



# **An Proficient and Sheltered Statistics Storage in Mobile Cloud Computing through RSA and Hash Function**

V.Shobana Devi<sup>1</sup>, R.Suguna<sup>2</sup>

Research Scholar, Dept. of CS, PGP College of Arts & Science, Namakkal, Tamilnadu, India<sup>1</sup>

Assistant Professor, Dept. of CS, PGP College of Arts & Science, Namakkal, Tamilnadu, India<sup>2</sup>

**ABSTRACT:** Cloud computing is the way of providing computing resources in the form of services over internet. The cloud computing allows storing the user's data and to measure the applications and services provided by cloud server. There is an ample data stored at cloud storage server. Security is one of the major issues which reduces the growth of cloud computing. So Cloud computing entails encyclopedic security solutions. This paper presented secure file exchanging on Cloud using Blowfish, RSA algorithms which is capable of solving data security, authentication, and integrity problems of files on the cloud. Data security is improved by cryptography algorithms. The rightness of data is verified by introducing techniques. Enhanced system (RSA value) compares with simple RSA and Blowfish on basis of some performance parameters like:- throughput, encryption time, cipher text and delay time. In our enhanced system we integrate symmetric, asymmetric and algorithms which provide better results for performance parameters. TPA which has to match the code for the integrity of user data in cloud on behalf of Data Owners. Data Owner can get notification from TPA when the data integrity is lost. If any unauthorized person access the data in cloud, they will be blocked.

**KEYWORDS:** cloud storage server, cloud computing, TPA.

## **1. INTRODUCTION**

Cloud storage becomes an increasing attraction in cloud computing paradigm, which enables users to store their data and access them wherever and whenever they need using any device in a pay-as-you-go manner. Moving data into cloud offers great conveniences to the users since they do not have to care about the large capital investment in both the maintenance and management of the hardware infrastructures. Amazon's Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3) and apple icloud are well known examples of cloud data storage. However, once data goes into cloud, the users lose the control over the data. This lack of control raises new formidable and challenging issues related to confidentiality and integrity of data stored in cloud. The confidentiality and integrity of the outsourced data in clouds are of paramount importance for their functionality. The reasons are listed as follows: 1) the CSP, whose purpose is mainly to make a profit and maintains a reputation, has intentionally hide data loss an incident which is rarely accessed by the user's 2) The malicious CSP might delete some of data or is able to easily obtain all the information and sell it to the biggest rival of Company. 3) An attacker who intercepts and captures the communications is able to know the user's sensitive information as well as some important business secrets. 4) Cloud infrastructures are subject to wide range of internal and external threats. The examples of security breaches of cloud service providers appear from time to time. The users require that their data remain secure over the CSP and they need to have a strong assurance from the cloud servers that CSP store their data correctly without tampering or partially deleting because the internal operation details of service providers may not be known to the cloud users. Thus, an efficient and secure scheme for cloud data storage has to be in a position to ensure the data integrity and confidentiality. Encrypting the data before storing in cloud can handle the confidentiality issue. However, verifying integrity of data is a difficult task without having a local copy of data or retrieving it from the server. Due to this reason, the straightforward cryptographic primitives cannot be applied directly for protecting outsourced data. Besides, a naive way to check the data integrity of data storage is to download the stored data in



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

order to validate its integrity, which is impractical for excessive I/O cost, high communication overhead across the network and limited computing capability. Therefore, efficient and effective mechanisms are needed to protect the confidentiality and integrity of user's data with minimum computation, communication and storage overhead. Remote data integrity checking is a protocol that focuses on how frequently and efficiently we verify whether cloud server can faithfully store the user's data without retrieving it. In this protocol, the user generates some metadata. Later, he can challenge the server for integrity of certain file blocks through challenge-protocol. Then the server generates responses that the server still possesses the data in its original form to corresponding challenge sent by the verifier who may be original user or trusted third party entity. Recently, several researchers have proposed different variations of remote data integrity checking protocols under different cryptography schemes. However, all these protocols focus on static data verification. One of the design principles of cloud storage is to provide dynamic scalability of data for various applications. This means, the data stored in cloud are not only accessed by the users but also frequently updated through block operations such as modification, insert and delete operations. Hence, it is crucial to develop more secure and efficient mechanism to support dynamic audit services. The protocols to verify dynamic data in cloud are proposed. Although the existing schemes aim at providing integrity verification for different data storage systems, but problem of confidentiality of data has not been fully addressed. The protocols have been proposed to ensure the confidentiality and integrity of remote data. But, all these schemes are unable to provide strong security assurance to the users, because these schemes verifying integrity of outsourced data based on pseudorandom sequence, which does not cover the whole data while computing the integrity proof. Therefore, probabilistic verification schemes based on pseudorandom sequence does not give guarantee to the users about security of their data. Syam et al. proposed a distributed verification protocol using Sobol sequence to ensure availability and integrity of data, but it is also not addressed the data confidentiality issue. How to achieve a secure and efficient design to seamlessly integrate these two important components for data storage service remains an open challenging task in Cloud Computing.

