iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** XII **Month of publication:** December 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Privacy in Data Storage and MultiOwner Authentication with Load Balancing in Cloud Computing

Thulasibai.U[1], Arul prem.G[2], Deepika.J[3]
Sri Venkateshwara College of Engineering and Technology, India

Abstract: ***Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the encumbrance of local data storage and maintenance. the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public audit ability for cloud storage is of critical prominence so that users can resort to a Third Party Auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. In this project, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.***
Keyword: Cloud Storage Security, batch auditing, Tpa, load balancing, Rdpc

I. INTRODUCTION

Cloud computing is recognized as an alternative to traditional information technology due to its intrinsic resource-sharing and low-maintenance characteristics. In cloud computing, the cloud service providers (CSPs), such as Amazon, are able to deliver various services to cloud users with the help of powerful datacenters. By migrating the local data management systems into cloud servers, users can enjoy high-quality services and save significant investments on their local infrastructures. One of the most fundamental services offered by cloud providers is data storage. Let us consider a practical data application. A company allows its staffs in the same group or department to store and share files in the cloud. By utilizing the cloud, the staffs can be completely released from the troublesome local data storage and maintenance. However, it also poses a significant risk to the confidentiality of those stored files. Specifically, the cloud servers managed by cloud providers are not fully trusted by users while the data files stored in the cloud may be sensitive and confidential, such as business plans. To preserve data privacy, a basic solution is to encrypt data files, and then upload the encrypted data into the cloud . Unfortunately, designing an efficient and secure data sharing scheme for groups in the cloud is not an easy task due to the following challenging issues.

Cloud computing takes the technology, services, and applications that are similar to those on the Internet and turns them into a self-service utility. The use of the word “cloud” makes reference to the two essential concepts:

Abstraction: Cloud computing abstracts the details of system implementation from users and developers. Applications run on physical systems that aren't specified, data is stored in locations that are unknown, administration of systems is outsourced to others, and access by users is ubiquitous.

Virtualization: Cloud computing virtualizes systems by pooling and sharing resources. Systems and storage can be provisioned as needed from a centralized infrastructure, costs are assessed on a metered basis, multi-tenancy is enabled, and resources are scalable with agility.

Cloud storage enables users to remotely store their data. However, this new paradigm of storage service also makes the integrity protection of outsourced data a very challenging issue. Recently many integrity auditing protocols have been proposed, but most of

them focus on the single cloud environment or don't support batch auditing. In this paper, we propose a public batch data integrity auditing protocol for multi-cloud storage. In the protocol a third party auditor can simultaneously verify the multiple auditing requests from different users on distinct data files stored on different cloud storage servers. By utilizing homomorphic cipher text verification and recoverable coding approach, the proposed protocol not only provides privacy preserving public auditing for data integrity, but also achieves quick identification of corrupted data. With the batch auditing, the total auditing time can be reduced and the communication cost is also low. Extensive security and performance analysis show the proposed protocol is secure and efficient.

In cloud computing, load balancing distributes workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units or disk drives. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource. Using multiple components with load balancing instead of a single component may increase reliability through redundancy. Load balancing usually involves dedicated software or hardware, such as a multilayer switch or a Domain Name System server process.

Load balancing differs from channel bonding in that load balancing divides traffic between network interfaces on a network socket (OSI model layer 4) basis, while channel bonding implies a division of traffic between physical interfaces at a lower level, either per packet or on a data link

Remote data possession checking is a topic that focuses on how to frequently, efficiently and securely verify that a storage server can faithfully store its client's (potentially very large) original data without retrieving it. The storage server is assumed to be un-trusted in terms of both security and reliability. There are two types of schemes, namely provable data possession (PDP) and proof of retrievability (POR). The difference between PDP and POR is that POR checks the possession of data and it can recover data in case of a failure. Usually, a PDP can be transformed to a POR by addingera sure or error correcting codes.

Advances in networking and computing technologies have prompted many organizations to outsource their storage needs on demand. This new economic and computing paradigm is commonly referred to as cloud storage. It brings appealing benefits including relief of the burden for storage management, universal data access with independent geographical locations ,and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc.
However, there are barriers that hinder migration to the cloud. One of the main barriers is that ,due to lack of physical control over the outsourced data, a cloud user may worry about whetherher data are kept as expected. If the cloud user is a company, apart from the risk of remote malicious attacks on the cloud, the traditional concerns posed by malicious company insiders are now supplemented by the even more hazardous threat of malicious outsiders who are given
the power of insiders. A recent EU bill forces companies migrating to the cloud to be liable for any data corruption or privacy breach into which their cloud service provider (CSP) may incur ,even when they do not retain control over their data. Convincing cloud users that their data are intact is especially vital when users are companies. Remote data possession checking (RDPC) is a primitive designed to address this issue.
Instead of purchasing and maintaining their own computing infrastructure, scientists can now run data-intensive scientific applications in cloud computing environment by facilitating its vast storage and computation capabilities. During the scheduling of such scientific applications for execution, various computation data flows will happen between the controller and computing server instances. Amongst various quality-of-service (QoS) metrics, data security is one of the greatest concerns to scientists because their data may be intercepted or stolen by malicious parties during those data flows.
An existing typical method for addressing this issue is to apply Internet Key Exchange (IKE) scheme to generate and exchange session keys, and then to apply these keys for performing symmetric-key encryption which will encrypt those data flows. However, the IKE scheme suffers from low efficiency due to its low performance of asymmetric-key crypto logical operations over a large amount of data and high-density operations which are exactly the characteristics of scientific applications. In this paper, we propose Cloud Computing Background Key Exchange (CCBKE), a novel authenticated key exchange scheme that aims at efficient security-aware scheduling of scientific applications. Our scheme is designed based on randomness-reuse strategy and Internet Key Exchange (IKE) scheme. Theoretical analyses and simulation results demonstrate that, compared with the IKE scheme, our CCBKE scheme can significantly improve the efficiency by dramatically reducing time consumption and computation load without sacrificing the level of

security.

