International Journal of Computer Science Engineering & Technology
APPROVED BY
NATIONAL SCIENCE LIBRARY (NSL)
NATIONAL INSTITUTE OF SCIENCE COMMUNICATION AND INFORMATION RESOURCES (NISCAIR)
COUNCIL OF SCIENTIFIC AND INDUSTRIAL RESEARCH (CSIR)– NEW DELHI INDIA .
ISSN :2455-9091

PUBLIC INTEGRITY AUDITING FOR SHARED DYNAMIC CLOUD DATA WITH GROUP USER REVOCATION

[1]R.Bharathi, [2]R.Jamuna Devi
[1]Assistant Professor, [2]Research Scholar
[1,2]Department of Computer Science,
[1,2]Sengunthar College of Arts and Science,
[1,2]Tiruchengode, Tamilnadu.

ABSTRACT -The advent of the cloud computing makes storage outsourcing becomes a rising trend, which promotes the secure remote data auditing a hot topic that appeared in the research literature. Recently some research considers the problem of secure and efficient public data integrity auditing for shared dynamic data. However, these schemes are still not secure against the collusion of cloud storage server and revoked group users during user revocation in practical cloud storage system. In this paper, we figure out the collusion attack in the exiting scheme and provide an efficient public integrity auditing scheme with secure group user revocation based on vector commitment and verifier-local revocation group signature. We design a concrete scheme based on the scheme definition. Our scheme supports the public checking and efficient user revocation and also some nice properties, such as confidently, efficiency, count ability and traceability of secure group user revocation. Finally, the security and experimental analysis show that compared with its relevant schemes our scheme is also secure and efficient.

Key Words: [Cloud Service Providers (CSPs), Asymmetric Group Key Agreement (AGKA), Computational Diffie Hellman (CDH), Message Authentication Code (MAC), Third Part Auditor (TPA), Vector Commitment (VC).]

1. INTRODUCTION

The development of cloud computing motivates enterprises and organizations to outsource their data to third-party cloud service providers (CSPs), which will improve the storage limitation of resource constrain local devices. Recently, some commercial cloud storage services, such as the simple storage service on-line data backup services of Amazon and some practical cloud based software Google Drive, Dropbox, Mozy, Bitcasa , and Memopal, have been built for cloud application. Since the cloud servers may return an invalid result in some cases, such as server hardware/software failure, humanmaintenance and malicious attack, new formsof assurance of data integrity and accessibility arerequired to protect the security and privacy of clouduser's data. To overcome the above critical security challengeof today's cloud storage services, simple replicationand protocols like Rabin's data dispersion scheme are far from practical application. The formers are not practical because a recent IDC report suggeststhat data-generation is outpacing storage availability. The later protocols ensure the availability of datawhen a quorum of repositories, such ask-out-of-n ofshared data, is given. However, they do not provideassurances about the availability of each repositories,which will limit the

assurance that the protocols canprovide to relying parties.

In these solutions, when a scheme supports data modification, we call it *dynamic* scheme, otherwise *static* one (or limited dynamic scheme, if a scheme could only efficiently support some specified operation, such as append). A scheme is *publicly verifiable* means that the data integrity check can be performed not only by data owners, but also by any third-party auditor. However, the dynamic schemes above focus on the cases where there is a data owner and only the data owner could modify the data.

Recently, the development of cloud computing boosted some applications where the cloud service is used as a collaboration platform. In these software development environments, multiple users in a group need to share the source code, and they need to access, modify, compile and run the shared source code at any time and place. The new cooperation network model in cloud makes the remote data auditing schemes become infeasible, where only the data owner can update its data. Obviously, trivially extending a scheme with an online data owner to update the data for a group is inappropriate for the data owner. It will cause tremendous communication and computation overhead to dataowner, which will result in the single point of dataowner. However, the scheme assumed that theprivate and authenticated channels exist between each pare of entities and there is no collusion among them.Also, the auditing cost of the scheme is linear to the group size. Another attempt to improve the previous scheme and make the scheme efficient, scalable and collusion resistant is Yuan and Yu [24], who designed a dynamic public integrity auditing scheme with group user revocation. The proxy tag update techniques in their scheme, which make their scheme support public checking and efficient user revocation. However, in their scheme, the authors do not consider the data secrecy of group users. It means that, their scheme could efficiently support plaintext data update and integrity auditing, while not clip hypertext data. In their scheme, if the data owner trivially shares a group key among the group users. Also, the data owner does not take part in the user evocation phase, where the cloud itself could conduct the user revocation phase. In this case, the collusion of revoked user and the cloud server will give chance to malicious cloud server where the cloud server could update the data as many time as designed and provide a legal data finally. To the best of our knowledge, there is still no solution for the above problem in public integrity auditing with group user modification.

