International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-04, Issue-04, July 2018

An Efficient Identity Based Data Integrity Auditing Protocol for Secure Cloud Storage

***Misbah U.Mulla, [1]Prabhu R.Bevinamarad**

***PG Scholar, [1]Assistant Professor, Department of Computer Science and Engineering, B.L.D.E.A's V.P. Dr. P.G. Halakatti College of Engineering and Technology, Vijayapur, Karnataka, India,**

***misbahmulla786@gmail.com, [1]prabhubev@gmail.com**

Abstract - Due to rapid and fast development of cloud computing and its services more and more users have started to store their data remotely on cloud servers which helps in easy access to the information. The various security issues arise while processing the client data on the cloud such as the maintenance of data integrity and verification of the information for integrity check. In this research work, we propose a scheme for verification of the data integrity in cloud based on user's identity to prove that it is storing the data of the client honestly. The proposed scheme also uses homomorphic cryptographic primitive to reduce the complexity and cost for the creation and maintenance of public key framework. Further, the zero-knowledge privacy is achieved where the information is kept confidential from the verifier during the verification process.

Keywords – Cloud computing, Data integrity verification, user's identity, zero-knowledge privacy, confidential, verifier.

I. INTRODUCTION

Cloud computing nowadays is seeking more and more attention from many areas[1] such as industries, academics and so on it is due to its shared resources such as storage space, applications, services and processing capability. The benefits introduced by cloud computing are, the client can reduce the basic investment cost both on hardware and software resources, the users need not be involved in management task and are not required to stay nearby devices to access the data as they can access the required information if connected via a network. The owners who store their data on the cloud are very much concerned about their data loss as sometimes some part of their data will be discarded by the server without the knowledge of data owner so verifying whether the data is stored correctly and honestly by the server is an important task. Hence, the entity called third-party auditor is involved as an intermediator between the client and the server, the auditor on behalf of the data owner request the cloud server to prove the genuineness of the data which they are holding.

The paper is organized as follows. After reviewing the related work in section II, we discuss about existing system and its various drawbacks in section III, in section IV we introduce the proposed scheme, the details of implementation results is provided in section V and the final conclusion is drawn in section VI.

II. LITERATURE SURVEY

A. Survey on: Bidirectional verification for the security of storage in cloud: A statistical analysis supported by bidirectional verification[2] where the main control is given to the centralised authority and the certificates for other participating entities are provided by the centralised entity and the common platform checks and authenticate the verifier, it manages and keeps records of dynamic operations on files and it is proved to be secure in the random oracle model it also manages distribution of work and on user side there is less computation overhead and it involves low initial establishment cost.

B. Survey on: data integrity auditing based on fuzzy identity: The main aim of this approach was to overcome the issues of complex key management, in this number of attributes were combined to describe the identity of the user and the biometrics were involved in establishing the fuzzy identity[3] ,here the file and its metadata both are stored at server making the auditing process efficient and reliable.

C. Survey on: Distributed Data Possession in Cloud based on Identity: In this scheme the bilinear pairing is used and here credential generation is smoothly eliminated making it flexible, this protocol effectively perform delegated, public and private auditing. The file blocks along with its encrypted form are sent to the combiner[4] where the combiner forwards and distributes among different servers in this same fashion the challenge from the third party auditor is forwarded to all the servers via combiner and the results are send back to the data owner, this provides satisfactory results.

III. EXISTING SYSTEM

The existing methods for remote data integrity checking cannot efficiently verifies the indices of the blocks in

 DOI : 10.18231/2454-9150.2018.0555

public verifiability which may leads to replay attack, in this integrity check with public verifiability the information should not be leaked to the third party verifier as the data sometimes might be very private and should be held confidential and this issue of preserving the privacy of the data is not considered important in most of the techniques. In the privacy-preserving scheme, while data auditing[5]the whole blocks cannot be recovered from the cloud server response which may not strongly face dictionary attack, in the data integrity checking framework[6]they are evaluationally and computationally zero knowledge but it is also desirable to protect and keep the encrypted files confidential.

3.1 Disadvantages of the Existing system:

- Preserving the privacy of the data against the verifier is an important task otherwise it may lead to different types of attacks.
- The encrypted files are not held private.
- Some methods involve complex key management procedures.
- Some existing models are PKI(Public Key Infrastructure) based but does not work in identity-based infrastructure.
- Some approaches due to complex activities are time-consuming and expensive.
- Some schemes fail to achieve the property of soundness in case of challenged file blocks.