With the exception of the one-time pad, no cipher has been theoretically proven to be unbreakable. Furthermore, some recurring properties may be found in the ciphertexts generated by the first cipher. Since those ciphertexts are the plaintexts used by the second cipher, the second cipher may be rendered vulnerable to attacks based on known plaintext properties (see references below).

This is the case when the first layer is a program P that always adds the same string S of characters at the beginning (or end) of all ciphertexts (commonly known as a magic number). When found in a file, the string S allows an operating system to know that the program P has to be launched in order to decrypt the file. This string should be removed before adding a second layer.

To prevent this kind of attack, one can use the method provided by Bruce Schneier in the references below: generate a random pad of the same size of the plaintext, then XOR the plaintext with the pad, resulting in a first ciphertext. Encrypt the pad and the first ciphertext with a different cipher and a different key, resulting in 2 more ciphertexts. Concatenate the last 2 ciphertexts in order to build the final ciphertext. A cryptanalyst must break both ciphers to get any information. This will, however, have the drawback of making the ciphertext twice as long as the original plaintext.

Note, however, that a weak first cipher may merely make a second cipher that is vulnerable to a chosen plaintext attack also vulnerable to a known plaintext attack. However, a block cipher must not be vulnerable to a chosen plaintext attack to be considered secure. Therefore, the second cipher described above is not secure under that definition, either. Consequently, both ciphers still need to be broken. The attack illustrates why strong assumptions are made about secure block ciphers and ciphers that are even partially broken should never be used.

A. Our contribution

In this paper, we have the following main contributions:We propose a remote data integrity checking protocol for cloud storage, which can be viewed as an adaptation of Seb_e et al.'s protocol [1]. The proposed protocol inherits the support of data dynamics from [1], and supports public verifiability and privacy against third party verifiers, while at the same time it doesn't need to use a third-party auditor.

We give a security analysis of the proposed protocol, which shows that it is secure against the untrusted server and private against third party verifiers.

The **Merkle signature scheme** is a digital signature scheme based on hash trees (also called Merkle trees) and one-time signatures such as the Lamport signature scheme. and is an alternative to traditional digital signatures such as the Digital Signature Algorithm or RSA.

The advantage of the Merkle Signature Scheme is that it is believed to be resistant against quantum computer algorithms. The traditional public key algorithms, such as RSA and ELGamal would become insecure in case an effective quantum computer can be built (due to Shor's algorithm). The Merkle Signature Scheme however only depends on the existence of secure hash functions. This makes the Merkle Signature Scheme very adjustable and resistant against quantum computing

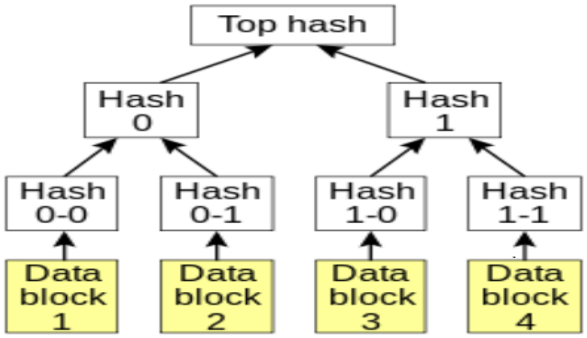


Figure1 merkel hash tree

Thus our contribution can be summarized as:

1. Pinpointing the source of the security flaw in Wang et al.'s scheme [1] and analyzing the inefficiency of [2], we propose secure privacy preserving public auditing protocol which is efficient. Our protocol provides the integrity assurance efficiently with strong evidence that unfaithful server cannot pass the verification process unless it indeed keeps the correct data intact.
2. We give a formal security proof (both for data storage correctness and privacy preserving assurance) of the propose scheme based on our security model, and prove that our scheme is secure against internal and external attacks.
3. We analyze the performance of our scheme mathematically and put comparisons with the state-of-the-art.Organization. First we review some related work and describe system model, design goals, security model and then we give privacy-preserving public auditing algorithm definitions and elaborate ourproposed scheme. batch auditing is described briefly and we give security and performance analysis

II. RELATED WORK

The remote data possession checking schemes can be categorized into two types, namely "provable data possession" (PDP) and "proof of retrievability" (POR). Usually, a PDP can be transformed to a POR by adding erasure or error correcting codes. PDP utilize RSA based holomorphic verifiable tags to achieve public auditability. Juels *et al*. propose a formal POR protocol definitionand accompanying security definitions [6]. Their scheme use spot-checking and error-correcting codes to ensure both "possession" and "retrievability" of data files on remote archive service systems.