The deficiency of above schemes motivates us to explore how to design an efficient and reliable scheme, while achieving secure group user revocation. To the end, we propose a construction which not only supports group data encryption and decryption during the data modification processing, but also realizes efficient and secure user revocation. Our idea is to apply vector commitment scheme over the database. Then we leverage the Asymmetric Group Key Agreement (AGKA) and group signatures to support ciphertext data base update among group users and efficient group user revocation respectively. Specifically, the group user use the AGKA protocol to encrypt/decrypt the share database, The group signature will prevent the collusion of cloud and revoked group users, where the data owner will take part in the user revocation phase and the cloud could not revoke the data that last modified by the revoked user.

2. RELATED WORK

J. Yuan and S. Yu,[1] presented efficient public integrity checking for cloud data sharing with multi-user modification in which is featured by salient properties of public integrity checking and continual computational cost on user side. We achieve

this through our novel design on polynomial based authentication tags which allows accumulation of tags of different data blocks.

X. Chen, J. Li, J. Weng, J. Ma, and W. Lou,[2] proposed verifiable computation over large database with incremental updates. Authors formalize the notion of verifiable database with incremental updates (Inc-VDB). Besides, they propose a general Inc-VDB framework by incorporating the primitive of vector commitment and the encrypt-thenincremental MAC mode of encryption. They present a concrete Inc-VDB scheme based on the computational DiffieHellman (CDH) assumption and prove that system can achieve the desired security properties.

E. Shi, E. Stefanov, and C. Papamanthou,[4] proposed practical dynamic proofs of retrievabilitywith constant client storage whose bandwidth cost is comparable to a Merkle hash tree, thus being very practical. There construction outperforms the constructions of Stefanov et al. and Cash et al., both in theory and in practice. Specifically, for n outsourced blocks of β bits each, writing a block requires β +O(λ log n) bandwidth and O(β log n) server computation (λ is the security parameter). Audits are also very efficient, requiring β + O(λ 2 log n) bandwidth. They also show how to make there scheme publicly verifiable, providing the first dynamic PoR scheme with such a property. They finally provide a very efficient implementation of our scheme.

B. Wang, L. Baochun, and L. Hui,[5] presented public auditing for shared data with efficient user revocation in the cloud. By utilizing the idea of proxy re-signatures, they allow the cloud to re-sign blocks on behalf of existing users during user revocation, so that existing users do not need to download and re-sign blocks by themselves. In addition, a public verifier is always able to audit the integrity of shared data without retrieving the entire data from the cloud, even if some part of shared data has been re-signed by the cloud.

C. Wang, Q. Wang, K. Ren, and W. Lou,[6] presented privacy-preserving public auditing for data storage security in cloud computing utilize the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files.

SYSTEM MODEL
PROBLEM DESCRIPTION
EXISTING SYSTEM:

For providing the integrity and availability of remote cloud store, some solutions and their variants have been proposed. In these solutions, when a scheme supports data modification, we call it *dynamic* scheme, otherwise *static* one (or limited dynamic scheme, if a scheme could only efficiently support some specified operation, such as append). A scheme is *publicly verifiable* means that the data integrity check can be performed not only by data owners, but also by any third-party auditor. However, the dynamic schemes above focus on the cases where there is a data owner and only the data owner could modify the data. To support multiple user data operation, Wang et al. proposed a data integrity based on ring signature.Tofurther enhance the previous scheme and support group user revocation, Wang et al. designed a scheme based on proxy re-signatures.Another attempt to improve the previous scheme and make the scheme efficient, scalable and collusion resistant is Yuan and Yu, who designed a dynamic public integrity auditing scheme with group

user revocation. The authors designed polynomial authentication tags and adopt proxy tag update techniques in their scheme, which make their scheme support public checking and efficient user revocation.