IV. PROPOSED SCHEME

The proposed scheme is user's identity based data integrity auditing protocol whose main aim is to help the data owners to verify the genuineness of their data on cloud .

The important contributions of our proposed scheme are as follows

1. In this identity-based scheme using the unique identity of the data owner, the verifier can verify the data integrity this public verifiability is more efficient and desirable.
2. This challenge-response protocol is a key agreement between the verifier and server and it is based on asymmetric group key agreement which makes this scheme more effective.
3. The main advantage of this scheme is it successfully achieves zero-knowledge privacy which keeps the data private and confidential and preserves its privacy against the third party verifier during the process of verification.
4. The evaluation involved in this scheme during the auditing operations are done on the cipher text which is the encrypted form of the data instead of plain text making it reliable and secure.

4.1 System Architecture

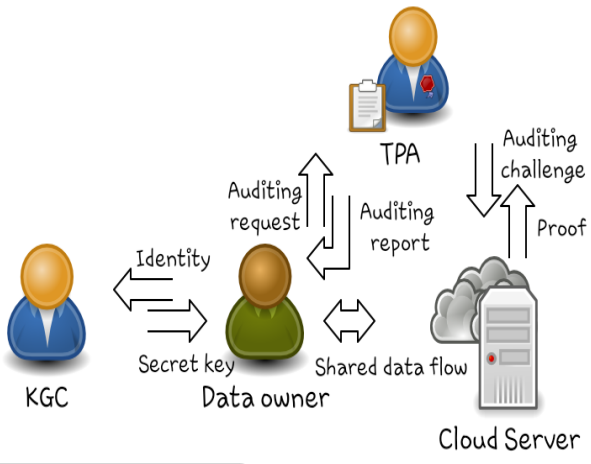

Figure 1:System Architecture

There are four involved entities in the system model as shown in fig 1 those are the key generation centre, third-party auditor, data owners and the cloud server.

- The cloud server is the one which provides various resources and storage space and allows the users to store their data in logical pools and helps in easy and user-friendly access of the data and runs the proof generation whenever verifier request the proof of the data.
- Data owners are the one who store their data on servers and using their identity they generate tags and upload their data onto the server and they verify the integrity of their information by generating a request to TPA.
- Key generation centre(KGC) generates the secret key using the system parameters, master secret key and the identity of the user, which is required for encryption and verification.
- Third-party auditor (TPA) is the one who accepts and process the data verification request from the owner and generates the challenge for the cloud server and upon receipting the response verifies and forwards the verification report to the data owner.

4.2 Preliminaries

1] (Bilinear pairing): It maps a couple of gathering components to another gathering component, Let Gp1 and Gp2 be the cyclic gatherings and gp1 and gp2 are their generators respectively than the function

$e: Gp1 \times Gp1 \rightarrow Gp2$ represents the bilinear pairing.

2](Equality of Discrete Logarithm): Let the modulus of the group G is equal to some prime number a, the protocol [7]

Helps the server to check that two components k1 and k2 have met the discrete logarithm to base gp1 and gp2 respectively.

 DOI : 10.18231/2454-9150.2018.0555

3](ID-Based Signature[8,9]): in this, the signature involves

setup(t)	where t is the security element
extract(ms,ID)	ms is the master secret key and ID is the users identity, based upon which the private key is created.
taggen(sk)	using the secret key the tags for the blocks are generated.
proofcheck(t,id,tag,)	using the security element, the owners identity and the tags the integrity check is done.

4.3 Algorithm/Technique Used

Digital Signature Algorithm (DSA): Is a federal information processing standard for generating the digital signatures and adopted as FIPS 186 in 1993.

In das the digest of the file block is signed using the private key where the signature is very much small compared to the information in its plain text form due to which less there is less amount of storage, less processing and evaluation load and small amount of bandwidth is required, the mathematical evaluations are involved in generation of tags as a digital signatures consisting of two 160 numbers from the private key and the message digest.

Important terms used

m: the message whose signature is to be generated

h(m): hash of message M by making use of SHA-1

(r, s): signature

Key Generation

Keys are generated generally to encrypt and decrypt the information in this the private key is used to signing the data which creates the tags of the file blocks, it is used only for signing and verification, it involves generation of two prime numbers p which is within the range of 512-1024 bits and q is of bit length which is multiple of 64 and p-1 is the multiple of q,

g is the generation such that g^q=1 mod p

The (p ,q, g) is shared among the involved entities of the system.