However, the scheme can only be applied to encrypted files and can handle a prior. There has been a considerable amount of work done on untrusted outsourced storage. The shortest way to enforce the integrity control is to hire cryptographic hash function. Yumerefendi and Chase proposed a solution for authenticated network storage (Yumerefendi and Chase, 2007; Hsiao et al., 2009), using a hash tree (called as Merkle tree) as the underlying datastructure. However their processing of updates is computationally expensive. Fu et al. (2002) described and implemented a method for efficiently and securely accessing a read-only file system that has been distributed to many providers.

Dynamic data have also attracted attentions in the recent literature on efficiently providing the integrity guarantee of remotely stored data. Ateniese et al. is the first to propose a partially dynamic version of the prior PDP scheme, using only symmetric key cryptography but with a bounded number of audits. In [9], Wang et al. consider a similar support forpartially dynamic data storage in a distributed scenario with additional feature of data error localization.

In a subsequent work, Wang et al. [8] propose to combine BLS-based HLA with MHT to support fully data dynamics. Concurently, Erwayet al. develop a skip list based scheme to also enable provable data possession with full dynamics support. However, the verification in both protocols requires the linear combination of sampled blocks as an input, like the designs and thus doesnot support privacy-preserving auditing.

In other related work, Sebe et al. thoroughly study a set of requirements which ought to be satisfied for a remote data possession checking protocol to be of practical use. Their proposed protocol supports unlimited times of file integrity verifications and allows preset tradeoff between the protocol running time and the local storage burden at the user.

Schwarz and Miller propose the first study of checking the integrity of the remotely stored data across multiple distributed servers. Their approach is based on erasure-correcting code and efficient algebraic signatures.

Cloud is made up of massive resources. Management of these resources requires efficient planning and proper layout. While designing an algorithm for resource provisioning on cloud the developer must take into consideration different cloud scenarios and must be aware of the issues that are to be resolved by the proposed algorithm. Therefore, resource provisioning algorithm can be categorized into different classes based upon the environment, purpose and technique of proposed solution

III. PROBLEM STATEMENT

A. The system and thread model

We illuminate descriptive cloud data storage network architecture in Figure2. Three different entities can be identified as follows:

cloud user, who has large sum of data files to be stored in the cloud; *cloud storage server* (CSS) which is managed by the *cloud service provider* (CSP), provides data storage service and has significant storage space and computation resources; *third party auditor* (TPA), who has knowledge and skills that cloud users do not have and is trusted to assess the reliability of cloud storag service on behalf of the user upon request.

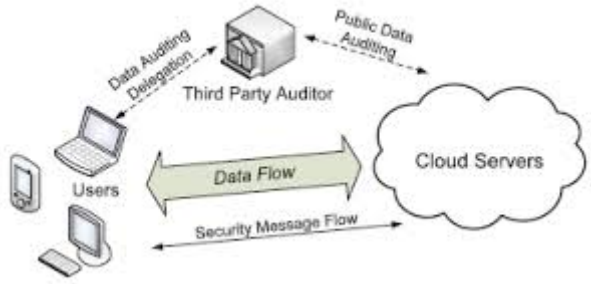


Figure 2 Architecture of cloud data storage

B. Design goals

The following security and performance guarantees should be achieved in our designed protocol:

1) Privacy-preserving: to ensure that the TPC cannot derive users' data content from the collected checking information.

2) Lightweight: to allow TPC to perform checking with minimum communication and computation cost.

3) Enable dynamic data operation: to fully support block level operations on the data file, which includes insertion, modification and deletion.

IV. PROPOSED SCHEME

In this scheme we plan to remove the security thread for the cloud architecture using the encrypted technique: Diffie Hellman and Elliptical Curve Cryptography .hence the proposed architecture will provide the better reliability and security also it will provide the data integrity at the user point of view. Our scheme involves the following step:

A. Creation of drop box
B. Establish connection
C. Account creation
D. Authentication by user
E. Data partitioning using MHT
F. Third party auditing
G. Batch auditing scheme
H. Load balancing

A. Creation of drop box

The first and for most step is to create a drop in the cloud. which act as the storage for the data. and it is owned by the data owner.is explained in the figure3.

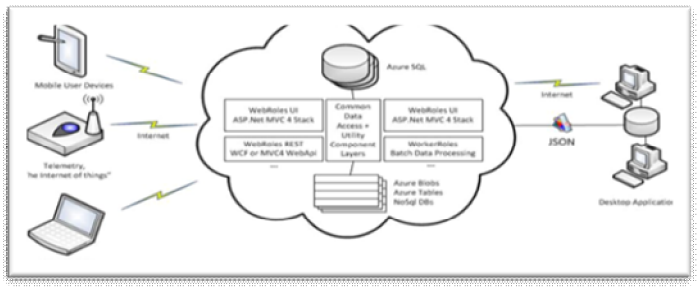


Figure3.creation of dropbox

B. Establish the connection

As soon as the creation of the drop box next step is to establish a connection with the help of the RDPC protocol
It is show in the figure4.