DISADVANTAGES OF EXISTING SYSTEM:

In the Wang et al. scheme, the user revocation problem is not considered and the auditing cost is linear to the group size and data size.However, the scheme assumed that the private and authenticated channels exist between each pair of entities and there is no collusion among them. Also, the auditing cost of the scheme is linear to the group size.However, in Yuan and Yu scheme, the authors do not consider the data secrecy of group users. It means that, their scheme could efficiently support plaintext data update and integrity auditing, while not ciphertext data. In their scheme, if the data owner trivially shares a group key among the group users, the defection or revocation any group user will force the group users to update their shared key. Also, the data owner does not take part in the user revocation phase, where the cloud itself could conduct the user revocation phase. In this case, the collusion of revoked user and the cloud server will give chance to malicious cloud server where the cloud server could update the data as many time as designed and provide a legal data finally.

3. PROPOSED SYSTEM:

The deficiency of above schemes motivates us to explore how to design an efficient and reliable scheme, while achieving secure group user revocation. To the end, we propose a construction which not only supports group data encryption and decryption during the data modification processing, but also realizes efficient and secure user revocation. Our idea is to apply vector commitment scheme over the database. Then we leverage the Asymmetric Group Key Agreement (AGKA) and group signatures to support ciphertext data base update among group users and efficient group user revocation respectively. Specifically, the group user uses the AGKA protocol to encrypt/decrypt the share database, which will guarantee that a user in the group will be able to encrypt/decrypt a message from any other group users. The group signature will prevent the collusion of cloud and revoked group users, where the data owner will take part in the user revocation phase and the cloud could not revoke the data that last modified by the revoked user.

ADVANTAGES OF PROPOSED SYSTEM:

In this paper explore on the secure and efficient shared data integrate auditing for multi-user operation for ciphertext database. By incorporating the primitives of victor commitment, asymmetric group key agreement and group signature, we propose an efficient data auditing scheme while at the same time providing some new features, such as traceability and countability. Provide the security and efficiency analysis of our scheme, and the analysis results show that our scheme is secure and efficient.

MODULES SYSTEM MODEL AND SECURITY MODEL

4.1 OVERVIEW PRELIMINARIES

Our scheme makes use of bilinear groups. The security of the scheme depends on the Strong Diffie- Hellman assumption and the Decision Linear assumption. In this section, we review the definitions of bilinear groups and the complexity assumption.

BILINEAR GROUPS

We review a few concepts related to bilinear maps, which follow the notation of . Let G1 and G2 be
two multiplicative cyclic groups of prime order p, g1 is a generator of G1 and g2 is a generator of G2. ψis
an efficiently computable isomorphism from G2 to G1 with $\psi(g2) = g1$, and $e : G1 \times G2$

→ GT is a bilinear map with the following properties:

Computability : there exits an efficiently computable algorithm for computing map e;

Bilinearity : for all u ∈G1, v ∈G2 and a, b ∈Zp, e(ua, vb) = e(u, v)ab;

Non-degeneracy: e(g1, g2) 6= 1.

COMPLEXITY ASSUMPTION

The security of our scheme relies on the difficulty of some problems: the Strong Diffie-Hellman problem, the Decision Linear problem, and the Computational Diffie-Hellman problem. We describe these problems as follows.

4.1.1 Definition1

q-StrongDiffie-Hellman problem. Let G1, G2 be cyclic group of prime order p, where possibly G1 = G2. Let g1 be a generator of G1 and g2 be a generator of G2. Given a (q + 2) −

tuple(g1, g2, g 2 , g(2) 2 , ..., g(q) 2) as input, output a pair (g1/(+x) 1 , x) where x ∈Z∗p.

1.The assumption could be used to construct short signature scheme without random oracles .

2.The assumption has properties similar to the Strong-RSA assumption and the properties are adopted for building short group signature in our scheme.

4.1.2 Definition 2

Decision Linear problem. Let g1 be a generator of G1, and G1 be a cyclic group of prime

orderp. Given u, v, h, ua, ub,uc∈G1 as input, output yes if a + b = c and no otherwise.

Boneh et al. introduced the Decision Linear assumption and they proved that the problem is

intractable in generic bilinear groups.

4.1.3 Definition 3

Square Computational Diffie-Hellman (Square-CDH) problem. With g ∈G1 as above, given

(g, gx) for x ∈R Zp as input, output gx2

It has been proved that the Square-CDH assumption is equivalent to the classical CDH assumption.

