Deduplication and Decentralized Access Control in Cloud with Efficient Public Auditing Mechanism

Sohail A. Khan[1], Dr. Shafi K. Pathan[2]

[1,2]Savitribai Phule Pune University, Smt. Kashibai Navale College of Engineering, Pune, India

Abstract: ***Cloud computing is a very important area which Permits client to remotely store their data into the cloud and enjoy the on-interest excellent applications and services. However in cloud computing, since the data is put away anyplace over the globe, the client organization has less control over the stored data. So here, security and privacy are very important things. The user should to validate himself / herself before beginning any transaction, and user privacy is also important so that the cloud or other user in cloud do not know the identity of user who stored the data. This is an access control system for data store away in cloud that gives anonymous authentication. In this scheme, the cloud confirms the user without knowing the users identity before storing data in the cloud by using ABS (Attributes Based Signature). The proposed scheme uses ABE (Attribute-Based Encryption) in which the attributes are match and according to matching attribute the various access control (Read, Write) are provide to user. Also for Integrity checking trusted outsider security service provider is used who does not store any data at its end, and it's only confined to providing security service. Thus, enabling public auditability for cloud storage is of discriminating significance so that client can resort to a third-party auditor (TPA) to check the integrity of outsourced data and be effortless. In Proposed system the task of Key distribution is done in a decentralized way, for this more than one KDC are used who shares the same databases. The scheme not only verifies the integrity of data but also performs De-duplication of files. Moreover, this authentication and access control scheme is decentralized and robust, unlike other access control schemes designed for clouds which are centralized. The Experimental result show that the proposed system is more efficient in terms of data storage and security, and also it reduces the computation overhead of user by making the use of TPA.***

Keywords: Cloud Computing, Key Distribution Center (KDC), Attribute Based Encryption (ABE), Third Party Auditor (TPA), Integrity Checking, De-duplication checking, Access Control

1. Introduction

Cloud computing is the conveyance of registering services over the Internet. Cloud services allow individuals and organizations to use both software and hardware that are managed by third parties at remote locations. In Today's era because of the advances in network technology and an increase in the need for computing resources have prompted numerous organizations to outsource their storage and computing needs. This new economic and computing model is generally referred to as cloud computing and incorporates various sorts of services, for example, infrastructure as a service (IaaS), where a customer makes utilization of an service providers computing, storage infrastructure; platform as a service (PaaS), where a client influences the providers resources to run custom applications; lastly software as a service (SaaS), where clients use software that is run on the providers infrastructure.

It is important thing to protect the security of data and privacy of user. Cloud should guarantee that the user attempting to access data and services are authorized users. Authentication of user can be achieved utilizing different public key cryptographic techniques. User should guarantee that the cloud is not altering with their data and computational results. It is also be important to hide the user's identity for security reasons. For example, while putting away medical records, the cloud should not to have the capacity of getting the records of a specific patient. User should also guarantee that the cloud is ready to perform computations on the data without knowing the data values. One approach to hide the data from the cloud, but carry on computation on the data, is by the utilization of homomorphic encryption methods [7]. User sends messages encrypted using homomorphic encryption technique to cloud, while the cloud without knowing the actual data performs computations on these encrypted messages and gives back outcomes to the user.

Consider now the following situation. Patients store their personal medical records in the cloud. Different users can get access to various data fields. Here the same data fields may be accessed by a specific group of people which are authorized. For example the patient's medical history and drug organization can be accessed to by doctors and nurses, but not by hospital management staff.

In online social networking , generally owners are members of the networking site, they keep their personal details, music ,recordings, pictures, videos in the cloud and different individuals can view them depending upon their access rights. A member can post a message or transfer a photo whenever, which will be visible only to the friends and selected groups that she belongs to, but not available to the rest. It is important to also protect privacy of these data from the cloud. Giving access rights to some authorized users and preventing the other user from getting an access to that data, is called access control. One approach to achieve this is to put a list of all valid users in cloud who can access the data, this called as user based access control. In cloud computing, such records can be much long and frequently dynamic, which will make taking care of such records to a great degree troublesome. Every time the list must be verified whether the user is valid. This outcomes in a tremendous computation and storage costs. Another approach is to encrypt data is by using public keys of valid users, so that only they are able to decrypt data using their secret keys. However the same data then should be encrypted for few times individually for every user, which may bring about huge storage costs. Along these lines here it is helpful to utilize the cryptographic technique called Attribute Based Encryption (ABE) [2]. To achieve access control in clouds. Using ABE, data owners encrypt data with attributes that they possess and store the data in the cloud. The cloud is not able to read the stored data. Users are given attributes and secret keys by a key distribution center i.e. KDC. Those with matching set of attributes are able to decrypt the information. For example, in public health records repository [3], the medical records contain history of the patients and might be accessed either by medical professionals like doctors and nurses, researchers and academicians or management authorities such as insurance companies and government policy makers. Different people are allowed to access different records. Every user is given attributes such as the (Hospital names), designation (occupation and specialization) etc.

