INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

SECURITY – CONSERVE COMMON EVALUATE FOR MUTUAL INFORMATION IN THE CLOUD

S.T.DEEPA[1], M.PRIYA[2]

[1]*Head, Department of Computer Science,Shri S.S.Shasun Jain College,deepatheodore@gmail.com*
[2] *M.Priya,Research scholar, Mother Teresa Women's University, priya.mayavan@gmail.com*
Author Correspondence: Head, Department of Computer Science,Shri S.S.Shasun Jain College,3,Madley Road, T.Nagar, Chennai – 17, 9444024628, deepatheodore@gmail.com

Abstract: - Conjunct Scrutinize for shared data — while perpetuate identity privacy — remains to be an open challenge. In this paper, we propose the first concealment perpetuate mechanism that allows conjunct scrutinize for communal information to be stored in the cloud. In particular, we exploit ring signatures to compute the verification information needed to audit the integrity of communal data. The security of a cryptographic system can rely either on the computational infeasibility of breaking it (computational security), or on the theoretical impossibility of breaking it, even using infinite computing power (information-theoretic or unconditional security). However, in doing so, these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when fine-grained data access control is desired, and thus do not scale well. The problem of simultaneously achieving scalability and data confidentiality of access control actually still remains unresolved. We achieve this goal by exploiting and uniquely combining techniques of Attribute-Based Encryption (ABE), Proxy Re-Encryption, and Lazy Re-Encryption. Our proposed scheme also has salient properties of user access privilege confidentiality and user secret key accountability. Extensive analysis shows that our proposed scheme is highly efficient and provably secures under existing security models.

Keywords*:* Attribute-Based Encryption, cloud, mutual information.

1. Introduction

A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour; it is elastic -- a user can have as much or as little of a service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access). Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing.

A cloud can be private or public. A public cloud sells services to anyone on the Internet. (Currently, Amazon Web Services is the largest public cloud provider.) A private cloud is a proprietary network or a data center that supplies hosted services to a limited number of people. When a service provider uses public cloud resources to create their private cloud, the result is called a virtual private cloud. Private or public, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services.

In this proposed system mainly focus two interesting problems, one of them is traceability, which means the ability for the group manager to reveal the identity of the signer based on verification metadata in some special situations. It's based on ring signatures, where the identity of the signer is unconditionally protected. Another problem is how to prove data freshness (prove the cloud possesses the latest version of shared data) while still perpetuate identity privacy. Another similar model called Proofs of Retrievability (POR), which is also able to check the correctness of data on an untrusted server. The original file is added with a set of randomly-valued check blocks called sentinels. The verifier challenges the untrusted server by specifying the positions of a collection of sentinels and asking the untrusted server to return the associated sentinel values. The first scheme is built from BLS signatures, and the second one is based on pseudo-random functions.

2. Related Work

[1] An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences: (1) Couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key. (2) A message can be "signed" using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in "electronic mail" and "electronic funds transfer" systems. A message is encrypted by representing it as a number M, raising M to a publicly specified power e, and then taking the remainder when the result is divided by the publicly specified product, n, of two large secret primer numbers p and q. Decryption is similar; only a different, secret, power d is used, where $e * d \equiv 1(\text{mod } (p - 1) * (q - 1))$. The security of the system rests in part on the difficulty of factoring the published divisor, n.

[2] We introduce a short signature scheme based on the Computational Diffie–Hellman assumption on certain elliptic and hyper elliptic curves. For standard security parameters, the signature length is about half that of a DSA signature with a similar level of security. Our short signature scheme is designed for systems where signatures are typed in by a human or are sent over a low-bandwidth channel. We survey a number of properties of our signature scheme such as signature aggregation and batch verification.

[3] In this paper we formalize the notion of a ring signature, which makes it possible to specify a set of possible signers without revealing which member actually produced the signature. Unlike group signatures, ring signatures have no group managers, no setup procedures, no revocation procedures, and no coordination: any user can choose any set of possible signers that includes himself, and sign any message by using his secret key and the others' public keys, without getting their approval or assistance. Ring signatures provide an elegant way to leak authoritative secrets in an anonymous way, to sign casual email in a way which can only be verifed by its intended recipient, and to solve other problems in multiparty computations. The main contribution of this paper is a new construction of such signatures which is unconditionally signer-ambiguous, provably secure in the random oracle model, and exceptionally efficient: adding each ring member increases the cost of signing or verifying by a single modular multiplication and a single symmetric encryption.

[4] An aggregate signature scheme is a digital signature that supports aggregation: Given *n* signatures on *n* distinct messages from *n* distinct users, it is possible to aggregate all these signatures into a single short signature. This single signature (and the *n* original messages) will convince the verifier that the *n* users did indeed sign the *n* original messages (i.e., user *i* signed message *Mi* for *i* = 1, . . . , *n*). In this paper we introduce the concept of an aggregate signature, present security models for such signatures, and give several applications for aggregate signatures. We construct an efficient aggregate signature from a recent short signature scheme based on bilinear maps due to Boneh, Lynn, and Shacham. Aggregate signatures are useful for reducing the size of certificate chains (by aggregating all signatures in the chain) and for reducing message size in secure routing protocols such as SBGP. We also show that aggregate signatures give rise to verifiably encrypted signatures. Such signatures enable the verifier to test that a given ciphertext *C* is the encryption of a signature on a given message *M*. verifiably encrypted signatures are used in contract-signing protocols. Finally, we show that similar ideas can be used to extend the short signature scheme to give simple ring signatures.