VECTOR COMMITMENT

Commitment is a fundamental primitive in cryptography and it plays an important role in security protocols such as voting, identification, zero-knowledge proof, etc. The hiding property of commitment requires that it should not reveal information of the committed message, and the binding property requires that the committing mechanism should not allow a sender to change his/her mind about the committed message.

Recently, Catalano and Fiore put forward a new primitive called Vector Commitment. Vector Commitment satisfies position binding that an adversary should not be able to open a commitment to two different values at the same position, and the Vector Commitment is concise, which means that the size of the commitment string and its openings have to be independent of the vector length. We provide the formal definition of Vector Commitment as follows.

4.1.4 Definition 4

(Vector Commitment) A vector commitment scheme is a collection of six polynomial-time algorithms (VC.KeyGen, VC.Com, VC.Open, VC.Ver, VC.Update, VC.ProofUpdate) such that:

VC.KeyGen(1k, q). Given the security parameter k and the size q of the committed vector (with q =poly(k)), the key generation outputs some public parameters pp.

VC.Compp(m1, ...,mq). On input a sequence of q messages m1, ...,mq∈ M (M is the message space) and the public parameters pp, the committing algorithm outputs a commitment string C and an auxiliary information aux.

VC.Openpp(m, i, aux). This algorithm is run by the committer to produce a proof ithatmis the i-th committed message. In particular, notice that in the case when some

updates have occurred the auxiliary information aux can include the update information produced by these updates.

VC.Verpp(C,m, i,_i). The verification algorithm accepts (i.e., it outputs 1) only if _i is a valid proof

thatCwas created to a sequence m1, ...,mq such that m = mi.

VC.Updatepp(C,m,m′, i). This algorithm is run by the committer who produces C and wants to update it by changing the i-th message to m′. The algorithm takes as input the old message m, the new message m′ and the position i. It outputs a new commitment C' together with an update information U.

VC.ProofUpdatepp(C,_j,m′, i,U). This algorithm can be run by any user who holds a proof _j for some message at position j w.r.t. C, and it allows the user to compute an updated proof _′ j (and theupdated commitment C′) such that _′j will be validwith regard to C′ which contains m′ as the newmessage at position i. Basically, the value U containsthe update information which is needed to computesuch values.

The primitive of verifiable database with efficientupdate based on vector commitment is useful to solvethe problem of verifiable data outsourcing. Recently,Chen et al. [34], [35] figured out that the basic vectorcommitment scheme suffers from forward automaticupdate attack and backward substitution update attack.They also proposed a new framework for verifiabledatabase with efficient update from vectorcommitment, which is not only public verifiable fordynamic outsourced data but also secure against thetwo attacks. The solution in their scheme is easy toapply in our scheme, which will overcome the attacksthey figured out in our scheme.

Group Signature with User Revocation. Here present the formal definition of group signatures with verifier-local revocation as follows.

4.1.5 Definition 5

A verifier-local group signature scheme is a collection of three polynomial-time algorithms (VLR.KeyGen, VLR.Sign, VLR.Verify), which behaves as follows:

VLR.KeyGen(n). This randomized algorithm takes as input a parameter n, the number of Nembers of the group. It outputs a group public key gpk, an n-element vector of user keys gsk= (gsk[1], gsk[2], ..., gsk[n]), and an n-element vector of user revocation tokens grt, similarly indexed.

VLR.Sign(gpk, gsk[i],M). This randomized algorithm takes as input the group public key gpk, a private key gsk[i], and a message $M \in \{0, 1\}*$, and returns a signature σ.

VLR.Verify(gpk,RL, σ,M). The verification algorithm takes as input the group public key gpk, a set of revocation tokens RL(whose elements form a subset of the elements of grt), and a purported signature σ on a message M. It returns either valid or invalid. The latter response can mean either that σ is not a valid signature, or that the user who generated it has been revoked.

PERFORMANCE ANALYSIS ENCRYPTION:

In cryptography, encryption is the process of encoding a message or information in such a way that only authorized parties can access it. Encryption does not itself prevent interference, but denies the intelligible content to a would-be interceptor. In an encryption scheme, the intended information or message, referred to as plaintext, is encrypted using an encryption algorithm, generating ciphertext that can only be read if decrypted. For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. It is in principle possible to decrypt the message without possessing the key, but, for a well-designed encryption scheme, considerable computational resources and skills are required. An authorized recipient can easily decrypt the message with the key provided by the originator to recipients but not to unauthorized users.