## II. RELATED WORK

Under The security of remote storage applications has been increasingly addressed in the recent years, which has resulted in various approaches to the design of storage verification primitives. The literature distinguishes two main categories of verification schemes. Deterministic verification schemes check the conservation of a remote data in a single, although potentially more expensive operation and probabilistic verification schemes rely on the random checking of portions of outsourced data.

### Deterministic Secure Storage

Deterministic solutions are verifying the storage of the entire data at each server. Deswarte et al. and Filho et al. are firstly proposed a solution to remote data integrity. Both use RSA-based functions to the whole data file for every verification challenge. They require pre-computed results of challenges to be stored at verifier, where a challenge corresponds to the ing of the data concatenated with a random number. However, both of them are inefficient for the large data files, which need more time to compute and transfer their values. Carmoni et al. described a simple deterministic approach with unlimited number of challenges is proposed, where the verifier like the server is storing the data. In this approach, the server has to send MAC of data as the response to the challenge message. The verifier sends a fresh unique random value as the key for the message authentication code to prevent the server from storing only the result of the previous ing of the data. Golle et al. proposed a SEC (Storage Enforcing Commitment) deterministic verification approach. This approach uses homomorphic verifiable tags, whose number is equal to two times of number of data chunks and the verifier choose a random value that will be used to shift indexes of tags to be associated with the data chunks when the integrity proof constructed by the server. Sebe et al. presented a remote data checking protocol such that it allows an unlimited number of verifications and the maximum running time can be chosen at setup time and traded off against storage at verifier. However, none of the schemes were considered the problem of remote data confidentiality and dynamic data verifications.

### Probabilistically Secure Storage

Ateniese et al. proposed a RDC using PDP. In their system, the client pre-computes the tags for each block of a file using homomorphic verifiable tags and stores the file and its tags with the server. Then, the client can verify that



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

server integrity of the file by generating a random challenge, which specifies the selected positions of file blocks. Using the queried blocks and their corresponding tags and the server generates a proof of integrity. Juels et al. proposed a formal definition of POR and its security model. In this model, the encrypted data is being divided into small data blocks, which are encoded with Reed-Solomon codes. The “sentinels” are embedded among encrypted data blocks to detect whether it is intact. However, this can verify only limited number of times because this scheme has only finite number of “sentinels” in the file. When the finite “sentinels” are exhausted, the file must be sent back to the owner to re-compute new “sentinels”. Ateniese et.al proposed a new scheme with homomorphic linear authenticators (HLA) of which communication complexity is independent of the file length. This scheme supports unlimited number of verification, but it cannot verify publicly. Later, Shacham et al. proposed the two POR protocols: The first one built from BLS signatures and has the shortest query and response with public verifiability. The second one is based on pseudorandom functions (PRFs) with private verifiability, but it requires a longer query. Both schemes rely on the homomorphic property-aggregating verification proofs into a small value. Dodis et al first formally define the POR code, this construction improves the prior POR constructions. The main insight of their work comes from the simple connection between POR schemes and the notion of hardness amplification, extensively studied in complexity theory. Browsers et al.introduced a theoretical framework for previous POR protocols using integrated forward error-correcting codes. In their subsequent work, Browsers et al described a HAIL (High-Availability and Integrity Layer), in which the key insight is to embedded MACs in the parity blocks of the dispersal code. As both MACs and parity blocks can be based on universal functions. Schwarz et al. used a XOR-based, parity  $m/n$  erasure codes to create  $n$  shares of a file that stored at multiple sites. Curtomola et al. extended the PDP to the multiple servers, which are called Multiple Replica-Provable Data Possession (MR-PDP), it is aimed to ensure availability and reliability of data across distributed servers. In this scheme, the user stores multiple replicas of a single file across distributed servers, thus we can get an original file from any one of the servers even if any server fails.