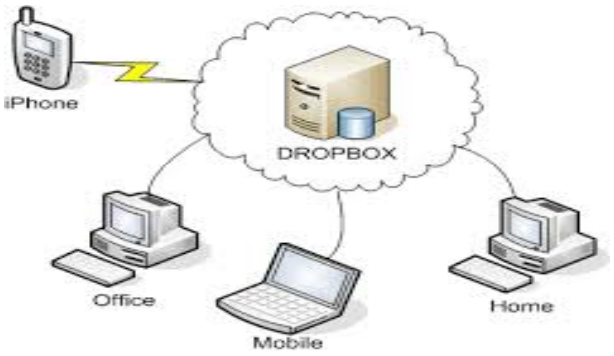


Figure4.Establish connection with drop box

C. Account creation in the cloud server

When the secured connection is established between the server and the client, data owner creates the account in the main cloud server. The user is asked to fill the account details and these details are sending over the internet. The account is now created in a system and it is put over in a cloud drop box further the connection is established by the RDPC protocol.to start the process.

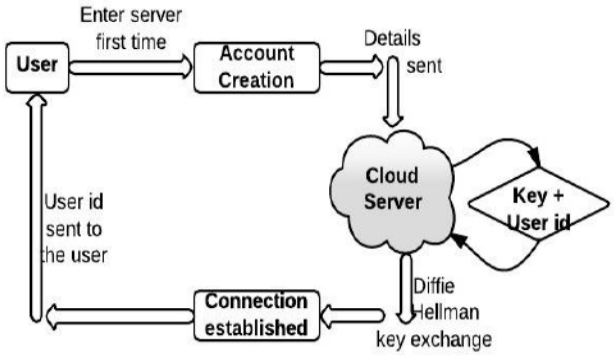


Figure5.account creation in the cloud server

D. Authentication by the user

Once the account is created the user need to be authenticated, and the secrete userid is send back to them once the authentication process gets completed. The end of the authentication process private key and algorithm is encrypted and sen back to the user.

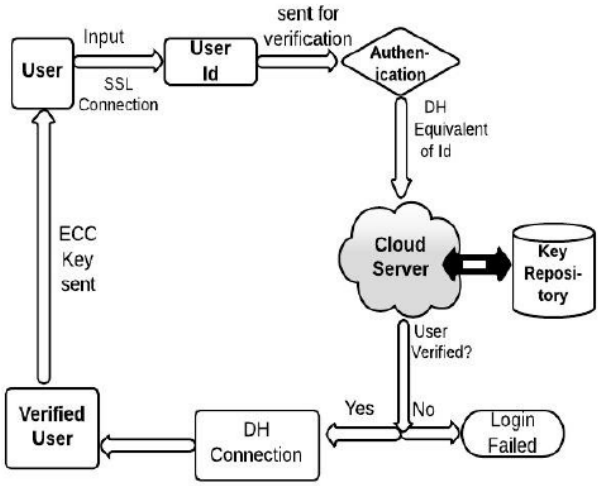


Figure6 authentication by user

E. Data partitioning using merkel hash tree algorithm

Once the authentication process is completed data is uploaded in the main cloud server it is encrypted ,and spilt in the equal parts to create the top hash.hasing is performed with the help of the MHT.

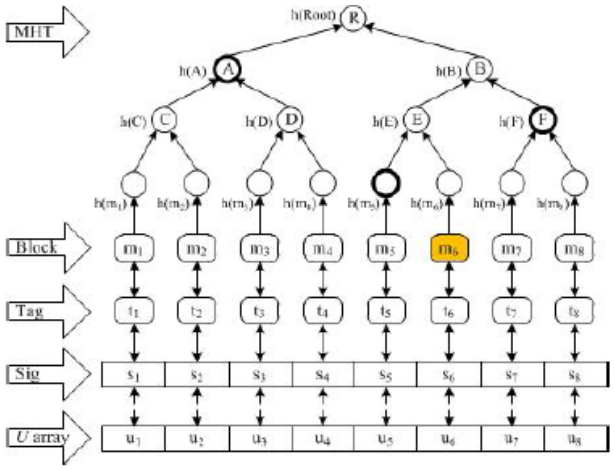


Figure7 data partition using MHT

F. Third party auditing

The Third Party Auditor (TPA) maintains the signature for shared data files. TPA performs the public data verification for data providers. Data integrity verification is performed is using Merkel Hashing tree algorithm (MHT).Homomorphic linear authenticator and random masking techniques are used for privacy preservation process. Using this technique tpa cannot able to misuse the data.as it is masked with the random masking technique.

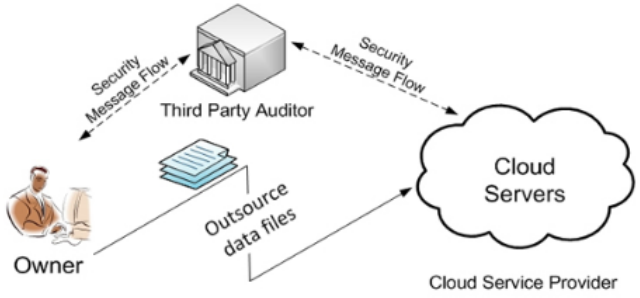


Figure8 third party auditing

G. Batching auditing scheme

Data integrity verification is carried out under auditing process. Batch auditing is applied to perform simultaneous data verification process. Batch auditing is tuned for multi user environment, Data dynamism is integrated with batch. The privacy preserving public auditing scheme is enhanced to perform data verification for multi user environment. Batch verification scheme is adapted to multi user data sharing environment. Data dynamism is integrated with public data auditability scheme. The system is improved to support public auditing based data sharing under commercial cloud environment. The cloud data sharing scheme is designed to manage data sharing based on economic model. Batch auditing mechanism is adapted for the data verification process. Dynamic data updates are managed with auditing process. The system is divided into five major modules. They are data center, Third Party Auditor, client, data dynamism handler and batch auditing. The cloud data center manages the shared data values. Auditing operations are initiated by the Third Party Auditor. Client application is designed to manage data upload and download operations. Data update operations are managed under data dynamism module. Batch auditing is designed for multi user data verification process