Signing

To generate the digital signature the hash function is required and the message whose signature is to be created the following equations are used to generate the signature

$r=(g^k mod p) mod q$

$s=(k^{-1}h(m)+x(r)) mod q$

where (r,s) denotes the signature.

Checking

The proof of the data is verified using the following equation

$v=((g^{u1}.y^{u2} mod\ p)\ mod\ q)$

where u1 is computed by using the value of hash and u2 is computed by making use of the value of message and finally compared and tested whether v=r .

4.4 Design Goals

The important design goals of our scheme are as follows:

1) **Key provider**: The data owners can get their secret key for tag generation based upon the unique identity easily and securely from the KGC without any certification procedures.
2) **Confidentiality of information**: The data is held secured by converting it into digital form and privacy of the data is preserved against the verifier by achieving zero-knowledge privacy.
3) **Access control**: The data is accessible only to the authorized users and they are able to use the services and perform related task and operations, the system users are checked by the process of authentication.
4) **Efficiency**: Here the verification is done precisely by the auditor using the data owners identity and each step of processing is done accurately, the owners can verify the integrity of their data without any complications and also the privacy of the information is maintained.

V. IMPLEMENTATION DETAILS

The system has four main entities the key generation centre, data owner, third-party auditor and the cloud server.

The data owner registers to the system and uploads the files on the cloud server, the secret key which is required for the tag generation of the blocks is provided by key generation centre in extract process as shown in fig 2.

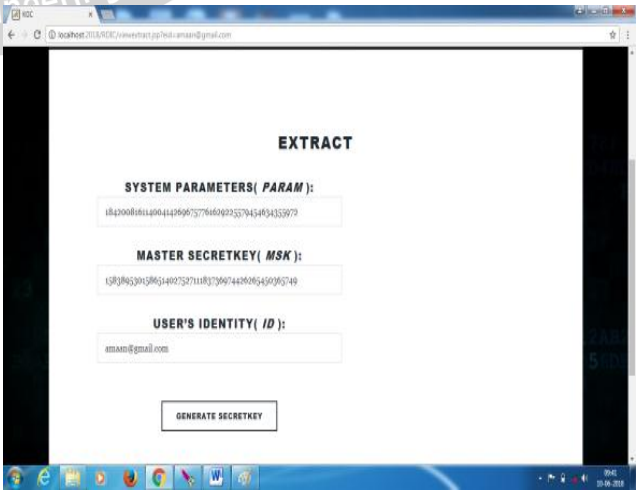

Figure 2 : Generation of the secret key

 DOI : 10.18231/2454-9150.2018.0555

Whenever the data owner wants to verify the integrity of their files it sends a verifying request to the auditor as shown in fig 3

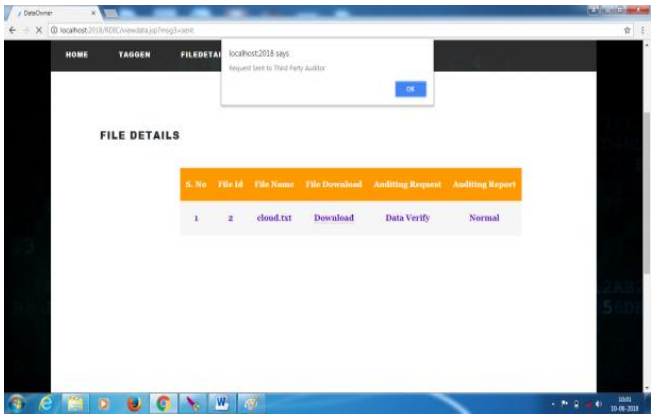

Figure 3 : Data integrity verification request

After receiving the request the TPA generates the challenge and forwards the challenge to the auditor as shown in fig 4.

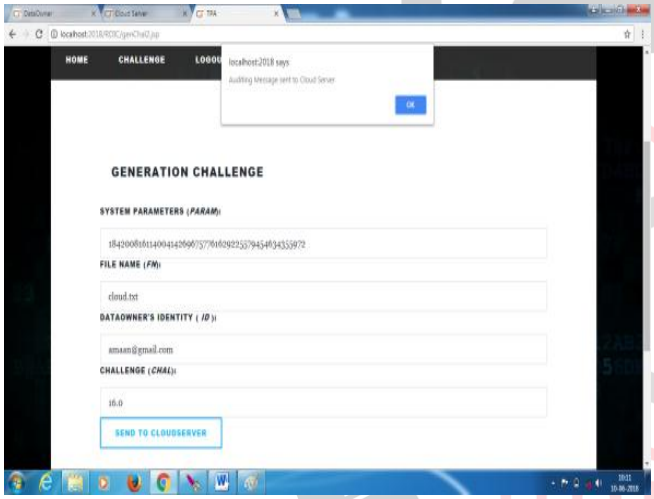

Figure 4: Challenge generation

The CS generates the proof of the file for which the verifier has demanded and sends that proof to the verifier as shown in fig 5.after this the TPA runs proof check function and verifies the proof and sends the report to the data owner.