## EXISTING SYSTEM

The existing schemes aim at providing integrity verification for different data storage systems, but problem of confidentiality of data has not been fully addressed. All these schemes are unable to provide strong security assurance to the users, because these schemes verifying integrity of outsourced data based on pseudorandom sequence, which does not cover the whole data while computing the integrity proof.

### 2.1 DISADVANTAGES OF EXISTING SYSTEM

#### Data Security:

Traditionally sensitive data stored within organization boundary, but in cloud enterprise data is stored outside the enterprise boundary, which required strong encryption.

#### Cloud Privacy and Confidentiality:

Confidentiality is defined as the sensitive data is not disclosed to unauthorized process, devices and person .cloud service provider knows where the user’s private data located in the cloud.

#### Data location and relocation:

cloud storage is black box. Consumers always don’t know the location of their data. Cloud computing offers high degree of data mobility.

#### Storage, Backup and Recovery:

When customers move their data on cloud the cloud service provider ensure adequate data resilience storage system. Cloud storage providers will manage the data in multiple copies across many independent locations.

#### Data Integrity:

Data integrity is the rightness of data stored at cloud. The alterations between two updates of a data record violate the data integrity.

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

## III. PROPOSED SYSTEM

Our proposed work used the concept of function along with several cryptographic tools to provide better security to the data stored on the cloud. TPA checks the integrity of the data stored on cloud on behalf of the data owner. TPA checks the of the message to verify the integrity of the data. The Integrity Verification is provided by the TPA which reduces a lot of work of the data owner. In our proposed system Blowfish and RSA cryptography algorithms are used to integrate the features from both and make it a better system than existing system. . In our proposed system we focus on integrity detection and integrity prevention mechanism implemented using some locking techniques.

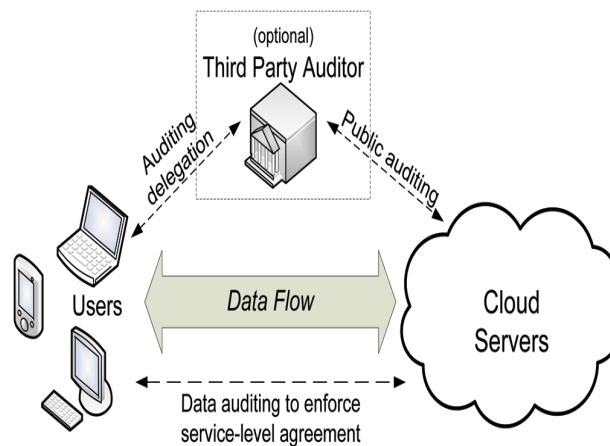


Fig 1. System Overview.

### ADVANTAGES OF PROPOSED SYSTEM:

- It should detect all data corruption if anybody deletes or modifies the data in cloud storage, since we are using Sobol sequence instead of pseudorandom sequence for challenging the server for the integrity verification.
- Our scheme achieves the confidentiality of data
- It is efficient in terms of computation, storage, because its key size is low compared to RSA based solutions.

## IV. ALGORITHMS

### RSA

RSA(Rivest Shammir Adleman) is public key cryptography algorithm involves two different keys. Public key for encryption and private key for decryption. RSA also provide authentication . RSA Algorithm RSA is a commonly adopted public key cryptography algorithm. RSA can be used for key exchange, digital signatures, or encryption of small blocks of data. RSA uses a variable size encryption block and a variable size key. RSA has been widely used for establishing secure communication channels and for authentication and the identity of service provider over insecure communication medium. file. The RSA algorithm involves three steps:

- 1) Key generation,
- 2) Encryption and
- 3) Decryption.