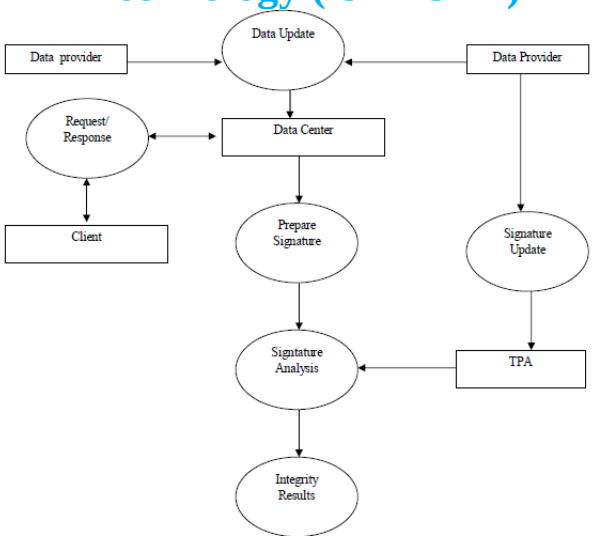


Figure9 batch auditing using TPA

H. Load balancing

Once the User send the request to process the job, the Cloud Service Provider will pass the request to the any of the sub cloud server in the Cloud Server Provider. Each three data owner has one database based request they retrieve the data via cloud server. So that the User requested Job will be assigned to the available sub cloud server via cloud service provider which contains minimum load and the concerned sub cloud server will process the User requested Job.

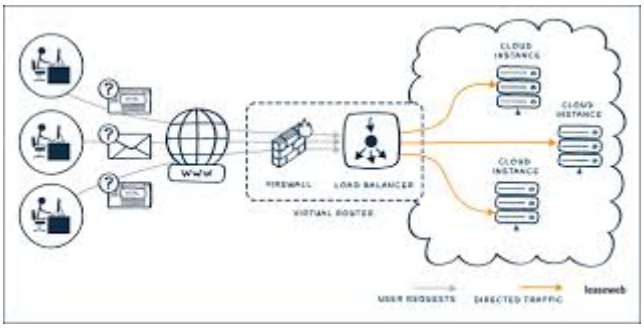


Figure10 load balancing in user requested job

V. PROTOCOL DETAILS

Checking data possession in networked information systems such as those related to critical infrastructures is a matter of crucial importance. Remote data possession checking protocols permit to check that a remote server can access an uncorrupted file in such a way that the verifier does not need to know beforehand the entire file that is being verified. Unfortunately, current protocols only allow a limited number of successive verifications or are impractical from the computational point of view. In this paper, we present a new remote data possession checking protocol such that: i) it allows an unlimited number of file integrity verifications; ii) its maximum running time can be chosen at set-up time and traded off against storage at the verifier.

A remote data possession checking (RDPC) protocol using algebraic signatures. It achieves many desirable features such as high efficiency, small challenges and responses, non-block verification. In this paper, we find that the protocol is vulnerable to replay attack and deletion attack launched by a dishonest server. Specifically, the server can either fool the user to believe that the data is well maintained but actually only a proof of the challenge is stored, or can generate a valid response in the integrity checking process after deleting the entire file of the user. We then propose an improved scheme to fix the security flaws of the original protocol without losing the desirable features of the original protocol.

A. Proposed RDPC Protocol

In this section, we present an efficient RDPC protocol. The proposal is built based on bilinear pairings, which are briefly reviewed below.

Bilinear pairings

Let $G1$ and $G2$ be two cyclic multiplicative groups with the same prime order q, *i.e.*, $|G1| = |G2| = q$.

Let $e : G1 \times G1 \rightarrow G2$ be a bilinear map [22], which satisfies the following properties:

1) Bilinearity: $\forall g1, g2, g3 \in G1$ and $a, b \in Zq$,

$e(g1, g2g3) = e(g2g3, g1) = e(g2, g1)e(g3, g1)$

$e(g1a, g2b) = e(g1, g2)ab$

2) Non-degeneracy: $\exists g4, g5 \in G1$ such that $e(g4, g5) \neq 1G2$.

3) Computability: $\forall g6, g7 \in G1$, there is an efficient algorithm to calculate $e(g6, g7)$.

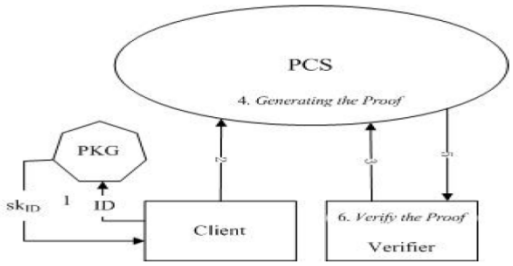


Figure11 architecture of RDPC protocol

B. Components of a RDPC protocol

These protocols consist of the following algorithm.