Types:

- Symmetric key / Private key
- Public key

Symmetric Key / Private Key

In symmetric-key schemes, the encryption and decryption keys are the same. Communicating parties must have the same key before they can achieve secure communication.

Public Key

In public-key encryption schemes, the encryption key is published for anyone to use and encrypt messages. However, only the receiving party has access to the decryption key that enables messages to be read Public-key encryption was first described in a secret document in 1973 before then all encryption schemes were symmetric-key (also called private-key). A publicly available public key encryption application called Pretty Good Privacy (PGP) was written in 1991 by Phil Zimmermann, and distributed free of charge with source code; it was purchased by Symantec in 2010 and is regularly updated.

USES

Encryption has long been used by militaries and governments to facilitate secret communication. It is now commonly used in protecting information within many kinds of civilian systems. For example, the Computer Security Institute reported that in 2007, 71% of companies surveyed utilized encryption for some of their data in transit, and 53% utilized encryption for some of their data in storage. Encryption can be used to protect data "at rest", such as information stored on computers and storage devices (e.g. USB flash drives). In recent years, there have been numerous reports of confidential data, such as customers' personal records, being exposed through loss or theft of laptops or backup drives. Digital rights management systems, which prevent unauthorized use or reproduction of copyrighted material and protect software against reverse engineering (see also copy protection), is another somewhat different example of using encryption on data at rest.In response to encryption of data at rest, cyber-adversaries have developed new types of attacks. These more recent threats to encryption of data at rest include cryptographic attacks, stolen ciphertext attacks, attacks on encryption keys,insider attacks, data corruption or integrity attacks, data destruction attacks, and ransomware attacks. Data fragmentation and active defense data protection technologies attempt to counter some of these attacks, by distributing, moving, or mutating ciphertext so it is more difficult to identify, steal, corrupt, or destroy.

Encryption is also used to protect data in transit, for example data being transferred via networks (e.g. the Internet, e-commerce), mobile telephones, wireless microphones, wireless intercom systems, Bluetooth devices and bank automatic teller machines. There have been numerous reports of data in transit being intercepted in recent years. Data should also be encrypted when transmitted across networks in order to protect against eavesdropping of network traffic by unauthorized users.

MESSAGE VERIFICATION

Encryption, by itself, can protect the confidentiality of messages, but other techniques are still needed to protect the integrity and authenticity of a message; for example, verification of a message authentication code (MAC) or a digital signature. Standards for cryptographic software and hardware to perform encryption are widely available, but successfully using encryption to ensure security may be a challenging problem. A single error in system design or execution can allow successful attacks. Sometimes an adversary can obtain unencrypted information without directly undoing the encryption. See, e.g., traffic analysis, TEMPEST, or Trojan horse. Digital

signature and encryption must be applied to the ciphertext when it is created (typically on the same device used to compose the message) to avoid tampering; otherwise any node between the sender and the encryption agent could potentially tamper with it. Encrypting at the time of creation is only secure if the encryption device itself has not been tampered with.

DATA ERASURE:

Conventional methods for deleting data permanently from a storage device involve overwriting its whole content with zeros, ones or other patters – a process which can take a significant amount of time, depending on the capacity and the type of the medium. Cryptography offers a way of making the erasure almost instantaneous, as long as all data on a device is encrypted and the key is stored on the same device. Although this setup on its own does not offer any protection in case an unauthorised person gains physical access to the device, it means that all information on it can be made inaccessible by erasing only the key. An example implementation of this method can be found on iOS devices, where the cryptographic key is kept in a dedicated 'Effaceable Storage'.

DECRYPTION

Pirate decryption most often refers to the decryption, or decoding, of pay TV or pay radio signals without permission from the original broadcaster. The term "pirate" in this case is used in the sense of copyright infringement and has little or nothing to do with sea piracy, nor with pirate radio, which involved the operation of a small broadcast radio station without lawfully obtaining a license to transmit. The MPAA and other groups which lobby in favour of intellectual property (specifically copyright and trademark) regulations have labelled such decryption as "signal theft" even though there is no direct tangible loss on the part of the original broadcaster, arguing that losing out on a potential chance to profit from a consumer's subscription fees counts as a loss of actual profit.Decryption is the process of transforming data that has been rendered unreadable through encryption back to its unencrypted form. In decryption, the system extracts and converts the garbled data and transforms it to texts and images that are easily understandable not only by the reader but also by the system. Decryption may be accomplished manually or automatically. It may also be performed with a set of keys or passwords.

CRYPTOSYSTEM:

In cryptography, a cryptosystem is a suite of cryptographic algorithms needed to implement a particular security service, most commonly for achieving confidentiality (encryption). Typically, a cryptosystem consists of three algorithms: one for key generation, one for encryption, and one for decryption. The term *cipher* (sometimes *cypher*) is often used to refer to a pair of algorithms, one for encryption and one for decryption. Therefore, the term *cryptosystem* is most often used when the key generation algorithm is important. For this reason, the term *cryptosystem* is commonly used to refer to public key techniques; however both "cipher" and "cryptosystem" are used for symmetric key techniques.

Asymmetric Key Encryption

The encryption process where different keys are used for encrypting and decrypting the information is known as Asymmetric Key Encryption. Though the keys are different, they are mathematically related and hence, retrieving the plaintext by decrypting ciphertext is feasible.

CLOUD STORAGE MODEL

In the cloud storage model as shown in Figure 1, there are three entities, namely the cloud storage server, group users and a Third Part Auditor (TPA). Group users consist of a data owner and a number of users who are authorized to access and modify the data by the data owner. The

cloud storage server is semi-trusted, who provides data storage services for the group users. TPA could be any entity in the cloud, which will be able to conduct the data integrity of the shared data stored in the cloud server. In our system, the data owner could encrypt and upload its data to the remote cloud storage server. Also, he/she shares the privilege such as access and modify (compile and execute if necessary) to a number of group users. The TPA could efficiently verify the integrity of the data stored in the cloud storage server, even the data is frequently updated by the group users. The data owner is different from the other group users, he/she could securely revoke a group user when a group user is found malicious or the contract of the user is expired.

THREAT MODEL AND SECURITY GOALS

Our threat model considers two types of attack:

An attacker out side the group (include therevoked group user cloud storage server) may obtain some knowledge of the plaintext of the data. Actually, this kind of attacker has to at lease break the security of the adopted group data encryption scheme.

The cloud storage server colludes with the revoked group users, and they want to provide a illegal data without being detected.

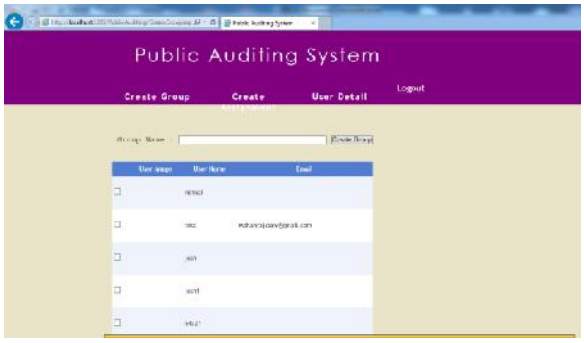
Actually, in cloud environment, we assume that the cloud storage server is semi-trusted. Thus, it is easonable that a revoked user will collude with the cloud server and share its secret group key to the cloud storage server. In this case, although the server proxy group user revocation way brings much communication and computation cost saving, it will make the scheme insecure against a malicious cloud storage server who can get the secret key of revoked users during the user revocation phase. Thus, a malicious cloud server will be able to make data m, last modified by a user that needed to be revoked, into a malicious data m′. In the user revocation process, the cloud could make the malicious data m′ become valid. To overcome the problems above, we aim to achieve the following security goals in our paper:

MODULES

Login

Register

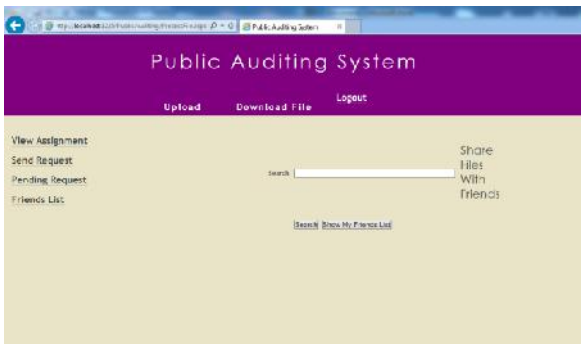
Admin Create Group



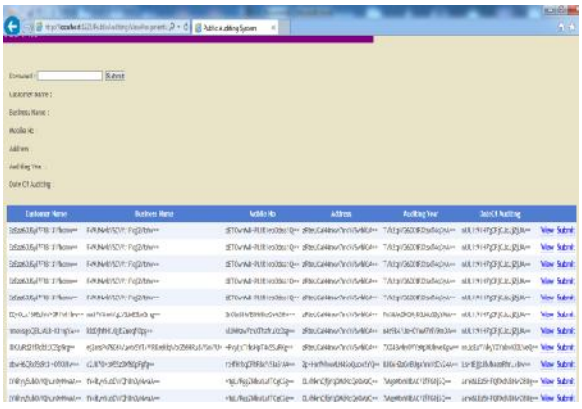
Create Assignment

Assignment

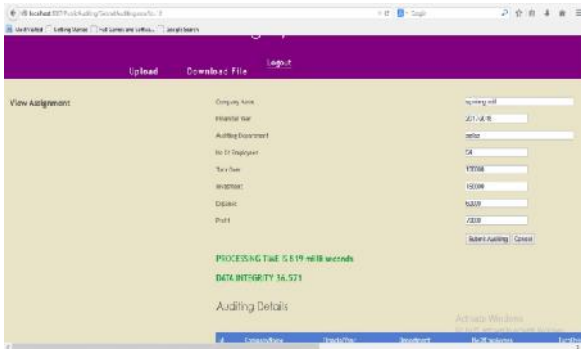
Download File



View and Submit Assignment



Result



5. CONCLUSION:

This section presents integrity auditing scheme which provides a complete outsourcing solution of data. After introducing notations considered and brief preliminaries, started from an overview of proposed data Integrity auditing scheme. Then, presenting main scheme and show how to extent the proposed scheme to support integrity auditing for the TPA upon delegations from multiple users. Finally, the proposed how to generalize integrity auditing keeping data privacy scheme and its support of dynamic data.

FUTURE WORK:

1) Correctness: the verifier must accept all valid proof information generated by the cloud server;

2) Public Auditing: Any entity with public keys can audit the integrity of shared data without retrieving the data file back from the cloud;

3) Efficient User Revocation: once a user is revoked from the group, the cloud should be able to help group users update blocks tags generated by the revoked user;

4) Scalability: the data integrity auditing cost on users should be independent or grow practically slow (e.g., logarithmic) to the data size and the number of data modifiers.

5) Security Goals: if the data are corrupted, the cloud servers are not able to produce valid integrity proof information; any illegitimate user shall not be able to

impersonate valid users and generate legitimate tags behalf of valid users.

6. BIBILIOGRAPHY:

[1] Amazon. (2007) Amazon simple storage service (amazon s3).Amazon.[Online]. Available: http://aws.amazon.com/s3/
[2] Google. (2005) Google drive. Google.[Online]. Available: http://drive.google.com/
[3] Dropbox. (2007) A file-storage and sharing service.Dropbox.[Online]. Available: http://www.dropbox.com/
[4] Mozy. (2007) An online, data, and computer backup software. EMC.[Online]. Available: http://www.dropbox.com/
[5] Bitcasa. (2011) Inifinitestorage.Bitcasa.[Online]. Available: http://www.bitcasa.com/
[6] Memopal. (2007) Online backup. Memopal.[Online]. Available: http://www.memopal.com/ [7] M. A. et al., "Above the clouds: A berkeley view of cloud computing," Tech. Rep. UCBEECS, vol. 28, pp. 1–23, Feb. 2009.
[8] M. Rabin, "Efficient dispersal of information for security," Journal of the ACM (JACM), vol. 36(2), pp. 335–348, Apr. 1989.
[9] J. G. et al. (2006) The expanding digital universe: A forecast of worldwide information growth through 2010. IDC. [Online]. Available: Whitepaper
[10] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. of ACM CCS, Virginia, USA, Oct. 2007, pp. 598–609.
[11] A. Juels and B. S. Kaliski, "Pors: Proofs of retrievability for large files," in Proc. of ACM CCS, Virginia, USA, Oct. 2007, pp. 584–597.
[12] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: theory and implementation," in Proc. of CCSW 2009, llinois, USA, Nov. 2009, pp. 43–54