### Key generation

RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated the following way:

Choose two distinct prime numbers  $p$  and  $q$ . For security purposes, the integers  $p$  and  $q$  should be chosen at random. Compute  $n = p \cdot q$ .  $n$  is used as the modulus for both the public and private keys. Its length, usually expressed in

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

bits, is the key length. Compute  $\phi(n) = \phi(p)\phi(q) = (p-1)(q-1)$ , where  $\phi$  is Euler's totient function. Choose an integer  $e$  such that  $1 < e < \phi(n)$  and  $\gcd(e, \phi(n)) = 1$ ; i.e.  $e$  and  $\phi(n)$  are coprime.  $e$  is released as the public key exponent. Determine  $d$  as  $d \cdot e \equiv 1 \pmod{\phi(n)}$ , i.e.,  $d$  is the multiplicative inverse of  $e$  (modulo  $\phi(n)$ ). This is more clearly stated as solve for  $d$  given  $d \cdot e \equiv 1 \pmod{\phi(n)}$ .  $d$  is kept as the private key exponent. By construction,  $d \cdot e \equiv 1 \pmod{\phi(n)}$ . The public key consists of the modulus  $n$  and the public (or encryption) exponent  $e$ . The private key consists of the modulus  $n$  and the private (or decryption) exponent  $d$ , which must be kept secret.  $p$ ,  $q$ , and  $\phi(n)$  must also be kept secret because they can be used to calculate  $d$ .

### Encryption

Alice transmits her public key  $(n, e)$  to Bob and keeps the private key secret. Bob then wishes to send message  $M$  to Alice. He then computes the ciphertext  $c$  corresponding to:  $C \equiv m \pmod{n}$ . Bob then transmits  $c$  to Alice.

### Decryption

Alice can recover  $m$  from  $c$  by using her private key exponent  $d$  via computing:  $m \equiv c^d \pmod{n}$ . Given  $m$ , she can recover the original message  $M$  by reversing the padding scheme.

## V. IMPLEMENTATION

In our proposed work we compare Blowfish, RSA and integrate system (Blowfish+RSA) based on the performance parameters like:- throughput, cipher text, encryption text and delay time. Formulas for these parameters as following :

### 1) Throughput:

It is number of bits transferred per unit time. Its unit is byte/sec.

Formula: Throughput = uploaded file size ÷ Delay time.

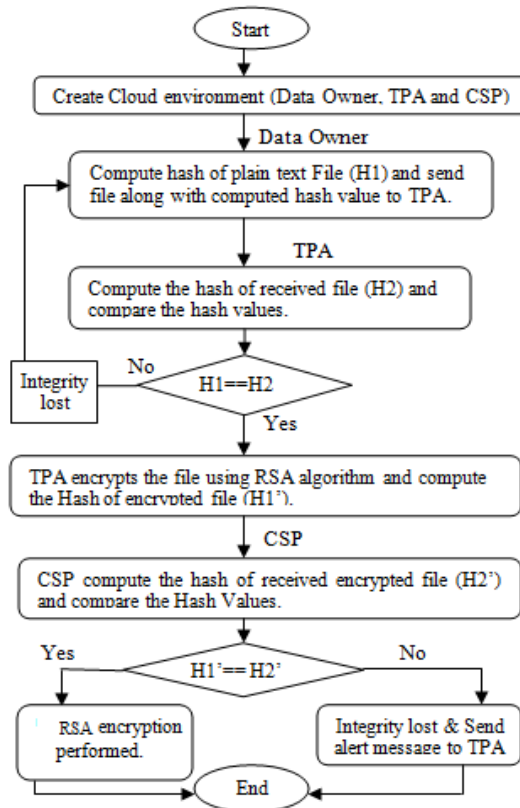


Fig 2: Proposed System flow chart

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

## 2) Ciphertext size:

It is length of encrypted data. Its unit is in bytes.

Formula : Ciphertext size = Length of encrypted data

## 3) Encryption time:

Time taken by server to encrypt any file. Its unit is nano seconds.

Formula : Encryption time = Encryption End Time - Encryption Start Time.

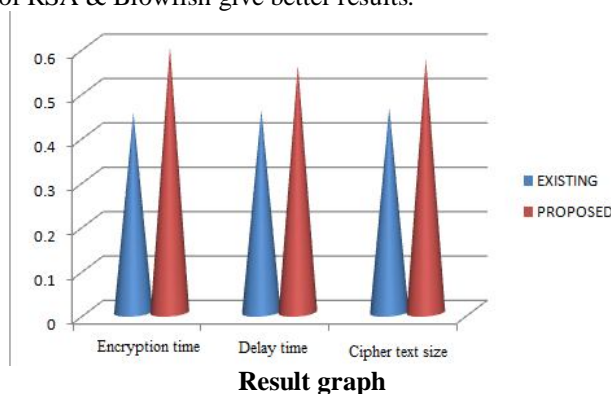
## 4) Delay Time:

Delay time is time difference between start uploading time and end uploading time. Its unit is in nano seconds.

formula : Delay Time = End Uploading Time - Start Uploading Time .