1) KeyGen,
2) TagGen,
3) Challenge,
4) ProofGen,
5) ProofVerify

keyGen is probably run by the user these take the security parameter k and return the secrete key as the output. TagGen algorithm is run by the cloud user to generate the tag for their files.it takes the input of the secrete key k and data block d and return the output of the block tag Challenge algorithm is run by the cloud user to generate the challenge response message. These message the main success scenario the protocol. These message is used query the integrity of the data. proofGen is a deterministic algorithm which is run by the cloud server it takes the input of the block tag T and challenge message to generate the proof of the possession of the data. Proof verification is also runs by the cloud user which takes the input of the secrete key and also proof of the data block. This will determine storage corrects of the data.

C. Forgery attack on the protocol

The forgery attack, a serious security threat to RDPC protocols, indicates that the server can forge a valid tag on any block. When responding a challenge from a cloud user. With this attack, the server is able to modify users' data on its own purpose without being detected, or discard users' data to save storage space or hide data loss accidents to maintain a good reputation. In the forgery attack, if a challenged block is corrupted, the server can use the modified block or chosen data arbitrarily to compute the corresponding tag, and then utilize them to generate a valid proof to cheat the user that all the data stored in the cloud are well maintained. Therefore, provided that the MHT is well maintained, the server can always respond a proof that can make the verification holds. Specifically, suppose the block bt in file F which is being challenged has been modified to t or deleted, but the MHT of F is well preserved, the server can conduct the procedures shown in fig 12 to cheat the user in the execution of the RDPC protocol.

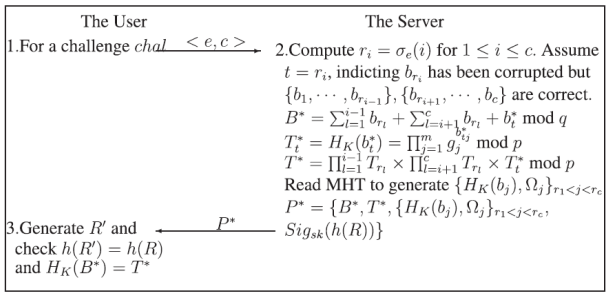


Figure12 forgery attack on RDPC protocol

D.working of RDPC protocol

The working of the protocol is explained with the help of the table.

TABLE 1 Interaction between TPA and CSS

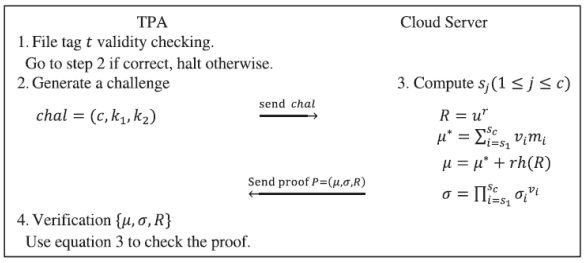
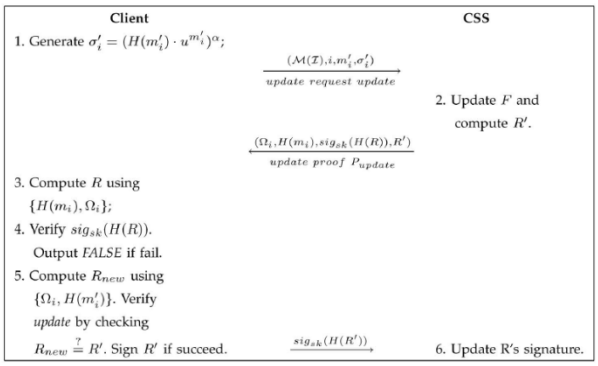


TABLE2 Interaction between client and sever



VI. ALGORITHM DETAILS

In this paper we use three different types of the algorithm for encryption purpose we are using elliptical curve algorithm and Diffie-hellman key exchange algorithm for hashing purpose merkel hash tree is used.

A. Diffie –hellman key exchange

Diffie-Hellman key exchange, also called exponential key exchange, is a method of digital encryption that uses numbers raised to specific powers to produce decryption keys on the basis of components that are never directly transmitted, making the task of a would-be code breaker mathematically overwhelming.

To implement Diffie-Hellman, the two end users Alice and Bob, while communicating over a channel they know to be private, mutually agree on positive whole numbers *p* and *q*, such that *p* is a prime number and *q* is a generator of *p*. The generator *q* is a number that, when raised to positive whole-number powers less than *p*, never produces the same result for any two such whole numbers. The

value of *p* may be large but the value of *q* is usually small.

Once Alice and Bob have agreed on *p* and *q* in private, they choose positive whole-number personal keys *a* and *b*, both less than the prime-number modulus *p*. Neither user divulges their personal key to anyone; ideally they memorize these numbers and do not write them down or store them anywhere. Next, Alice and Bob compute public keys *a** and *b** based on their personal keys according to the formulas

$$a^* = q^a \bmod p$$

and

$$b^* = q^b \bmod p$$

The two users can share their public keys *a** and *b** over a communications medium assumed to be insecure, such as the internet or a corporate wide area network (WAN). From these public keys, a number *x* can be generated by either user on the basis of their own personal keys. Alice computes *x* using the formula. $x = (b^*)^a \bmod p$

Bob computes *x* using the formula

$x = (a^*)^b \bmod p$

The value of *x* turns out to be the same according to either of the above two formulas. However, the personal keys *a* and *b*, which are critical in the calculation of *x*, have not been transmitted over a public medium. Because it is a large and apparently random number, a potential hacker has almost no chance of correctly guessing *x*, even with the help of a powerful computer to conduct millions of trials. The two users can therefore, in theory, communicate privately over a public medium with an encryption method of their choice using the decryption key *x*.