[5] We construct a short group signature scheme. Signatures in our scheme are approximately the size of a standard RSA signature with the same security. Security of our group signature is based on the Strong Diffie-Hellman assumption and a new assumption in bilinear groups called the Decision Linear assumption. We prove security of our system, in the random oracle model, using a variant of the security definition for group signatures recently given by Bellare, Micciancio, and Warinschi.

[6] This paper describes the direct anonymous attestation scheme (DAA). This scheme was adopted by the Trusted Computing Group (TCG) as the method for remote authentication of a hardware module, called Trusted Platform Module (TPM), while preserving the privacy of the user of the platform that contains the module. DAA can be seen as a group signature without the feature that a signature can be opened, i.e., the anonymity is not revocable. Moreover, DAA allows for pseudonyms, i.e., for each signature a user (in agreement with the recipient of the signature) can decide whether or not the signature should be linkable to another signature. DAA furthermore allows for detection of "known" keys: if the DAA secret keys are extracted from a TPM and published, a verifier can detect that a signature was produced using these secret keys. The scheme is provably secure in the random oracle model under the strong RSA and the decisional Diffie-Hellman assumption

[7] We introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems. We present two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

[8] In this paper, we define and explore *proofs of retrievability* (PORs). A POR scheme enables an archive or back-up service (prover) to produce a concise proof that a user (verifier) can retrieve a target file *F*, that is, that the archive retains and reliably transmits file data sufficient for the user to recover *F* in its entirety.

[9] In a proof-of-retrievability system, a data storage center convinces a verifier that he is actually storing all of a client’s data. The central challenge is to build systems that are both efficient and provably secure—that is, it should be possible to extract the client’s data from any prover that passes a verification check. In this paper, we give the first proof-of-retrievability schemes with full proofs of security against arbitrary adversaries in the strongest model, that of Juels and Kaliski. Our first scheme, built from BLS signatures and secure in the random oracle model, has the shortest query and response of any proof-of-retrievability with public verifiability. Our second scheme, which builds elegantly on pseudorandom functions (PRFs) and is secure in the standard model, has the shortest response of any proof-of-retrievability scheme with private verifiability (but a longer query). Both schemes rely on homomorphic properties to aggregate a proof into one small authenticator value.

[10] Storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated in the past. However, Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature. The main issue is how to frequently, efficiently and securely verify that a storage server is faithfully storing its client's (potentially very large) outsourced data. The storage server is assumed to be untrusted in terms of both security and reliability. In this paper, we construct a highly efficient and provably secure PDP technique based entirely on symmetric key cryptography, while not requiring any bulk encryption. Also, in contrast with its predecessors, our PDP technique allows outsourcing of dynamic data, i.e, it efficiently supports operations, such as block modification, deletion and append.

[11] As storage-outsourcing services and resource-sharing networks have become popular, the problem of efficiently proving the integrity of data stored at untrusted servers has received increased attention. In the provable data possession (PDP) model, the client preprocesses the data and then sends it to an untrusted server for storage, while keeping a small amount of meta-data. The client later asks the server to prove that the stored data has not been tampered with or deleted (without downloading the actual data). However, the original PDP scheme applies only to static (or append-only) files. We present a definitional framework and efficient

constructions for dynamic provable data possession (DPDP), which extends the PDP model to support provable updates to stored data. We use a new version of authenticated dictionaries based on rank information. The price of dynamic updates is a performance change from O(1) to O(log n) for a file consisting of n blocks, while maintaining the same (or better, respectively) probability of misbehavior detection. Our experiments show that this slowdown is very low in practice (e.g., 415KB proof size and 30ms computational overhead for a 1GB file). We also show how to apply our DPDP scheme to outsourced file systems and version control systems (e.g., CVS).

[12] Cloud computing is an emerging computing paradigm in which resources of the computing infrastructure are provided as services over the Internet. As promising as it is, this paradigm also brings forth many new challenges for data security and access control when users outsource sensitive data for sharing on cloud servers, which are not within the same trusted domain as data owners. To keep sensitive user data confidential against untrusted servers, existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users. However, in doing so, these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when fine-grained data access control is desired, and thus do not scale well. The problem of simultaneously achieving fine-grainedness, scalability, and data confidentiality of access control actually still remains unresolved. This paper addresses this challenging open issue by, on one hand, defining and enforcing access policies based on data attributes, and, on the other hand, allowing the data owner to delegate most of the computation tasks involved in fine-grained data access control to untrusted cloud servers without disclosing the underlying data contents. We achieve this goal by exploiting and uniquely combining techniques of attribute-based encryption (ABE), proxy re-encryption, and lazy re-encryption. Our proposed scheme also has salient properties of user access privilege confidentiality and user secret key accountability. Extensive analysis shows that our proposed scheme is highly efficient and provably secure under existing security models.