The Proposed System uses Third Party Auditor, which performs the integrity of data stored in the cloud on behalf of user request. So here, we are having public audit ability for cloud storage that users will resort to a third-party auditor (TPA) to ascertain the integrity of information. Here, this paper provides the varied problems associated with privacy whereas storing the user’s knowledge to the cloud storage throughout the TPA auditing. Also the data deduplication is perform by data owner which reduces the computation cost and storage space of the system. In computing, data deduplication is a specialized data compression/reduction technique for eliminating duplicate copies of repeating data. This technique is used to improve and increase storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. In the deduplication process, unique chunks of data, or byte patterns, are identified and stored during a process of analysis. As the analysis continues, other chunks are compared to the stored copy and whenever a match occurs, the redundant chunk of information replaced with a small reference that points to the stored chunk. Given that the same byte pattern may occur dozens, hundreds, or even thousands of times the amount of data that must be stored or transferred can be greatly reduced.

The paper is organized as; section 2 contains information about related work. Section 3 contains implementation details which includes system architecture, systems overview, mathematical model, algorithms and experimental setup. The section 4 contains results and discussion of the proposed work done so far. The last section 5 contains the conclusion of research work done. At the end various references are mentioned which are used in this paper.

2. Related Work

Here first think of some as existing plans. Fuzzy IBE [6] offers two interesting new applications. The first one is an Identity Based Encryption system that uses biometric identity. That is one can see a user's biometric, for instance an iris scan, as that user identity depicted by a few traits and after that encode to the client utilizing their biometric character. Subsequent to biometric estimations are boisterous, it is bad to utilize existing IBE system. Be that as it may, the mistake resistance property of Fuzzy IBE takes into consideration a private key which is gotten from an estimation of a biometric to unscramble a cipher text encoded with a somewhat diverse estimation of the same biometric. Also, Fuzzy IBE scheme can be utilized for an application that can be called "attribute based encryption". In this application a gathering will wish to scramble an archive to all clients that have a certain set of characteristics. For instance, in a computer science department, the chairperson might need to encode an archive to all system personnel on a contracting board of trustees. For this situation it would encode to the character "hiring committee”, "faculty", "systems". Any user who has a identify that contains these qualities could decrypt the archive. The point of interest to utilizing Fuzzy IBE is that the report can be put away on a straightforward untrusted capacity server as opposed to depending on trusted server to perform authentication checks before conveying a report. ABE was proposed by Sahai and Waters [6]. In ABE, a user has an arrangement of ascribes not withstanding its remarkable ID. There are two classes of ABEs. In key-strategy ABE or KP-ABE [2], the sender has an entrance strategy to encode information. An essayist whose properties and keys have been repudiated can't compose back stale data. The recipient gets traits and mystery keys from the characteristic power and can unscramble data in the event that it has coordinating traits. In Cipher content approach, CP-ABE, the recipient has the entrance arrangement as a tree, with attributes as leaves and monotonic access structure with AND, on the other hand and other threshold gates.

KP-ABE [2] is a crypto system for fine grained sharing of encoded information. In KP-ABE cipher text are mark with attribute and private key are connected with access structures that control which cipher text a user can decrypt. It is utilized for securing touchy information store away by outsiders on the web. In this system each encrypted cipher text is named by the encryptor with an arrangement of unmistakable qualities. Each private key is connected with an entrance structure that indicates which kind of cipher text the key can decode. Note this setting is reminiscent of mystery sharing plans. Utilizing known procedures one can manufacture a mystery sharing plan that indicates that an arrangement of gatherings must collaborate with a specific end goal to remake a mystery. For instance, one can indicate a tree access structure where the inside hubs comprise of AND as well as entryways and the leaves comprise of various gatherings. Any arrangement of gatherings that fulfill the tree can remake the mystery. In this development each user's key is connected with a tree-access structure where the leaves are connected with qualities. A user can unscramble a cipher text if the traits connected with a cipher text fulfill the key's entrance structure. The essential contrast between this setting and mystery sharing plans is that while mystery sharing plans take into consideration collaboration between various gatherings, in this setting, this is explicitly prohibited. Case in point, if Alice has the key connected with the entrance structure "X AND Y", what's more, Bob has the key connected with the entrance structure "Y What's more, Z", framework would not need them to have the capacity to unscramble a cipher text whose just trait is Y by conniving. To do this, this framework adjusts and sums up the strategies to manage more intricate settings. This cryptosystem gives an effective apparatus for encryption with fine-grained access control for applications, for example, sharing review log data.

