ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Decentralized Access Control with Anonymous Authentication of Data Stored in Cloud

Ch.M Siva Rama Krishna
MTech Student,
Department of CSE,
Loyola Institute of Technology and Management.

G.John Samuel Babu
Assistant Professor,
Department of CSE,
Loyola Institute of Technology and Management.

Abstract:

We propose a secure cloud storage model that addresses security and storage issues for cloud computing environments. Security is achieved by anonymous authentication which ensures that cloud users remain anonymous while getting duly authenticated.

For achieving this goal, we propose a digital signature based authentication scheme with a decentralized architecture for distributed key management with multiple Key Distribution Centers. Homomorphic encryption scheme using Paillier public key cryptosystem is used for encrypting the data that is stored in the cloud.

We incorporate a query driven approach for validating the access policies defined by an individual user for his/her data i.e. the access is granted to a requester only if his credentials matches with the hidden access policy. Further, since data is vulnerable to losses or damages due to the vagaries of the network, we propose an automatic retrieval mechanism where lost data is recovered by data replication and file replacement with string matching algorithm. We describe a prototype implementation of our proposed model.

Keywords:

Access Policies,Anonymous Authentication,DecentralizedArchitecture,Distributed Key Management, Homomorphic Encryption

1. INTRODUCTION:

Cloud computing is a collection of scalable resources and computing infrastructure which provides services to users with the “pay only for use” strategy. This kind of technology helps users in handling resources effectively on-site.

Though the advantages are clear, the critical factor in the present data outsourcing scenario is the enforcement of strong security mechanisms for data storage, transfer and processing in the cloud. The data that are stored in the cloud are often sensitive in nature. For example, medical records and user-driven data generated in social networks are often stored in public or private clouds.

Ensuring privacy and security of such data is important for users to trust the service providers. For achieving that, adequate authentication and access control techniques must be employed. A high level security system also ensures that only verified and valid services are provided to authorized users. Indeed, the process of authentication must be initiated for all valid transactions that are performed through the cloud.

The first goal of our work is to implement anonymous authentication of users. In [1], the authors discuss anonymous authentication of users and highlight its importance. The privacy settings of users must be followed in such a manner that the identity of the user should not become evident to either the cloud service providers or to other users. Thus, the anonymity of users is preserved.To provide secure data storage, the cloud data needs to be encrypted.

The second goal of our work is to ensure data privacy and security. Many homomorphic techniques have been discussed [2], [3]. This kind of encryption ensures that during the time that computations are performed on the data by a cloud’s computing resources, they are not able to read the data. For this, the data must be suitably encoded before being encrypted.

Cloud servers are prone to failures and attacks. Service providers should provide reliable and uninterrupted services to users by providing efficient retrieval mechanisms. Our third goal is to enhance the availability of cloud services. Wang et al. addressed the issue of secure and dependable cloud storage [4].

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

They specifically discussed about Byzantine failure, where the storage servers fail in arbitrary ways leading to data modification and loss. We deal with this issue by replicating data using backup files and recovering lost data with string matching algorithm.The rest of the paper is organized as follows. In section 2, we review prior work in the relevant domain .In section 3, we elaborate upon the details of our present work. In section 4, we describe a prototype implementation of our scheme. In section 5, we do a cryptanalysis of our system and assess its performance. We conclude our work in section 6 and project directions for further research.

2. PRIOR WORK:

We now take a brief survey of the existing approaches for handling various security issues such as key distribution, access control and authentication.

2.1.KEY DISTRIBUTION ARCHITECTURES:

The centralized architecture model implements a single Key Distribution Center (KDC) for key distribution as well as for incorporating security mechanism. Several existing works discuss about centralized access control mechanisms [5], [6], [7], [8], [17]. Though implementation of a single KDC structure is convenient, but it faces many potential problems:

•A critical problem is that of single point failure which is not at all desirable in a cloud environment where there are large numbers of active users.

•Significant overheads occur since a single KDC is used to distribute secret keys and attributes to all users.

Furthermore, the schemes discussed in [10] and [7] do not support authentication. In [10], the security system supports only single write and read operation.In view of the above problems, a decentralized cloud approach is emphasized where the task of key management isdone by multiple KDCs. A decentralized architecture for distributed key management is presented in [1]. However, in this work, access policies defined by a user from other users of the file. Thus, access rights associated with individual users are not hidden from the cloud.

2.2 AUTHENTICATION TECHNIQUES:

In [1], the authors describe anonymous authentication where users are authenticated without their identity being revealed. This approach is very useful in a real time scenario where users want to post some sensitive information without being recognized. Nevertheless, users should be able to prove that he/she is a valid user who has posted the information.For achieving anonymous authentication of users, cryptographic techniques and protocols are used. Digital signatures can uniquely authenticate the users and also help in detecting any unauthorized modifications to the data.

Digital Signature Standard provides a framework for generating digital signatures. Some commonly used signature techniques are ring, mesh and group signatures [11]. None of these techniques are quite feasible solutions for providing authentication for cloud users as the number of users is typically very large. Group signatures are also not possible in cloud services since predefined groups should be assumed which is usually not the case in cloud services. Similarly in mesh signatures, the identification of the exact source of information is not possible which makes the system vulnerable to colluding attacks.

2.3 DATA SECURITY:

As discussed earlier, most of the data that are outsourced are sensitive in nature. They are stored in servers located externally in different locations. The cloud service providers should adopt and use strong cryptographic techniques for handling the data with utmost security and safety. Though the paper [1] presented a decentralized architecture with anonymous authentication the data access policies and attributes associated with individual users are not hidden from cloud.

2.4 ACCESS CONTROL TECHNIQUES:

The following types of access controls techniques are commonly used:

•User Based Access Control (UBAC) [11] – An Access Control List contains the details of access rights defined for all users on different resources that are offered by the computing system.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

This method is not suitable for cloud services because of scalability issues [1]. It is difficult to update, maintain and store Access Control List (ACL). Moreover, for every operation the ACL needs to be referenced, which creates a performance bottleneck.

•Mandatory Access Control (MAC) [11] [14] – Users alone do not have the right to decide on their access control privileges. Rather, it is based on the combination of (i) The security levels associated with the data itself. These are defined by the metadata security labels according the sensitivity of data within a Multi-Level Secure (MLS) framework [15] (ii) The security clearance given to the individual processes that access data. MAC is designed forsingle write and read operation.

In view of the above problems, a decentralized cloud approach is emphasized where the task of key management is military based security applications and is not suitable for commercial cloud based applications.

•Discretionary Access Control (DAC) [16] – The owners of the resources decide on the access rights for different users for these resources. The DAC technique is not always a suitable form of security mechanism for cloud services because with multiple users sharing information, it becomes very tedious to define the rights for each user.

•Attribute Based Access Control (ABAC) – The users get access to various resources based on user attributes that include the corresponding access policy. This too is not suitable for secure cloud-based applications as the access policies depend only upon user attributes and cannot be defined or changed dynamically for each user. Moreover, it is not possible to maintain anonymity of users.

In our proposed security framework, we implement the Role Based Access Control (RBAC) model [11]. We preferred the RBAC access control method where users are classified based on their roles and the access rights are defined accordingly.

The implementation of anonymous authentication in RBAC is a challenging process and forms a new combination of secure cryptosystem in a cloud environment. There is a centralized administered control that defines the structure for interaction between "subjects" and "objects". The subjects are entities to which execution can be attributed such as users, processes, threads, or even procedure activations.

Objects are entities on which operations are defined including storage abstractions such as memory or files with read, write, and execute operations and code abstractions, such as modules or services with operations to initiate or suspend execution. Distinct privileges are typically associated with distinct operations on different objects [17]. In Table.1, we show a comparison of a number of past approaches for access control with the scheme proposed by us. It is quite evident that our decentralized scheme given in the last row is powered by the maximum number of features. It has multiple read and multiple write access, homomorphic encryption and performs anonymous authentication while hiding user attributes.

Table.1. Comparison of Proposed scheme with other Access Control schemes:

Ref Paper	Architecture	Write/read access	Type of Encryption	Privacy preserving authentication	User Revocation
[6]	Centralized	1-W-M-R	Symmetric key cryptography	No Authentication	No
[7]	Centralized	1-W-M-R	ABE	No Authentication	No
[10]	Decentralized	1-W-M-R	ABE	No Authentication	Yes
[12]	Centralized	1-W-M-R	ABE	No Authentication	Yes
[8]	Decentralized	1-W-M-R	ABE	No privacy preservation	No
[1]	Decentralized	M-W-M-R	ABE	Authentication	Yes
Our Scheme	Decentralized	M-W-M-R	Homomorphic Encryption combined with ABE	Authentication with attribute based hidden policy	Yes

Legend: W- Write, M- Multi, R- Read, ABE- Attribute Based Encryption [1,6]

3. PROPOSED WORK:

3.1 OVERVIEW:

The Fig.1 gives an overview of our proposed work. We focus on improving upon two main areas as discussed below.

3.1.1.Security Related Improvements:

1)We implement a decentralized architecture is implemented with multiple Key Distribution Centre (KDC) structure [1].

2)We implement a Role Based Access Control (RBAC) [11].

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

3)We achieve anonymous authentication is achieved by implementing a strong digital signature algorithm (SHA -1 hash function) where the attributes of users are hidden from cloud [13].

4)The access policies that are set by users are hidden from other users by implementing Query driven approach.

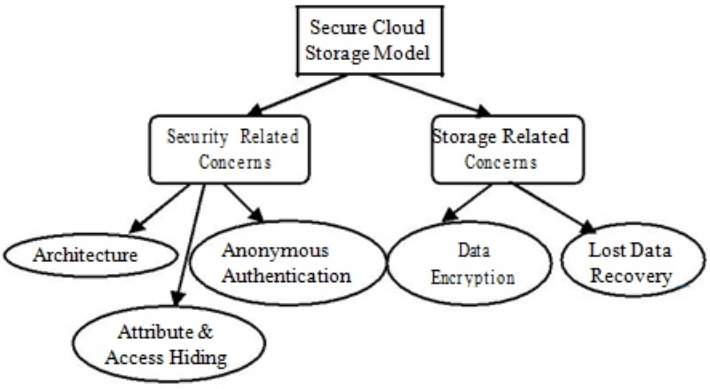

Fig.1. Overview of Proposed Work

3.1.2.Storage Related Improvements:

1)For secure storage of data, we implement a strong encryption and decryption technique.

2)We use Homomorphic encryption technique where Paillier public key cryptosystem is used.

3)We implement automatic data retrieval in which string matching algorithm is used for recovery of lost data.

3.2.PROPOSED SYSTEM ARCHITECTURE

The proposed architecture is a decentralized one where multiple numbers of KDCs are present for key distribution and management. These KDCs are geographically dispersed.In Fig.2 few users a scenario is presented with user 1 as owner of the file, user 2 as reader and user 3 as writer. These users are organized according to their roles based on their designation in the organization. If the user 1 wants to upload his file to the cloud he first needs to get registered to his corresponding KDCs. The output of this registration process is the generation of a unique user identifier for that user by the KDC. This user ID will be further used for all operations being performed by the user in the cloud. First level of authentication is achieved by a registration process where the users are identified as a legitimate.

Now the user needs to go in for a second level authentication, which is done by the trustee system. The trustee can be assumed as a trusted third party such as a government organization who uniquely identifies the users with some proof for instance, passport, vote id, driving license etc. This trustee system will generate a token for the user once he produces his unique Id to the trustee system. The generated token is further passed on to the corresponding KDC for generating the keys for encryption and decryption of the file that needs to be uploaded and/or downloaded.

For secure file storage, a Homomorphic encryption technique is adopted which implements an asymmetric key cryptography called Paillier Cryptosystem [13], [18]. This Cryptosystem is computationally strong and highly resistive to key-based attacks. It uses a series of complex mathematical functions for producing a single parameter. Now the files are encrypted using keys that are uniquely generated for this file according the access policy that has been defined by the owner of the file. The file is encrypted with keys that are generated by KDCs and also based on the access policy that is defined for that user by the owner of the file. Based on user authentication and claim policy, the files are encrypted and stored in the cloud. Before being encrypted, all files are encoded using Base64 encoder and a copy of the original encrypted file is stored in backup files.

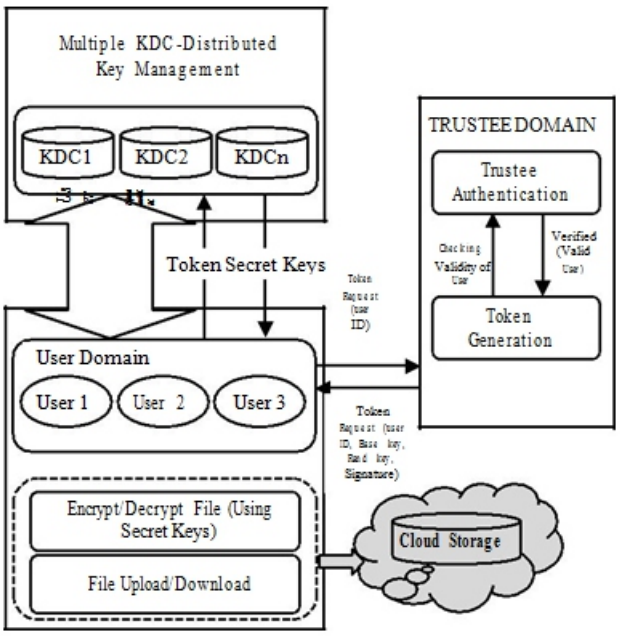

Fig.2. Overall View of Proposed Architecture

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

When some other user in cloud is interested in reading or writing the files, the access will be permitted based on the access policy defined for that particular user. Similar to the upload operation, the user sends a request for downloading the file from the cloud. He is authenticated and keys for decrypting the files are obtained using which the files are retrieved back. Before being downloaded, the cloud checks for the integrity of the file. If the file is found to be corrupted or changed, it will perform an automatic recovery operation. The file recovery process is carried out in two steps. First, the file's integrity is checked by using a string matching algorithm. Then, if found to be corrupted it is then recovered by file replacement of original file that is kept in backup.

3.3 FUNCTIONAL MODULES:

3.3.1.User Enrolment Access Control Techniques:

The users get registered to a particular KDCs by creating individual accounts by giving necessary details like user name, user id, password, email id and phone number. A successful registration is possible only if the user given details match with the KDC details. The KDC associated with this particular user will authenticate the user and allow for further user operation.Role Based Access Control (RBAC) model is implemented where users are classified based on individual roles. The RBAC can be adopted for implementing various important techniques like separation of duties, data abstraction and least privilege. The roles are defined by the system. The proposed work focuses on the RBAC technique, as anonymity of users is greatly preserved in this scheme.

3.3.2.Anonymous Authentication and Token Generation:

As shown in Fig.2, all users initially register with their KDCs with their own unique identity UID. The KDC draws at random, key KBASE G, where a generator g generates random group ofcyclic keys G. Let K0 = KBASE 1/a0. Now the token is generated typically as a combination of above parameters i.e. the user'sUID, key and user's signature. The output token is γ = (UID,KBASE, K0, ρ). Here ρ is the signature of the token generated using the authentication algorithm Digital Signature Standards withSecure Hash Algorithm (SHA-1). Hence the user details are hidden from cloud. In this way, an anonymous authentication is achieved.

3.3.3.Trustee and User Accessibility:

After the registration process, users can log into their individual accounts with their credentials. Once the user logs in, specific operations such as file upload, file download, listing of files, revocation list and key details associated with that user can be performed.Third party authentication is done by a trustee where the user needs to get a token from the trustee for carrying out further operations. The contents of the token are again a typical combinations of user's user id, key and user signature.

The hash function used for generating the user signature is Secure Hash Algorithm (SHA-1). Digital signatures are of great use in identifying the users uniquely and for checking the integrity of the content in case of tampering. In the proposed system the signature is generated by taking the user's UID as input and finally the signature is obtained in a condensed format called message digest. This is obtained as a result of applying the hash function, and is computationally difficult to interpret at any point of time.

3.3.4.File Encryption and Upload:

After the trustee's issuance of tokens to users, the users send their tokens to their respective KDC's for getting the keys for encryption and decryption of the files. The Paillier cryptosystem is implemented here for generating keys. The users now encrypt their files with the keys received keys. They also set their own access policies, i.e. privileges to the file.

The access policies set by individual users for their files are hidden from other users by implementing a query driven approach. The query-driven approach is an SQL coding written for the cloud database.Herein, the attributes and privileges of users are hidden from the cloud as their details are stored in encrypted format.

3.3.5.File Decryption and Download:

In this phase, the users can download the files from the cloud according to the access policies defined by the owners of the concerned files. The users satisfying access policy conditions can download a file and decrypt it using his private keys obtained from the corresponding KDC. This process is similar to file encryption and upload.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

3.3.6.File Recovery:

The files stored in cloud are prone to various attacks that may result in data loss or data corruption. In order to retrieve the exact file that was stored by the user, the cloud server invokes the file recovery function. All files that are stored in the cloud have a separate backup copy in the backend server. Before downloading, the cloud server initial checks the integrity. If the file is found to be corrupted, then it automatically invokes String matching algorithm function to compare the contents of corrupted file with original file copy. Finally it replaces the lost data in file and retrieves back the original content.

3.4 .PAILLIER CRYPTOSYSTEM:

The Paillier cryptosystem, which is a type of public key cryptography, enables high security with symmetric key data encryption. One of the striking features of the asymmetric algorithm is that the key used for encryption are different from the key used for decryption, so that users have separate sets of private and public keys. The public key is made available widely whereas the private key is kept secret. The files are encrypted using the pubic key but can be decrypted only using the corresponding private key. Even though the attributes of public and private keys are mathematically related but the private key cannot be derived from the public key.The Table.2 presents the notations used in the Paillier algorithm. The algorithms for the Pallier system are given in Fig.3.

Table.2. Notations used in Algorithm

Symbols	Computation
..	set of integers co - prime to ^{2}
..	set of integers co - prime to n
.	set of integers n

4. PROTOTYPE IMPLEMENTATION:

We developed a prototype model of proposed system and executed it as a cloud application by connecting 10 computer nodes using intranet. To host the developed application, we used the freeware eye OS private cloud application platform that provides web based desktop interface to run the system [19].

Since our model is based on RBAC access control method, it follows strictly designation based control and distinguished power. We developed an application for universities where the hierarchical roles of dean, secretary, principal, professors, assistant professors, lecturers and students have defined access rights according to their respective role.

A) Key Generation

1) Choose two large prime numbers p and q, such that gcd (pq, (p-1) (q-1)) = 1.
2) Compute n = pq, λ = lcm (p-1, q-1).
4) Calculate the following modular multiplicative inverse
5) = (L (g^{λ} mod n^2)$^{-1}$ mod n, where the function L is defined as L (u) = u – 1/n.
6) The public (encryption) key is (n, g).
7) The private (decryption) key is (λ,).

B) File Encryption

1) Let m be a message to be encrypted where
3) Compute cipher text as, =· od n^2

C) File Decryption

2) Compute message, = L(c^{λ} mod n^2) mod n

Fig.3. The Paillier Cryptosystem description

Registration and Access Rights definition:

Initially everyone registers with the KDC which is in the integrated server when manual verification is done at the First Level Verification. Individual colleges of the University maintain their own KDCs where users have defined access rights according to their role. Also, file owners have discretionary access control to the files that is uploaded by them. For example, suppose, the principal of the college wants to upload some file which he wants to get notified to all faculty members but not to the students - here he can set privilege for the file that he has uploaded by defining appropriate access rights to the users such as: read/write/download.

File Upload:

For uploading data, a user needs to log-in to his domain by giving his correct username and user-id and then perform operations like uploading, downloading or deleting files and revoking the rights previously defined by him etc.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

To upload the files the token need to be generated. The unique token is generated by trustee which is the third party verification domain. Only valid users will get the token. Here the Second Level Verification is done where the user details are verified by trustee by cross checking the details with the registered KDCs.

For generating token, the user-id is converted into a unique signature by using message digest using SHA-1 algorithm. This signature, a Random key and a Base Key forms the token contents. Now when the verified user passes the token to KDC, keys are generated and the file is encrypted. For encryption/decryption, we have implemented the Paillier algorithm. This provides the Third Level Verification.

Note that the privileges set by owner of the files are not viewed by the other users who do not have privilege to view. This system has advantage for Whistle Blowers who bring to the notice of higher authority of the organization without being their identity being revealed. At the same time the users mustauthenticated one. The rights that are previously defined by a user say principle can be dynamically reassigned by him/her later.

Data recovery:

All files that are uploaded to the cloud database are encoded using Base64 encoding algorithm and then encrypted. For data recovery, we have implemented the concept of data replication and file replacement with string matching algorithm. Before being stored in the cloud database, a copy of the encrypted file is stored in the backup database. While processing any file download request, the file is checked for correctness by comparing with its backup copy.

If any error is found, an automatic replacement of the lost data with original data takes place by running a string matching algorithm.One of the limitations of our existing prototype system is that currently, file size up to only 100 MB can be uploaded at a time. We are making changes to our scheme to increase the size of the file that is to be uploaded.

5. QUALITATIVE ANALYSIS:

Let us analyse the strength of the security mechanisms in our proposed system and its time-wise performance.

5.1 BREAKING OF SYSTEM SECURITY:

The system security can be analyzed based on the cryptographic algorithm that is used for encryption of data. In the proposed system a strong Homomorphic algorithm (Paillier) is implemented which uses very large prime numbers for generating keys.Even if the keys are hacked the contents cannot be decrypted, this is possible because of the Homomorphic additive or multiplicative property of the Paillier algorithm. Compared with the symmetric key encryption, the Paillier algorithm is strictly resilient to brute force attacks.

The algorithm has a distinguished characteristic of semantic security which protects the information from being extracted, even if the data on which computation is performed is known. The Paillier cryptosystem is secure as it is probabilistic polynomial time algorithm (PPTA), this property makes sure that no partial information is obtained even when part of cipher text for certain message and its length in given.Thus, the algorithm implemented in the system is strongly resilient against cryptanalysis attacks such as chosen cipher text and adaptive chosen cipher text attacks. This way a secure cloud storage model is developed, where the encrypted file remains secure against data losses.

5.2 PERFORMANCE ANALYSIS:

The efficiency of the system can be analyzed in terms of encryption and decryption time of the algorithm. We compare the performance of the system with a symmetric key encryption (3DES) system.In Table.4 shows the encryption and decryption time of different file size. From the Table.3 it is clear that the amount of encryption time taken by Paillier algorithm is almost half as compared to that of 3DES algorithm for the same input.Similarly the amount of decryption time is also half when compared to 3DES algorithm.

We can see that the proposed system is at par with performance when compared to symmetric key based system when files of different size were given as input and encrypted. The results show the encryption and decryption time is fast when compared with the symmetric algorithm based system.Thus the proposed system is fast and secure in terms of file recovery and file encryption. The time required to find the files been corrupted is also fast and the recovery of corrupted files back to original takes reduced amount of time.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

6. CONCLUSION AND FUTURE WORK:

In this paper, we addressed the security and storage issues simultaneously based on the type of architecture, access control methods and the authentication techniques. The key distribution is done in a distributed way by implementing multiple KDC structure. The users are anonymously authenticated and their attributes are hidden from the cloud by implementing digital signature algorithm.

The access policies associated with individual files are hidden from other users by implementing a Query based approach. Further, storage related security issues are enhanced by implementing a Homomorphic encryption technique to encrypting the outsourced data. Also, the cloud servers are prone to various types of attacks that can cause data loss or leakage. This issue is addressed by implementing a string matching algorithm that detects deviations and automatically retrieves the lost data using backed-up data.

The proposed scheme in the system uses strong authentication mechanism, where the users claim is validated at three levels. Initially, the users need to get themselves authenticated with KDC.For keys generation, a trusted third party verifies the user's credentials and gives back the secure token. Finally a message digest of the UID is generated using SHA-1 and the file to be uploaded is encrypted using Pallier cryptosystem. In this manner, a three way authentication is achieved.We are currently working towards extending our prototype to a larger-scale realization with more storage and distributed resources. We will verify its scalability in terms of performance and computation and communication overheads.

Table.4. Performance analysis of 3DES and Paillier algorithm with varied File size

Input File Size (KB)	Encryption Time		Decryption Time	
	Symmetric Key Algorithm (3DES)	Asymmetric Key Algorithm (Paillier)	Symmetric Key Algorithm (3DES)	Asymmetric Key Algorithm (Paillier)
3	3.54	2.26	3.53	2.24
5	5.76	2.35	5.81	2.33
7	8.31	1.77	8.21	1.73
11	13.78	7.99	13.65	7.87
16	22.19	8.65	21.89	8.64
21	28.28	19.06	27.99	17.99

REFERENCES:

[1]S. Ruj, M. Stojmenovic and A. Nayak, "DecentralizedAccess Control with Anonymous Authentication of Data Stored in Clouds", IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 2, pp. 556-563, 2013.

[2]J. Hur and Kun Noh, "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems", IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 7, pp. 1214-1221, 2010.

[3]C. Gentry, "A fully homomorphic encryption scheme", Ph.D., Dissertation, Stanford University, 2009.

[4]C. Wang, Q. Wang, K. Ren, N. Cao and W. Lou, "TowardSecure and Dependable Storage Services in CloudComputing", IEEE Transactions on Services Computing, Vol. 5, No. 2, pp. 220-232, 2012.

[5]G. Wang, Q. Liu and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services", ACM Conference on Computer and Communications Security, pp.735-737, 2010.

[6]W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and efficient access to outsourced data", ACM Cloud Computing Security Workshop (CCSW), pp. 55-66, 2009.

[7]M. Li, S. Yu, K. Ren, and W. Lou, "Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings",Security and Privacy in Communication Networks Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 50, pp. 89-106, 2010.

[8]F. Zhao, T. Nishide and K. Sakurai, "Realizing fine-grained and flexible access control to outsourced data with attribute-based cryptosystems", Information Security Practice and Experience Lecture Notes in Computer Science, Vol. 6672, pp. 83-97, 2011.

[9]Matt Bishop, "Computer Security: Arts and Science",Section 1.3.1 - Goals of Security, Addison-Wesley Professional, 2003.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[10]S. Ruj, A. Nayak and I. Stojmenovic, "DACC: Distributed access control in clouds", IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp. 91-98, 2011.

[11]Michael E. Whitman and Herbert J. Mattord, "Principles of Information Security", Cengage Learning, Fourth Edition, 2011.

[12]Kan Yang, Xiaohua Jia and Kui Ren, "DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems", IACR Cryptology ePrint Archive, pp. 419, 2012.

[13]William Stallings, "Cryptography and Network Security, Principles and Practice", Pearson Education, Fourth Edition, 2005.

[14]http://en.m.wikipedia.org/wiki/Mandatory_access_control

[15]www.cis.syr.edu/~wedu, Accessed on: 9 July 2014.

[16]http://en.m.wikipedia.org/wiki/Discretionary_access_contr ol, Accessed on: 13 July 2014.

[17]https://www.cs.cornell.edu, Accessed: 17 July 2014.

[18]www.cs.rit.edu, Accessed on: 20 July 2014.

[19]Eye OS Applications, www.eyeos-apps.org, Accessed on: 3 September 2014.