## VII. RESULT ANALYSIS

The criteria in performance of every algorithm is that encryption time ,delay time and cipher text size should be less and throughput should be more. During comparison of RSA, Blowfish and proposed system(RSA) this criteria is achieved and integrate system of RSA & Blowfish give better results.



## VIII. CONCLUSION AND FUTURE WORK

When a client stores its data on the cloud, there is always a big concern of whether the cloud service provider stores the file correctly or not. Security is the main concern in cloud computing. The proposed mechanism provides a security mechanism for securing the data in cloud computing with the help of Blowfish & RSA algorithms. This research paper has proposed a system to provide integrity, authentication and confidentiality to the data stored in cloud computing. Authentication is achieved because only registered client upload and download the files on the cloud. The proposed scheme used the combination of blowfish and RSA to secure the data in such a way that no leakage of data on cloud could be performed. Always encrypted file stored on the cloud. value matches at TPA server to check the integrity of file. There is always a scope for improvement in every field of work, so here also. We take one of the assumption made in all the models of security are that the TPA is neutral. So there is some need to do some work for making TPA more secure.





# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

## REFERENCES

- [1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems*, vol. 25, no. 6, June 2009, pp 599–616.
- [2] Amazon.com, "Amazon Web Services (AWS)," Online at <http://aws.amazon.com>, 2008.
- [3] Apple "iCloud" Online at <http://www.apple.com/icloud/what-is.html> 2010.
- [4] T Mather, S Kumaraswamy, and S Latif "Cloud Security and Privacy", O'REILLY Publication, first edition, sep- 2009.
- [5] H. Takabi, J.B.D. Joshi, and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments", *Article in IEEE Security and Privacy*, vol. 8, no.6, Nov- Dec. 2010, pp. 24-31.
- [6] N. Gohring, "Amazon's S3 down for several hours," Online at [http://www.pcworld.com/businesscenter/article/142549/amazons\\_down\\_for\\_sever\\_hours.html](http://www.pcworld.com/businesscenter/article/142549/amazons_down_for_sever_hours.html), 2008 .
- [7] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors", Online at <http://www.techcrunch.com/2008/07/10/mediamaxthelinkup-closes-its-doors/>, July 2008.
- [8] Y. Deswarte, J.-J. Quisquater, and A. Saidane. "Remote integrity checking". In *Proc. of Conference on Integrity and Internal Control in Information Systems (IICIS'03)*, November 2003. lausanne, Switzerland.
- [9] D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating Data Possession and Uncheatable Data Transfer," *cryptology ePrint Archive*, Report 2006/150, 2006, <http://eprint.iacr.org/>.
- [10] G. Caronni and M. Waldvogel, "Establishing Trust in Distributed Storage Providers", In *Third IEEE P2P Conference*, Linkoping 03, 2003.
- [11] P. Golle, S. Jarecki and I. Mironov, "Cryptographic Primitives Enforcing Communication and Storage Complexity", In *proc. of Financial Crypto 2002*. Southampton, Bermuda.
- [12] F. Sebe, J. Domingo-Ferrer, and A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient Remote Data Possession Checking in Critical Information Infrastructures", *IEEE Trans. Knowledge and Data Engineering*, vol. 20, no. 8, aug-2008, pp. 1034-1038
- [13] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Remote Data Checking using Provable Data Possession," *ACM Trans. ACM Transactions on Information and System Security*, Vol. 14, No. 1, Article 12, may 2011, pp. 12.1–12.34.
- [14] A. Juels and J. Burton S. Kaliski, "PORS: Proofs of Retrievability for Large Files," *Proc. of CCS '07*, pp. 584– 597, 2007. Alexandria, Va, USA.
- [15] G Ateniese, S. Kamara, J. Katz, "Proofs of Storage from homomorphic identification protocols". In *Proc. of ASIACRYPT '09*, 2009, pp. 319-333. Tokyo, Japan.