CP-ABE [1] is an approach to obtain complex control on encrypted data. This technique is utilized to keep encoded information secret. In this system, a user's private key is related with a arbitrary number of attributes communicated as strings. Then again, when a gathering encodes a message in this framework, they indicate a related access structure over properties. A user just can unscramble a cipher text if that client's properties go through the cipher text's entrance structure. At a numerical level, access structures in this framework are portrayed by a monotonic "access tree", where hubs of the access structure are made out of limit entryways and the takes off portray characteristics. There AND entryways can be built as n of n limit entryways as well as doors as 1 of n edge entryways. Moreover, this plan can deal with more unpredictable access controls, for example, numeric extents by changing over them to little access trees.

Multi-Authority Attribute-Based Encryption [5] metho permits any polynomial number of free authorities to attributes and disperse secret keys. An encryptor can pick, for every power, a number dk and an set of attributes; he can then encode a message such that a user can just unscramble on the off chance that he has in any event dk of the given attributes from every power k. Pursues scheme [5] is skilled of handling of disjoint arrangements of qualities that are appropriated among different powers. In this plan, an encrypting party indicates an arrangement of attributes AC with the properties in AC being controlled by a few powers. Give Ak a chance to be the set of attributes controlled by power k. At that point the cipher text C related with the trait set AC must be decoded by those clients with an arrangement of traits Au for which the cardinality of the crossing point Au Ak AC surpasses the individual edge dk, for every power k. one of the principle challenges in executing such a multi-power trait based encryption plan is the anticipation of conspiracy assaults among user that acquire secrete key parts from various powers. Also, it is alluring that there be no correspondence between the singular powers. To defeat these troubles, Chase's plan depends on a trusted focal power. The subsequent plan is fit for enduring different tainted powers, in any case, the genuineness of the focal power stays of key significance since, by the narrowing from [2], and the trusted power has the capacity of decrypting each cipher text.

3. Implementation Details

Following Figure. 1 shows the proposed system architecture.

A. System Overview

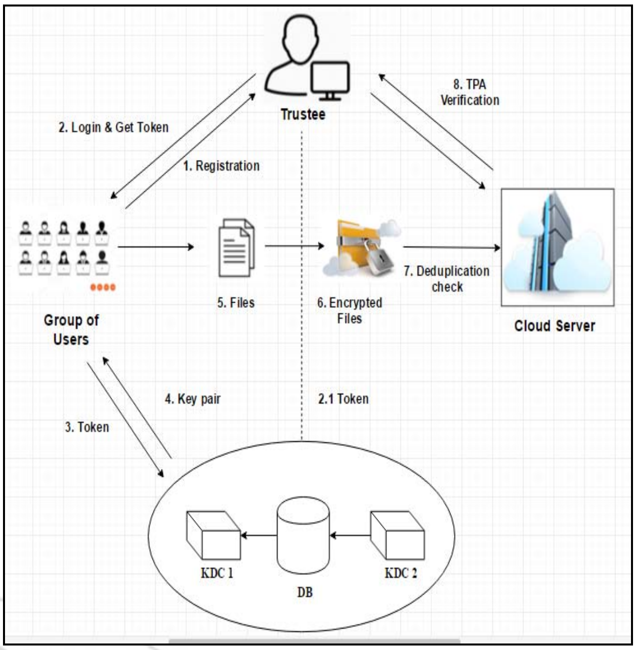

Figure 1: System Architecture

The proposed architecture is decentralized, meaning that there are several KDCs are used for key management. Creator, reader and writer be the different users in system. User receives a token from the trustee. A trustee can be someone like the federal government who manages user ID's etc. A user after validating the token from one or multiple KDC's, receives key pairs for encryption / decryption. The message is encrypted under the access policy. The access policy decides who can access the data stored in the cloud. After encrypting the file user generate the hash of file using SHA1 algorithm and Deduplication check is performed by matching the current hash with the previously uploaded files hash at cloud server. If current hash is matches with existing hash then user gets the link of file & there is no need to upload the file at cloud server as the same file is available at cloud server.

The creator decides on an access policy, to prove his/her authentication and signs the message using this claim. The cipher-text or Encrypted File is sent to cloud server while the hash of file is sent to TPA. TPA verifies the integrity of file on behalf of user request by proof generation and proof verify and result sent to the requested user. When a reader wants to read the file, then he/she request for file to cloud server, the cloud sends cipher-text or encrypted file. If the user has attributes matching with access policy, he/she can decrypt the ciphertext and get original message. Writing process is similar to file creation. When a reader wants to read data in the cloud, it tries to decrypt data in by using the secret keys which he receives from the KDCs. If user has sufficient attributes matching with the access policy, then he/she can decrypt the information stored in the cloud.

The main modules are:
1. Trustee: A trustee can be someone like the federal government who is responsible for managing social insurance numbers etc. On presenting unique id like health insurance or adhar card number, the trustee gives her/him a token.

2. KDC: The KDC is responsible for distribute secrete key and writer key to all authentic users. Cloud has multiple KDCs in different locations around the globe. If there is single KDC then it is centralize approach and if multiple KDCs then decentralize approach. KDC is a key distribution center which generates keys and assign the keys to the users, each organization or group of users have unique keys. KDC generates keys using key generation algorithm and random function. The proposed system uses decentralized access of KDC which are at different locations in the world. If one KDC is get failed then it automatically switches to another available KDC.

3. Client:
(A) Reader - Reader can perform only read operation on file. Reader reads the file online with help of secret key (SK).Reader perform the request to the KDC for the key. When the Reader enter the valid key only then the file is decrypted to the reader. The Decryption proceeds using algorithm ABE. Client is any user who want to read or write or modify the files which are stored on the cloud server.
(B) Writer - When the writer want to modify file then he is first authenticated using ABE. If the writer key is valid then he/she can update the file. To write the already existing file, User send its request to Cloud server, then cloud will send the encrypted file and ask for key (SK,PK) If key matches, then user is authenticated and allow to write.
(C) Creator - Authorized Creator can write the file and upload new files in the cloud. If any other user wants to read or modify the file of creator, he has to then send the request to the KDC to get access keys to the particular file. If KDC provide the key then only user is able to read, update or modify that file.

4. Cloud Server: Cloud server is used to storage of data where user can upload or stores the data. It also maintains the user database which is used for validating the user.

B. Algorithm

Algorithm 1: Deduplication check
Input: Encrypted Files EF, Set of previously generated hash HS
Output: Duplication Check result (Y/N)
Step 1: Input the encrypted file EF.
Step 2: Generate the hash of Encrypted File EF as HH using SHA1 Algorithm [8].
Step 3: Check the HH with HS.
Step 4: If HS contains HH then get the link of that file related to HH from the cloud server i.e Duplication Found (Y), then return Y;
Else
Step 5: Upload the EF and stored HH in HS and return N;
Step 6: End

Algorithm 2: ABE (Attribute Based Encryption)
It works under the following stages.
Step1: Setup
This is a randomized algorithm that takes no input other than the implicit security parameter. It outputs the public Parameters PK and a master key MK.

Step 2: Encryption
This is a randomized algorithm that takes as input a Message m, a set of attributes, and the public parameters PK. It outputs the cipher text E.

Step 3: Key Generation
This is a randomized algorithm that takes as input an access structure A, the master key MK and the public parameters PK. It outputs a decryption key D.

Step 4: Decryption
This algorithm takes as input the cipher text E that was encrypted under the set of attributes, the decryption key D for access control structure A and the public parameters PK. It outputs the message M if 2 A.

Algorithm 3: ECC Encryption
Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields.

1) Key Generation
Key generation is an important part where Alice have to generate both public key and private key. The sender will be encrypting the message with Bob's public key and the Bob will decrypt using its private key. Now, Alice have to select a number d within the range of 'n'. Using the following equation we can generate the public key.
Q = d * P
d = random number that we have selected within the range of (1 to n-1). P is the point on the curve. 'Q' is the public key and d is the private key.

2) Encryption
Let 'm' be the message that Alice is sending. Alice have to represent this message on the curve. This has in-depth implementation details. Consider 'm' has the point 'M' on the curve 'E'.
Randomly select 'k' from [1 - (n-1)].
Two cipher texts will be generated let it be C1 and C2.
C1 = k*P
C2 = M + k*Q
C1 and C2 will be send to Bob.

3) Decryption:
Bob wants the original message 'm' that is send to Alice,
M = C2 - d * C1
M is the original message that Bob wants to view.
How does we get back the message?
M = C2 - d * C1
'M' can be represented as 'C2 - d * C1'
C2 - d * C1 = (M + k * Q) - d *(k*P) (C2 = M + k * Q and C1 = k *P)
= M + k * d * P - d * k * P (canceling out k * d *P)
= M (Original Message)

C. Mathematical Model

System S is represented as:
S= {R, L, K, EC, E, D, En, Dn, T, C}

1. Registration Process
R= {R1, R2…Rn}

Where, R is the set of register users

2. Login Process
L= {L1, L2…Ln}
Where, L is the set of Login users in to the system

3. Keys

- K = { PK, SK}
  Where, K is set of keys in system.
- PK={ PK1,PK2,…, PKn}
  Where, PK represents set of public keys for group of users, PK1, PK2…, PKn represents public key.
- SK= {SK1, SK2,…,SKn}
  Where, SK represents set of secrete keys for group of users, SK1, SK2… SKn represents secrete key.

4. Elliptic Curve
EC represents EP (a,b) which is an Equation for an elliptic curve with coefficient a and c , and P represents random prime number.

5. For Encryption and Decryption
E = ECC (plain text, PKn)! Cipher text
D = ECC (cipher text, SKn)! Plain text
Where, ECC is Elliptic curve cryptography alternative to public key.

6. En be the set of encrypted text/files En = {e1, e2..., en} Where, e1, e2, e3... are the number of encrypted files / blocks.

7. De is set decrypted text/files Dn = {d1, d2, d3...dn} Where, d1, d2, d3... are the number of decrypted files / blocks.

8. T is a set for token generation T = {t1, t2, t3...tn} Where, t1, t2, t3... are the number of different tags.

9. C is cloud server for storage

D. Experimental Setup

The system is built using Java framework (version jdk 1.8) on Windows platform. The Net beans (version 8.0) is used as a development tool. The system doesn’t require any specific hardware to run; any standard machine is capable of running the application.

4. Results and Discussion

Figure 2. Shows data storage space comparison graph between existing system and proposed system.

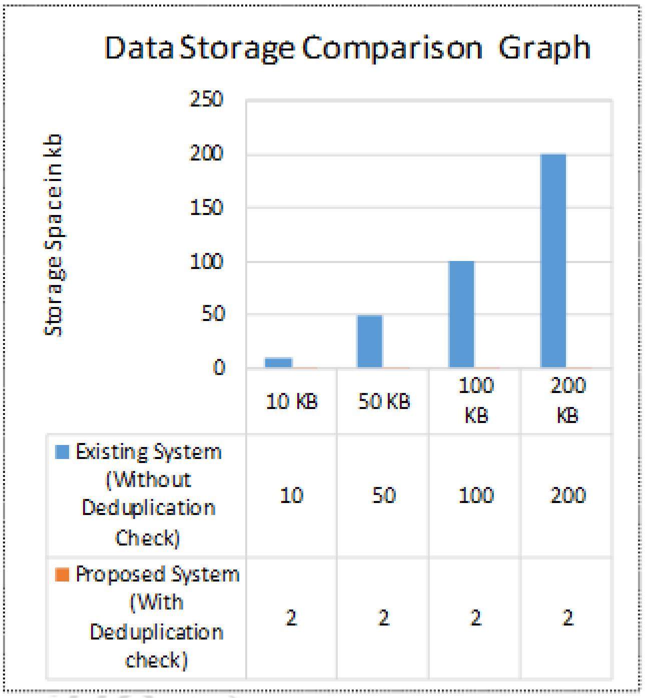

Figure 2: Data Storage Space Comparison Graph

The existing system does not performs the Deduplication check operation, which leads to uploading of same file at cloud server ,while proposed system checks for Deduplication of file at cloud server before uploading the file, which stores the server storage space and also reduce the computation cost of user. The X-axis shows various file size from 10 kb to 200 kb while Y-axis shows utilization of storage space by cloud server to store the duplicate data in kb.

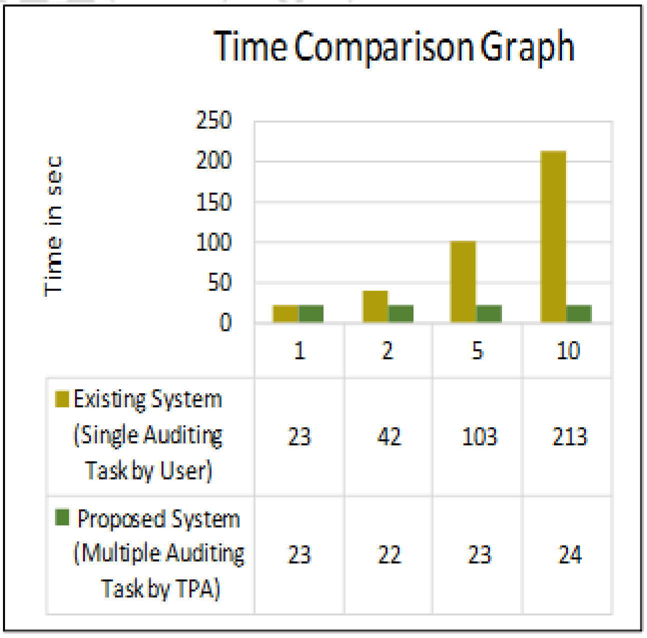

Figure 3: Time Comparison Graph

Figure.3 shows time comparison graph between existing system and proposed system. The existing system requires more time compare to propose system as existing system performs individual auditing task by individual user, while proposed system performs the batch auditing i.e. all users auditing task works simultaneously by TPA which reduces the time. The X-axis shows number of users with auditing task and Y-axis shows time required in seconds.

5. Conclusion

The decentralized access control technique with anonymous authentication in cloud, which provides de-duplication check and prevents replay attacks is proposed. The cloud does not know the identity of the user who stores information, but only verifies the user's credentials. Key distribution is done in a decentralized way. Also integrating checking is performed by Third Party Auditor on behalf of user request. As deduplication is performed before uploading the files at cloud which reduces the storage overhead and also reduces the computation cost. One limitation is that the cloud knows the access policy for each record stored in the cloud.

References

[1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext Policy Attribute Based Encryption," Proc. IEEE Symp. Security and Privacy, pp. 321334, 2007.R. Caves, Multinational Enterprise and Economic Analysis, Cambridge University Press, Cambridge, 1982. (book style)

[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute Based Encryption for Fine Grained Access Control of Encrypted Data," Proc. ACM Conf. Computer and Comm. Security, pp. 89□98, 2006.H.H. Crokell, "Specialization and International Competitiveness," in Managing the Multinational Subsidiary, H. Etemad and L. S, Sulude (eds.), Croom-Helm, London, 1986. (book chapter style)

[3] 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," Proc. Sixth Int'l ICST Conf. Security and Privacy in Comm. Networks (SecureComm), pp. 89□106, 2010.

[4] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed Access Control in Clouds," Proc. IEEE 10th Int'l Conf. Trust, Security and Privacy in Computing and Communications (TrustCom), 2011.

[5] M. Chase, "Multi-Authority Attribute Based Encryption," Proc. Fourth Conf. Theory of Cryptography (TCC), pp. 515□534, 2007.

[6] A. Sahai and B. Waters, "Fuzzy Identity Based Encryption," Proc. Ann. Intal Conf. Advances in Cryptology (EUROCRYPT), pp. 457473, 2005.

[7] C. Gentry, "A Fully Homomorphic Encryption Scheme," PhD dissertation, Stanford Univ., http://www.crypto.stanford.edu/ craig, 2009.

[8] Nalini C.Iyer and Sagarika Mandal,"Implementation of Secure Hash Algorithm-1 using FPGA" ISSN 0974 - 2239 Volume 3, Number 8 (2013), pp. 757-764 http://www. irphouse.com/ijict.htm

Author Profile

Sohail Ayyub Khan, received his B.E (Information Technology) degree from Dr. Babasaheb Ambedker Marathwada University Maharashtra India and is currently pursuing M.E (Computer Engineering) from Savitribai Phule Pune University, Maharashtra, India. His research interest include cryptography and cloud computing.

Pathan Mohd. Shafi is having more than 13 year of teaching experience and now currently working as a Asst. Prof. in Smt. Kashibai Navale College of Engineering, Pune for 7 years. He has worked as a lecturer in MIT Engineering College, Aurangabad for 7 years. Taught the subject like Computer Organization, Computer Graphics, Operating System, Network and Information Security, Information security and audit management, Java Programming Language. He has published four research paper in International Journal and eleven research paper in national conference.