 2009, pp. 355–370.
[23] M. P. V. Sontakke and M. A. A. Manjrekar, "Provable Multi-Copy Dynamic Data Possession With Multi-Owner In Cloud Computing System."
[24] A. Mohan and R. Katti, "Provable Data Possession Using Sigma-protocols," in IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 2012, pp. 565-572.
[25] M. S. Bajwa, "A Concern towards Data Security in Cloud Computing," International Journal of Computer Applications, vol. 114, 2015.