The most serious limitation of Diffie-Hellman in its basic or "pure" form is the lack of authentication. Communications using Diffie-Hellman all by itself are vulnerable to man in the middle attacks. Ideally, Diffie-Hellman should be used in conjunction with a recognized authentication method such as digital signatures to verify the identities of the users over the public communications medium. Diffie-Hellman is well suited for use in data communication but is less often used for data stored or archived over long periods of time. this can be explained using figure.

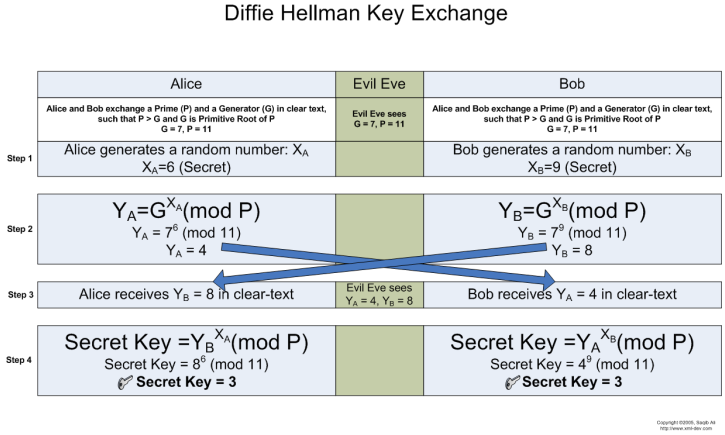


Figure Diffie hellman key exchange

B. Elliptic curve cryptography (ECC)

ECC is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. Elliptic curves are also used in several integer factorization algorithms that have applications in cryptography, such as Lenstra elliptic curve

factorization**.**

Elliptic Curve Cryptography (ECC) was discovered in 1985 by Victor Miller (IBM) and Neil Koblitz (University of Washington) as an alternative mechanism for implementing public-key cryptography. Public-key algorithms create a mechanism for sharing keys among large numbers of participants or entities in a complex information system. Unlike other popular algorithms such as RSA, ECC is based on discrete logarithms that are much more difficult to challenge at equivalent key lengths.

The equation of an elliptic curve is given as,

$$y^2 = x^3 + ax + b.$$

Few terms that will be used,

E -> Elliptic Curve

P -> Point on the curve

n -> Maximum limit (This should be a prime number)

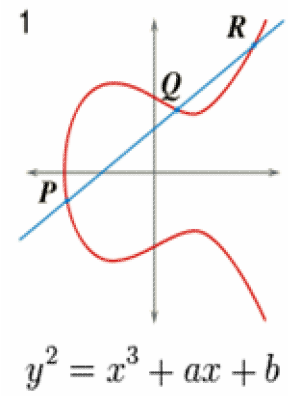


Figure Elliptical curve algorithm

C. Merkel Hash Tree Algorithm

Merkle trees are binary trees of hashes. Merkle trees in use a **double** SHA-256, the SHA-256 hash of the SHA-256 hash of something. Merkle Tree is a tree where the leaf nodes contain the hashes of some data blocks, and the internal nodes contain hashes of their children. It provides a quick way to verify data.

For example, in a peer-to-peer network, a peer can use a Merkle Tree or parts of it (explain later) to quickly verify the data it receives from other peers have not been tampered with, or the data are not corrupted during the transmission. Borrowing from wikipedia, below is a picture of what a Merkle Tree looks like.

1) Advantages

a) By providing the Public and Private key components the user is only allowed to access the data.

b) By allowing the Trusted party Auditor to audit the data will increase the By using Merkle Hash Tree Algorithm the data will be audited via multiple level of batch auditing Process.

c) As Business Point of view, the Company's Customers will be increased due to the Security and Auditing Process.