[13] Remote Data Checking (RDC) is a technique by which clients can establish that data outsourced at untrusted servers remains intact over time. RDC is useful as a prevention tool, allowing clients to periodically check if data has been damaged, and as a repair tool whenever damage has been detected. Initially proposed in the context of a single server, RDC was later extended to verify data integrity in distributed storage systems that rely on replication and on erasure coding to store data redundantly at multiple servers. Recently, a technique was proposed to add redundancy based on network coding, which offers interesting tradeoffs because of its remarkably low communication overhead to repair corrupt servers. Unlike previous work on RDC which focused on minimizing the costs of the prevention phase, we take a holistic look and initiate the investigation of RDC schemes for distributed systems that rely on network coding to minimize the combined costs of both the prevention and repair phases. We propose RDC-NC, a novel secure and efficient RDC scheme for network coding-based distributed storage systems. RDC-NC mitigates new attacks that stem from the underlying principle of network coding. The scheme is able to preserve in an adversarial setting the minimal communication overhead of the repair component achieved by network coding in a benign setting. We implement our scheme and experimentally show that it is computationally inexpensive for both clients and servers.

[14] Cloud Computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased. Developers with innovative ideas for new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it. They need not be concerned about over-provisioning for a service whose popularity does not meet their predictions, thus wasting costly resources, or under-provisioning for one that becomes wildly popular, thus missing potential customers and revenue. Moreover, companies with large batch-oriented tasks can get results as quickly as their programs can scale, since using 1000 servers for one hour costs no more than using one server for 1000 hours. This elasticity of resources, without paying a premium for large scale, is unprecedented in the history of IT. As a result, Cloud Computing is a popular topic for blogging and white papers and been featured in the title of workshops, conferences, and even magazines. Our goal in this paper to reduce that confusion by clarifying terms, providing simple figures to quantify comparisons between of cloud and conventional Computing, and identifying the top technical and non-technical obstacles and opportunities of Cloud Computing.

[15] Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the possibly large size of outsourced data makes the data integrity protection in Cloud Computing a very challenging and potentially formidable task, especially for users with constrained computing resources and capabilities. Thus, enabling public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed. To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met: 1. TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user; 2. The third party auditing process should bring in no new vulnerabilities towards user data privacy. So, here by utilizing and uniquely combining the public key based homomorphic authenticator with random masking to achieve the privacy-preserving public cloud data auditing system, which meets all above requirements. To support efficient handling of multiple auditing tasks, further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient.

[16] In this paper, we propose a dynamic audit service for verifying the integrity of an untrusted and outsourced storage. Our audit service is constructed based on the techniques, fragment structure, random sampling, and index-hash table, supporting provable updates to outsourced data and timely anomaly detection. In addition, we propose a method based on probabilistic query and periodic verification for improving the performance of audit services. Our experimental results not only validate the effectiveness of our approaches, but also show our audit system verifies the integrity with lower computation overhead and requiring less extra storage for audit metadata.

[17] With the increasing adoption of cloud computing for data storage, assuring data service reliability, in terms of data correctness and availability, has been outstanding. While redundancy can be added into the data for reliability, the problem becomes challenging in the "pay-as-you-use" cloud paradigm where we always want to efficiently resolve it for both corruption detection and data repair. Prior distributed storage systems based on erasure codes or network coding techniques have either high decoding computational cost for data users, or too much burden of data repair and being online for data owners. In this paper, we design a secure cloud storage service which addresses the reliability issue with near-optimal overall performance. By allowing a third party to perform the public integrity verification, data owners are significantly released from the onerous work of periodically checking data integrity. To completely free the data owner from the burden of being online after data outsourcing, this paper proposes an exact repair solution so that no metadata needs to be generated on the fly for repaired data. The performance analysis and experimental results show that our designed service has comparable storage and communication cost, but much less computational cost during data retrieval than erasure codes-based storage solutions. It introduces less storage cost, much faster data retrieval, and comparable communication cost comparing to network coding-based distributed storage systems.

[18] With cloud computing and storage services, data is not only stored in the cloud, but routinely shared among a large number of users in a group. It remains elusive, however, to design an efficient mechanism to audit the integrity of such shared data, while still preserving identity privacy. In this paper, we propose Knox, a privacy-preserving auditing mechanism for data stored in the cloud and shared among a large number of users in a group. In particular, we utilize group signatures to construct homomorphic authenticators, so that a third party auditor (TPA) is able to verify the integrity of shared data for users without retrieving the entire data. Meanwhile, the identity of the signer on each block in shared data is kept private from the TPA. With Knox, the amount of information used for verification, as well as the time it takes to audit with it, are not affected by the number of users in the group. In addition, Knox exploits homomorphic MACs to reduce the space used to store such verification information. Our experimental results show that Knox is able to efficiently audit the correctness of data, shared among a large number of users.

[19] Offering strong data protection to cloud users while enabling rich applications is a challenging task. Researchers explore a new cloud platform architecture called Data Protection as a Service, which

dramatically reduces the per-application development effort required to offer data protection, while still allowing rapid development and maintenance.

[20] With data storage and sharing services in the cloud, users can easily modify and share data as a group. To ensure shared data integrity can be verified publicly, users in the group need to compute signatures on all the blocks in shared data. Different blocks in shared data are generally signed by different users due to data modifications performed by different users. For security reasons, once a user is revoked from the group, the blocks which were previously signed by this revoked user must be re-signed by an existing user. The straightforward method, which allows an existing user to download the corresponding part of shared data and re-sign it during user revocation, is inefficient due to the large size of shared data in the cloud. In this paper, we propose a novel public auditing mechanism for the integrity of shared data with efficient user revocation in mind. By utilizing the idea of proxy re-signatures, we 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. Moreover, our mechanism is able to support batch auditing by verifying multiple auditing tasks simultaneously. Experimental results show that our mechanism can significantly improve the efficiency of user revocation.

[21] With the character of low maintenance, cloud computing provides an economical and efficient solution for sharing group resource among cloud users. Unfortunately, sharing data in a multi-owner manner while preserving data and identity privacy from an untrusted cloud is still a challenging issue, due to the frequent change of the membership. In this paper, we propose a secure multi-owner data sharing scheme, named Mona, for dynamic groups in the cloud. By leveraging group signature and dynamic broadcast encryption techniques, any cloud user can anonymously share data with others. Meanwhile, the storage overhead and encryption computation cost of our scheme are independent with the number of revoked users. In addition, we analyze the security of our scheme with rigorous proofs, and demonstrate the efficiency of our scheme in experiments.

[22] The emergence of cloud computing brings users abundant opportunities to utilize the power of cloud to perform computation on data contributed by multiple users. These cloud data should be encrypted under multiple keys due to privacy concerns. However, existing secure computation techniques are either limited to single key or still far from practical. Here two efficient schemes for secure outsourced computation over cloud data encrypted under multiple keys. Two non-colluding cloud servers to jointly compute polynomial functions over multiple users' encrypted cloud data without learning the inputs, intermediate or final results, and require only minimal interactions between the two cloud servers but not the users. We demonstrate our schemes' efficiency experimentally via applications in machine learning. Our schemes are also applicable to privacy-preserving data aggregation such as in smart metering.

[23] Due to the existence of security threats in the cloud, many mechanisms have been proposed to allow a user to audit data integrity with the public key of the data owner before utilizing cloud data. The correctness of choosing the right public key in previous mechanisms depends on the security of Public Key Infrastructure (PKI) and certificates. Although traditional PKI has been widely used in the construction of public key cryptography, it still faces many security risks, especially in the aspect of managing certificates. In this paper, we design a certificateless public auditing mechanism to eliminate the security risks introduced by PKI in previous solutions. Specifically, with our mechanism, a public verifier does not need to manage certificates to choose the right public key for the auditing. Instead, the auditing can be operated with the assistance of the data owner's identity, such as her name or email address, which can ensure the right public key is used. Meanwhile, this public verifier is still able to audit data integrity without retrieving the entire data from the cloud as previous solutions. To the best of our knowledge, it is the first certificateless public auditing mechanism for verifying data integrity in the cloud. Our theoretical analyses prove that our mechanism is correct and secure, and our experimental results show that our mechanism is able to audit the integrity of data in the cloud efficiently.

[24] 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 burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes 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 auditability for cloud storage is of critical importance 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 toward user data privacy, and introduce no additional online burden to user. In this paper, 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. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

[25] In the cloud, data is often shared by a group of users. To ensure the long-term correctness of cloud shared data, a third-party public verifier can be introduced to audit data integrity. During the auditing, protecting the privacy of the contributors of shared data from the public auditor is a fundamental issue. However, this makes it challenging to simultaneously support group membership dynamics efficiently, due to the significant amount of computation needed to update the signatures on shared data. In this paper, we propose a novel privacy-preserving public auditing mechanism for shared cloud data. With our proposed mechanism, a public verifier is able to audit the integrity of shared data without retrieving the entire data from the cloud, and also without learning private identity information of the group members. Group dynamics (user join and user revocation) are efficiently handled by outsourcing signature updating operations to the cloud via a secure proxy re-signature scheme. Experimental results show that our mechanism is highly efficient for dynamic groups.

[26] Nowadays, many organizations outsource data storage to the cloud such that a member (owner) of an organization can easily share data with other members (users). Due to the existence of security concerns in the cloud, both owners and users are suggested to verify the integrity of cloud data with Provable Data Possession (PDP) before further utilization on data. However, previous methods either unnecessarily reveal the identity of a data owner to the untrusted cloud or any public verifiers, or introduce significant overheads on verification metadata to preserve anonymity. In this paper, we propose a simple and efficient publicly verifiable approach to ensure cloud data integrity without sacrificing the anonymity of data owners nor requiring significant verification metadata. Specifically, we introduce a security-mediator (SEM), which is able to generate verification metadata (i.e., signatures) on outsourced data for data owners. Our approach decouples the anonymity protection mechanism from the PDP. Thus, an organization can employ its own anonymous authentication mechanism, and the cloud is oblivious to that since it only deals with typical PDP-metadata, Consequently, there is no extra storage overhead when compared with existing non-anonymous PDP solutions. The distinctive features of our scheme also include data privacy, such that the SEM does not learn anything about the data to be uploaded to the cloud at all, which is able to minimize the requirement of trust on the SEM. In addition, we can also extend our scheme to work with the multi-SEM model, which can avoid the potential single point of failure existing in the single-SEM scenario. Security analyses prove our scheme is secure, and experiment results demonstrate our scheme is efficient.

[27] This document describes the MD5 message-digest algorithm. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

3. Conjunct Scrutinize for shared data

In this proposed system mainly focus two interesting problems, one of them is traceability, which means the ability for the group manager to reveal the identity of the signer based on verification metadata in some special situations. It's based on ring signatures, where the identity of the signer is unconditionally protected. Another problem is how to prove data freshness (prove the cloud possesses the latest version of shared data) while still perpetuate identity privacy. Another similar model called Proofs of Retrievability (POR), which is also able to check the correctness of data on an untrusted server. The original file is added with a set of randomly-valued check blocks called sentinels. The verifier challenges the untrusted server by specifying the positions of a collection of sentinels and asking the untrusted server to return the associated sentinel values. The first scheme is built from BLS signatures, and the second one is based on pseudo-random functions.

Key Generation

Key generation is the process of generating keys for encrypting and decrypting the files. To create a private key and public key .

The public key is made available to anyone. A sender encrypts data with the public key only the holder of the private key can decrypt this data. In this we can change the data as unreadable format by using encryption method which is mentioned above. If any un-authorized persons are opened and trying to do something then they cannot understand that data, so by using this we secure our important files.

Decrypt the file

Decryption is the reverse process to encryption and decryption is user understandable language. Database decryption is converting the encryption cipher text into the original information using keys generated by the encryption algorithms. Here if you decrypt the file that time you will enter the key after that only your decrypt that particular file why because here that key is provide security for your personal data.

Cloud Verifier

Cloud Verifier a cloud storage framework that provides complete, correct, accurate and verifiable data file service to users. Users can leverage such framework to obtain a correct view of the runtime state of their computing environment and perform responsive reaction upon anomalies.

The Cloud verifier to compare cloud storage data and these data same or not after that same means it provide high security the cloud's hierarchical structure to build transitive trust starting in the cloud platform up to the instances themselves. Platform states are monitored by a Cloud Verifier against the cloud administrator's specified criteria, thereby preventing maliciously modified systems from executing data to user.

Cloud Signatures

Cloud Signature Support is an exclusive technical benefit that provides qualified cloud competency partners with an elevated level of technical support for select Microsoft cloud products. You will receive to original data that will work directly with you, have extensive product-specific knowledge, and are accountable for driving cases from start to finish. Where the identity of the signer is unconditionally protected the current design of ours

Here support traceability. Each user makes his encryption key public, and keeps the corresponding decryption key private signatures People and organizations buy or lease storage capacity from the providers to store user, organization, or Application data Cloud storage has several advantages over traditional data storage.

Cloud File Generation

File Generation to store and share a data file in the cloud, a group member performs the following operations: Getting the revocation list from the cloud. In this step, the member sends the group identity I group as a request to the cloud. Then, the cloud responds the revocation list RL to the member. Verifying the validity of the received revocation list.

On receiving the data, the cloud first checks its validity. If the algorithm returns true, the group signature is valid; otherwise, the cloud abandons the data. In addition, if several users have been revoked by the group manager, the cloud also performs revocation verification.

1. ATTRIBUTE-BASED ENCRYPTION (ABE)

ABE was proposed by Sahai and Waters. In traditional public-key cryptography, a message is encrypted for a specific receiver using the receiver’s public-key. Identity-based cryptography and in particular identity-based encryption (IBE) changed the traditional understanding of public-key cryptography by allowing the public-key to be an arbitrary string, e.g., the email address of the receiverIn ABE, a user has a set of attributes in addition to its unique ID. There are two classes of ABEs. In Key-policy ABE or KP-ABE is a public key cryptography. Here the sender has an access policy to encrypt data. A writer whose attributes and keys have been revoked cannot write back stale information. The receiver receives attributes and secret keys from the attribute authority and is able to decrypt information if it has matching attributes

In ciphertext-policy attribute-based encryption (CP-ABE) a user's private-key is associated with a set of attributes and a ciphertext specifies an access policy over a defined universe of attributes within the system. A user will be able to decrypt a ciphertext, if and only if his attributes satisfy the policy of the respective ciphertext.

Here KP-ABE is used to generate keys by the cloud user to store and retrieve informations in the cloud. To provide a fine-grained access control

2. PROXY RE-ENCRYPTION

The proxy re-encryption schemes are proposed by Mambo and Okamoto and Blaze et al. Proxy re-encryption is a cryptographic primitive which translates ciphertexts from one encryption key to another encryption key. It can be used to forward encrypted messages without having to expose the cleartexts to the potential users. The re-encryption protocol should be key independent to avoid compromising the private keys of the sender and the
Recipient.

The primary advantage of this PRE scheme is that they are unidirectional (i.e., Alice can delegate to Bob without Bob having to delegate to her) and do not require delegators to reveal their entire secret key to anyone.

A proxy re-encryption algorithm transforms a cipher text under a public key PKA to cipher text PKB by using the re-encryption key RK A →B. The server does not know the corresponding clear text, where PKA and PKB can only be decrypted by different key KA and KB respectively. Proxy re-encryption has many applications in addition to the previous proposals for email forwarding, secure network file storage, and performing cryptographic operations on storage limited devices.

3. LAZY RE-ENCRYPTION

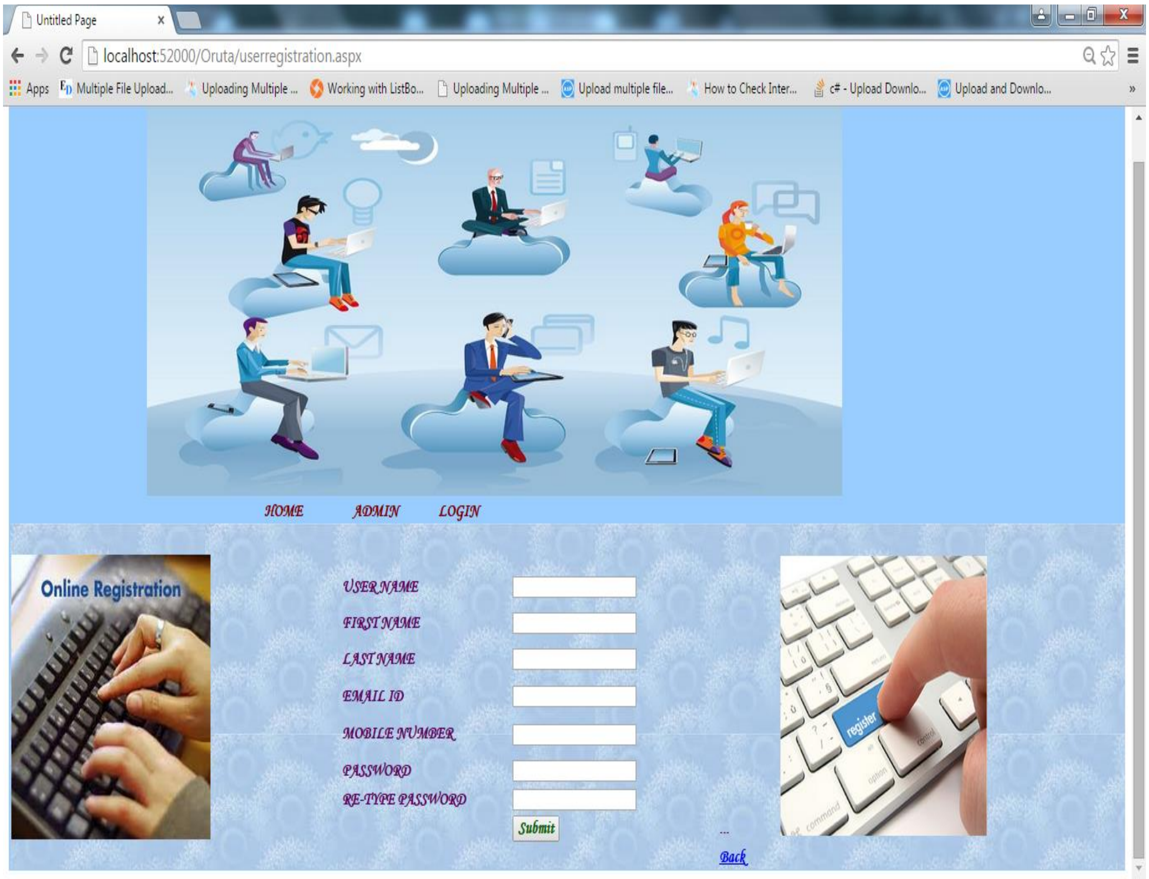
The lazy re-encryption technique and allow Cloud Servers to aggregate computation tasks of multiple operations. The operations such as

_ Update secret keys
_ Update user attributes
_ Files are not re-encrypted until a user wants access
_Spreads out the re-encryption over time to speed up access with the third party

ADVANTAGES OF PROPOSED WORK

- Revoked users cannot access data after they have been revoked.
- The access control and authentication are both collusion resistant, meaning that no two users can collude and access data or authenticate themselves, if they are individually not authorized
- Authentication of users who store and modify their data on the cloud.
- The identity of the user is protected from the cloud during authentication
- Low initial capital investment
- Shorter start-up time for new services
- Lower maintenance and operation costs
- Higher utilization through virtualization

- Easier disaster recovery



USER NAME

FIRST NAME

LAST NAME

EMAIL ID

MOBILE NUMBER

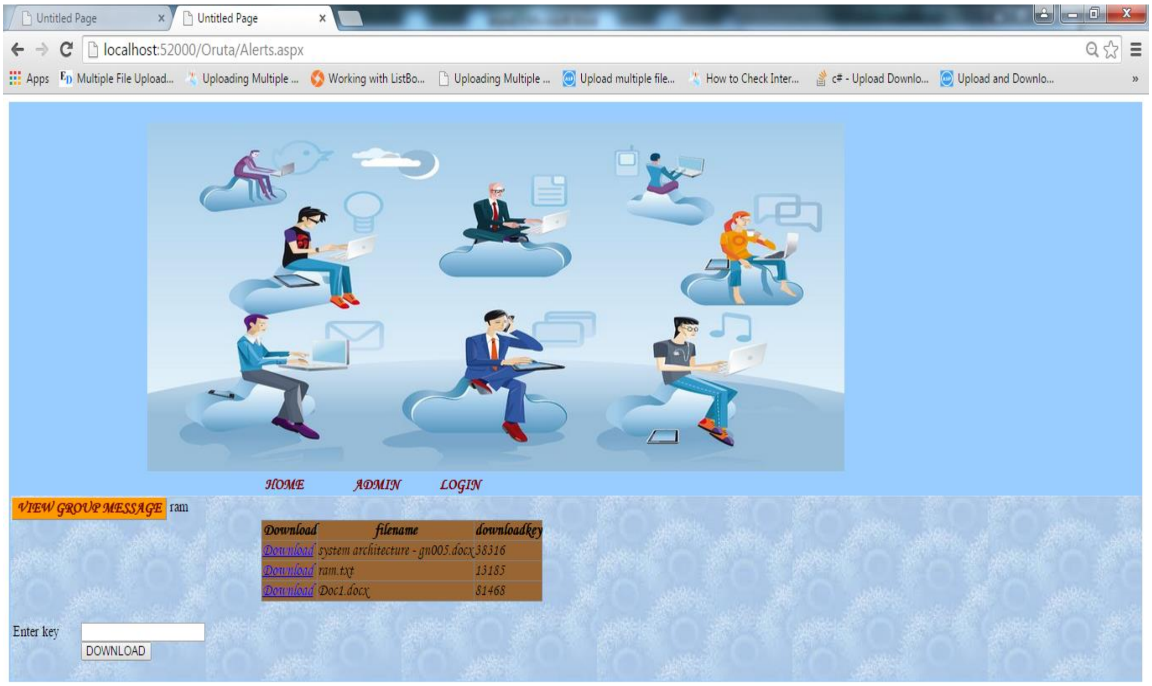
PASSWORD

RE-TYPE PASSWORD

Submit

Back

Fig 1: User Registration



VIEW GROUP MESSAGE ram

Download	filename	downloadkey
Download	system architecture - gn005.docx	38316
Download	ram.txt	13185
Download	Doc1.docx	81468

Enter key

DOWNLOAD

Fig 2: Key Matching and Download page for key messages

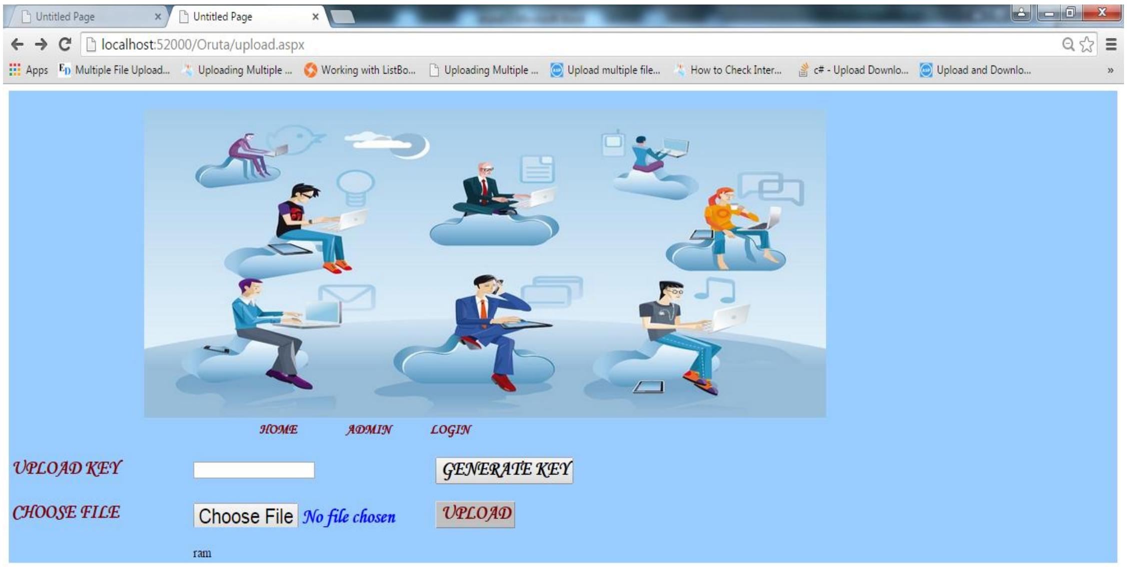


Fig 3: File upload page

4. Conclusion

The proposed system is for data storage security in Cloud Computing. Confidentiality of user access privilege and user secret key accountability can be achieved. Extensive analysis shows that the proposed scheme is highly efficient and provably secures under existing security models.

In the proposed system third party auditor (TPA) handled multiple auditing sessions from different users for outsourced data .In future this task can be extended into a multi-user setting TPA who performs multiple auditing tasks in a batch manner to give better efficiency compared to the proposed work

REFERENCES

1. R. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public Key Cryptosystems,” Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978.
2. D. Boneh, B. Lynn, and H. Shacham, “Short Signatures from the Weil Pairing,” Proc. Seventh Int’l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT’01), pp. 514-532, 2001.
3. R.L. Rivest, A. Shamir, and Y. Tauman, “How to Leak a Secret,” Proc. Seventh Int’l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT’01), pp. 552-565, 2001.
4. D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” Proc. 22nd Int’l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT’03), pp. 416-432, 2003
5. D. Boneh, X. Boyen, and H. Shacham, “Short Group Signatures,” Proc. 24th Ann. Int’l Cryptology Conf. (CRYPTO’04), pp. 41-55, 2004.
6. E. Brickell, J. Camenisch, and L. Chen, “Direct Anonymous Attestation,” Proc. 11th ACM Conf. Computer and Comm. Security (CCS’04), pp. 132-145, 2004.
7. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted Stores,” Proc. 14th ACM Conf. Computer and Comm. Security (CCS ’07), pp. 598-610, 2007.
8. Juels and B.S. Kaliski, “PORs: Proofs of Retrievability for Large Files,” Proc. 14th ACM Conf. Computer and Comm. Security (CCS’07), pp. 584-597, 2007.
9. H. Shacham and B. Waters, “Compact Proofs of Retrievability,” Proc. 14th Int’l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT ’08), pp. 90-107, 2008.

10. G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession,” Proc. Fourth Int’l Conf. Security and Privacy in Comm. Networks (SecureComm’08), 2008

11. C. Chris Erway , Alptekin K¨upc¸ ¨u , Charalampos Papamanthou, Roberto Tamassia “Dynamic Provable Data Possession” Brown University, Providence RI November 29, 2009
12. S. Yu et al., “Achieving Secure, Scalable, and Fine-Grained Access Control in Cloud Computing,” Proc. 30th IEEE Int’l Conf. Computer Communications (INFOCOM 10), IEEE Press, 2010, pp. 534–542
13. B. Chen, R. Curtmola, G. Ateniese, and R. Burns, “Remote Data Checking for Network Coding-Based Distributed Storage Systems,” Proc. ACM Workshop Cloud Computing Security Workshop (CCSW’10), pp. 31-42, 2010

14. 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.
15. C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,” Proc. IEEE INFOCOM, pp. 525-533, 2010.
16. Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S Yau, “Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds,” Proc. ACM Symp. Applied Computing (SAC’11), pp. 1550-1557, 2011.
17. N. Cao, S. Yu, Z. Yang, W. Lou, and Y.T. Hou, “LT Codes-Based Secure and Reliable Cloud Storage Service,” Proc. IEEE INFOCOM, 2012.

18. B. Wang, B. Li, and H. Li, “Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud,” Proc. 10th Int’l Conf. Applied Cryptography and Network Security (ACNS’12), pp. 507-525, June 2012.

19. D. Song, E. Shi, I. Fischer, and U. Shankar, “Cloud Data Protection for the Masses,” Computer, vol. 45, no. 1, pp. 39-45, 2012.
20. B. Wang, B. Li, and H. Li, “Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud,” IEEE Trans. Services Computing, 20 Dec. 2013
21. X. Liu, Y. Zhang, B. Wang, and J. Yan, “Mona: Secure Multi- Owner Data Sharing for Dynamic Groups in the Cloud,” IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1182-1191, June 2013.

22. B. Wang, M. Li, S.S. Chow, and H. Li, “Computing Encrypted Cloud Data Efficiently under Multiple Keys,” Proc. IEEE Conf. Comm. and Network Security (CNS ’13), pp. 90-99, 2013.
23. B. Wang, B. Li, and H. Li, “Certificate less Public Auditing for Data Integrity in the Cloud,” Proc. IEEE Conf. Comm. and Network Security (CNS’13), pp. 276-284, 2013
24. C. Wang, S.S. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage,” IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.

25. B. Wang, H. Li, and M. Li, “Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics,” Proc. IEEE Int’l Conf. Comm. (ICC’13), pp. 539-543, 2013.

26. B. Wang, S.S. Chow, M. Li, and H. Li, “Storing Shared Data on the Cloud via Security-Mediator,” Proc. IEEE 33rd Int’l Conf. Distributed Computing Systems (ICDCS’13), pp. 124-133, 2013.

27. The MD5 Message-Digest Algorithm (RFC1321). https://tools. ietf.org/html/rfc1321, 2014.

A Brief Author Biography

Dr.S.T.Deepa earned her research degree from Mother Teresa Women’s University. She did MCA degree from University of Madras, Chennai. The Mother Teresa University awarded her M.Phil. degree. She is the Head of the Department of Computer Science, Shri Shankarlal Sundarbai Shasun Jain College for Women, Chennai. Mother Theresa University, Kodaikanal have recognized her as a research guide for M.Phil. Computer Science. So far three candidates have completed their M.Phil. degree under his guidance, and 7 candidates pursuing research. He has published several research papers in national and international journals. He has organized Conferences and Seminars at state and national level.

Ms. M.Priya is a Mathematics graduate from Marudhar Kesari Jain College. She did her MCA in Sacred Heart College, Tirupattur. Currently she is perusing her M.Phil. In Mother Teresa Women’s University.