Figure 5: Proof generation

The data owner receives the auditing report as shown in fig 6 which shows the status of the file blocks being corrupted along with date and time of corruption and the file is replaced with correct content after the replace operation.

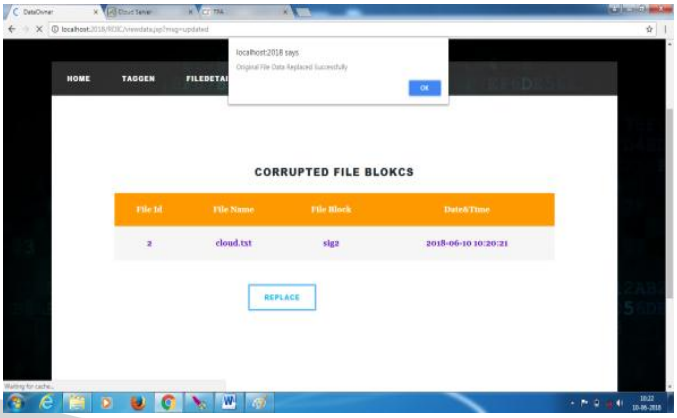

Figure 6: Auditing report

5.1 Advantages

- The communication cost between the server and the verifier is reduced significantly, making it more efficient and the setup and extract operations are invoked and completed fast within no time thus making it more effective.
- In our scheme the keys are generated by the key generation centre and the complex key management is eliminated, the users are provided with the secret key in a secure way and the key ones generated for the particular user can be used for creating tags for all their files.
- The verifier cannot learn or gain information from the stored data as we successfully achieve the zero-knowledge privacy.

VI. CONCLUSION

We have introduced a verification protocol based on identity to check the integrity of the stored data on cloud ,this protocol provides secure way of generating and distributing the secret key to the users, less communication cost and faster execution of set up and extract operations, here the system complexity and key management cost is reduced by making use of key homomorphic primitive of cryptography type, the data privacy is preserved by achieving zero-knowledge privacy through which no part of information is leaked to the verifier during the processing of operations and this proposed protocol is provably secure and efficient and provides reliable results. In future the protocol can be enhanced to support files dynamically in which the compatibility should be maintain between the data blocks and its respective tags.

REFERENCES

[1] P. Mell and T. Grance.(Jun. 3, 2009) Draft NIST Working Definition of Cloud Computing. Available

 DOI : 10.18231/2454-9150.2018.0555

http://csrc.nist.gov/ groups/SNC/cloudcomputing/ index.html.

[2] Bin Feng, Xinzhu Ma, Cheng Guo, Hui Shi, Zhangjie Fu, and Tie Qiu, "An Efficient Protocol With Bidirectional Verification for Storage Security in Cloud Computing",in special section on emerging trends, issues and challenges in energy-efficient cloud computing,pp.7899-7911,2016.

[3] Yannan Li, Yong Yu_, Geyong Min, Willy Susilo, Jianbing Ni and Kim-Kwang Raymond Choo " Fuzzy Identity-Based Data Integrity Auditing for Reliable Cloud Storage Systems", in Journal of Latex Class Files, vol. 14, pp.1-12, 2015.

[4] H. Wang, "Identity-based distributed provable data possession in multicloud storage", in IEEE Trans. Service Computing, vol. 8, pp. 328–340,2015.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–9.

[6] G. Ateniese, A. Faonio, and S. Kamara, "Leakage-resilient identification schemes from zero-knowledge proofs of storage," in Proc. IMA Int. Conf., 2015, pp. 311–328.

[7] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in Proc. CRYPTO, 2001, pp. 89–105.

[8] J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie- Hellman groups," in Proc. PKC, vol. 2567. 2003, pp. 18–30.

[9] F. Hess, "Efficient identity based signature schemes based on pairings," in Proc. Sel. Areas Cryptography, vol. 2595. 2003, pp. 310–324.

 DOI : 10.18231/2454-9150.2018.0555