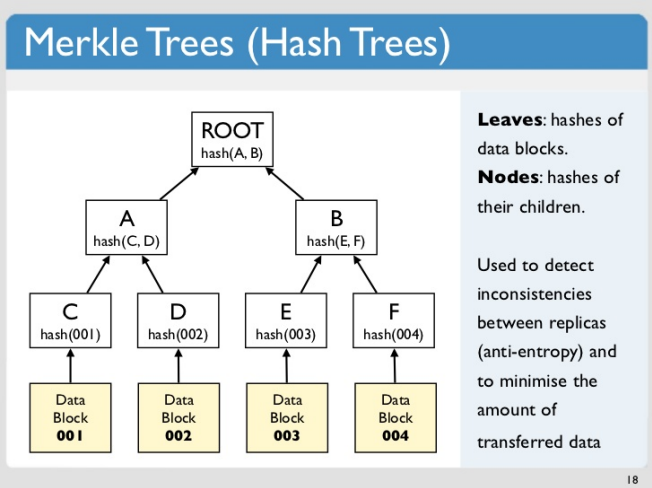


Figure Merkel hash tree algorithm

VII. CONCLUSION AND ACKNOWLEDGEMENT

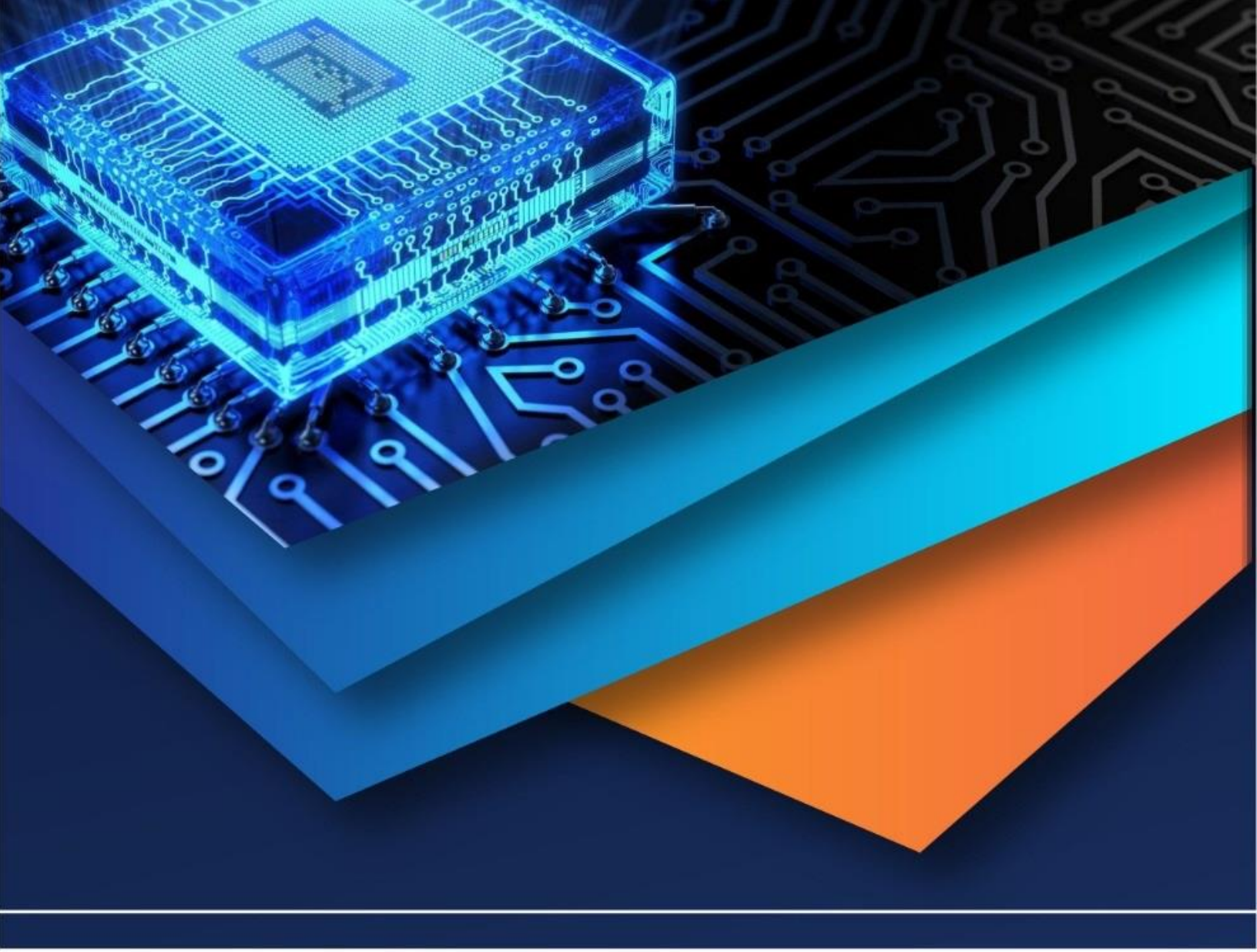
In this paper we discover the new ideas of how the data is stored securely. Data integrity is checked efficiently with the help of the TPA. The main theme of this paper is that to create a top hash using MHT and introduce the TPA so that it should not bring no new vulnerability. The load balancing scheme help us to perform the user requested job so the cloud server is more faster and can perform the job in the fraction of seconds. Hence the paper will provide the provable security and highly efficient.

VIII. ACKNOWLEDGEMENT

I wish to thank, first and foremost, my Professor MR.G.ARUL PREM and my parents…

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A View of Cloud Computing,” Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.
[2] S. Kamara and K. Lauter, “Cryptographic Cloud Storage,” Proc. Int’l Conf. Financial Cryptography and Data Security (FC), pp. 136- 149, Jan. 2010.
[3] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing,” Proc. IEEE INFOCOM, pp. 534-542, 2010.
[4] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, “Plutus: Scalable Secure File Sharing on Untrusted Storage,” Proc. USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.
[5] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, “Sirius: Securing Remote Untrusted Storage,” Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 131-145, 2003.
[6] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage,” Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 29-43, 2005.
[7] R. Lu, X. Lin, X. Liang, and X. Shen, “Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing,” Proc. ACM Symp. Information, Computer and Comm.Security, pp. 282-292, 2010.
[8] B. Waters, “Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization,” Proc. Int’l Conf. Practice and Theory in Public Key Cryptography Conf. Public Key Cryptography, http://eprint.iacr.org/2008/290.pdf, 2008.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY