A Symmetric Key Based Framework for Data Security in Cloud Computing

Mrinal Kanti Sarkar
Dept. of Computer Science & Engineering
University of Engineering & Management, Jaipur, India
mrinalkanti.sarkar@iemcal.com
Dr. V S Dhaka
Dept. of Computer Science & Engineering,
Jaipur National University, Jaipur, India
vijaypal.dhaka@gmail.com
Rupayan Das
Dept. of Computer Science & Engineering
University of Engineering & Management, Jaipur, India
rupayan.das@iemcal.com

---ABSTRACT---

Cloud computing offers the on demand computational infrastructure to the users which has the potential to decrease the huge cost to build IT based services. It can provide ubiquitous, convenient data storage facility. It is a significant issue as the whole data stored to a set of interconnected resource pools which are situated over different location of the world. Stored data can be accessed through virtual machines by unauthorized users. Today's major internet companies have built massive data centers and day by day this is increasing unbelievably. For this reason we are accessing various types of cloud flavor in terms of superb applications or services. But there is also a dark side of cloud, as insecurity creates major problem for cloud users. We know that security is an important and highly challenging issue in cloud computing. Now a day's data security can be provided in terms of some traditional concept like Cryptography, Steganography etc. The concept of Cryptography is much better than Steganography as space complexity is concern in reality. Here in this paper we approach cryptographic concept like Symmetric Key Based Encryption (SKBE) technique to encrypt data in encoded form and store it to the 3rd party cloud service provider. With help SKBE concept we also approach secure cloud model.

Keywords- cloud computing; data storage security; encryption

I. INTRODUCTION

Cloud computing is known as a model of latest technology over the internet which satisfy on demand services such as storage, software, resources and network [1]. It is growing rapidly and releasing the new services which required very less management of effort. Users can get the advantage of these various computing services without building its own infrastructure. In fact users can access cloud facility from any computer and from any location of the world. It has the ability to monitor the performance when it is allocate or reallocate the resources dynamically

Though there are many services that can be provided to the client by the cloud but data store is one of the main features that the cloud service providers provides to the users. But many clients/users are not ready to implement cloud computing model due to the lack of appropriate security mechanism or weakness in protection of data. There are so many cloud computing vendors, such as Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Services (S3), Using Amazon S3 [2], you can store and retrieve huge amount of data from anywhere of the world without any restriction of time and it is just a simple web services interface[2]. This services allow the developer to access the highly secure, scalable, fast, inexpensive reliable, infrastructure.

Data security is an important aspect regarding quality of services. But it is facing various types of security threat for number of reason. So, it is unable to fulfill the quality of services for the several reasons. Firstly we cannot implement the traditional cryptographic technology for the aim of data security as the user' loss their control on data storage. As we don't have the required knowledge of entire data, it is very tough job to verify the actual data using verification strategy. So, we can't implement a verification strategy. This leads to verify correctness of data which are stored in cloud storage and it face more challenging interface. Secondly, it is not a third-party data warehouse where the data will be stored. The data which are stored in cloud storage may be repeatedly modified by the user. So, for this frequent operation, it needs to more advanced technology to avoid data loss from the cloud storage. Last but not the least, the cloud data storage are running in a cooperated, simultaneously, and in distributed manner [3] and data of every client are stored in multiple physical locations in a random manner. So, we need a robust and protectable data storage technique in the real word which satisfies the distributed protocols for storage correctness and assurance.

In this paper, we have proposed a cloud security model based on SKBE (Symmetric Key Based Encryption).

Our contributions are summarized on the following aspects:

1) We propose a secret key based data encryption technique to hide the original data file in encoded form and successfully store it in to the 3rd party database.
2) It prevents data access from unauthorized users from cloud storage.
3) Our work incorporates efficient data storage and retrieval operation.

The paper is organized as follows: we briefly discuss the architecture of cloud computing and its security issues in Section II. Section III, we provides our system architecture, design goal, notations and security model. Cryptographic approach is presented in Section IV. We analyzed the security and performance evaluations in Section V. Section VI provide the related work. Finally, in Section VII, we conclude our remarks on our proposed model and its future scope.

II. CLOUD COMPUTING ARCHITECTURE AND SECURITY ISSUES

Service delivery models and the deployment models are two categories of Cloud computing services [1]. The deployment models are: 1) **Private cloud**: which can be used for single organizations, 2) **Private cloud**: which is provisioned for exclusive use by a particular community of customer, 3) **Public cloud**: is available to public user and they can register and use the available infrastructure, and 4) **Hybrid cloud**: it is a composition of private cloud and public cloud.

Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) are three layers of service delivery model [1].

Software as a Service (SaaS) offers services on demand. It is the topmost layer which provides a complete set of applications. It delivers application for end users and need not require installing application software on the customer's computer.

Platform as a Service (PaaS) provides platform oriented services for software execution. It is the middle layer and it delivers platforms, tools and other business oriented services that enable customer to develop and manage their own application, without installing any of the required platforms.

Infrastructure as a Service (IaaS) supports the basic infrastructure. It is the lowest layer and shares the hardware resources for executing services. It has the right for processing storage, networks etc

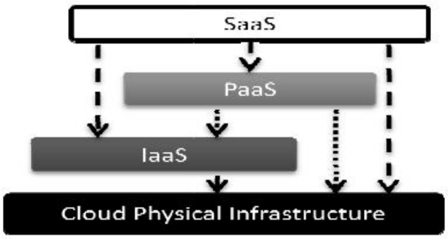

Fig-1: Cloud service delivery model

Fig-1 shows the service delivery model and it has several possible implementations which increase the complexity of the development of standard security model

As we are storing our data and running our software on somebody's CPU with the help of someone else's hard disk So it will face malicious security issues such as phishing, data loss, botnet (Collection of machines are running remotely). Moreover, the multi-tenancy and pooled computing resources have introduced different types of security challenges that need novel techniques to tackle. For example, hackers may use cloud to organize botnet because cloud often offers more reliable infrastructure at a relatively low price for them to start an attack [5].

III. PROBLEM STATEMENT

It is a crucial job to manage data-at-rest plays in cloud computing. The main problem with data-at-rest in cloud is loss of control. An unauthorized user can access the data as data are stored in a shared environment. Though, now-a-days storage devices are empowered by encryption techniques which restrict unauthorized access to data. Encryption methodologies fails to provide authorized access if encryption and decryption keys are available to malicious users. We are providing our system architecture.

A. *Schematic System Architecture*

The architecture of our proposed model [6] is illustrated in Fig-2. In this architecture, we define different network entities which can be identified below.

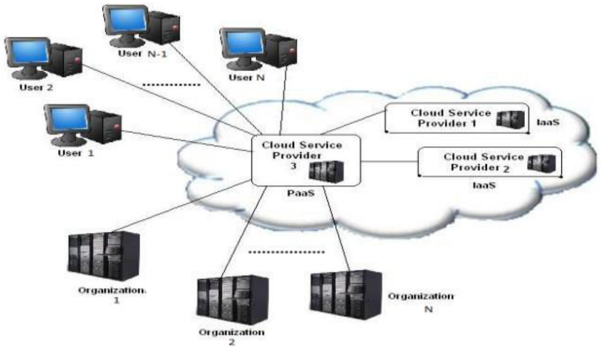

Fig-2: Schematic System Architecture for Cloud

- **User:** Users who want to use cloud infrastructure.
- **Cloud Service Provider-1(CSP-1):** The data will be stored here in the form of encoded file.
- **Cloud Service Provider-2(CSP-2):** Key generation, Encryption and decryption techniques will be stored.

This mechanism will encrypt file and retrieve file from encoded file.

- **Cloud Service Provider 3(CSP-3):** All the computations will be taken here by the user. CSP-3 will interact with CSP-1 and CSP-2.

B. Security Model

We are storing the encoded data file physically in order to maintain the privacy of user data and this concept is known as cryptography which tells that hide one piece of data/message in such a way that no one except the sender and intended recipient suspects the existence of the data. This is the new paradigm of security through obscurity. For example we divide the entire cloud security model into three parts.

We present the security model in Fig-3, Fig-4 and Fig-5. The computations done by the users will take place in CSP-3. If a user wants to store their data, the following process will happen:

1. CSP-3 requests CSP-1 for an encoded file.
2. CSP-1 will give acknowledge to CSP-3 by sending encoded file.
3. CSP-3 requests CSP-2 for the secret Key and data decryption algorithm stored in CSP-2.
4. CSP-2 will send secret key and data decryption algorithm to CSP-3 after getting request from CSP-3.
5. CSP-3 applies the decryption technique using secret key for decoding the encoded file and file will be saved into the temporary file in (F_{temp}) CSP-3.
6. CSP-3 again request CSP-2 for secret key and encryption algorithm.
7. CSP-2 sends secret key and encryption algorithm to CSP-3
8. CSP-3 applies the encryption technique with the help of secret key for encoding the decoded file (present in temporary file)
9. CSP-3 finally stores the encoded file to the database of CSP-1

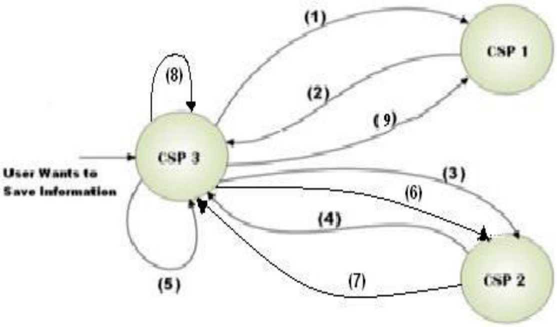

Fig-3: Encoding Previously stored data

1. CSP-3 gets original file direct from user and store it to the temporary file in CSP-3.
2. CSP-3 requests CSP-2 for secret key and encryption algorithm.
3. CSP-2 will give acknowledge to CSP-3 by sending secret key and encryption algorithm stored in CSP-2.
4. CSP-3 applies the encryption technique with the help of secret key for encoding the original file (present in temporary file)
5. CSP-3 stores the encoded file into the database containing encoded file in third party cloud service provider (CSP-1)

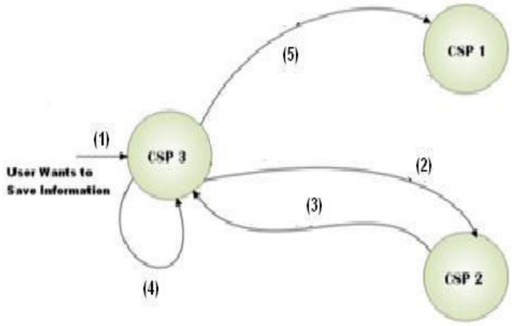

Fig-4: Encoding User Data

The following operation will be done whenever the user want to retrieve the data.

1. CSP-3 requests CSP-1 for encoded file.
2. CSP-1 will give acknowledge to CSP-3 by sending the required encoded file.
3. CSP-3 requests CSP-2 for the Decryption Algorithm stored in CSP-2.
4. CSP-2 will send data retrieval algorithm to CSP-3 after getting request from CSP-3.
5. Now the CSP-3 applies the retrieval algorithm on the Encoded file and those retrieved data will be stored into a file. This file will be displayed to the user.

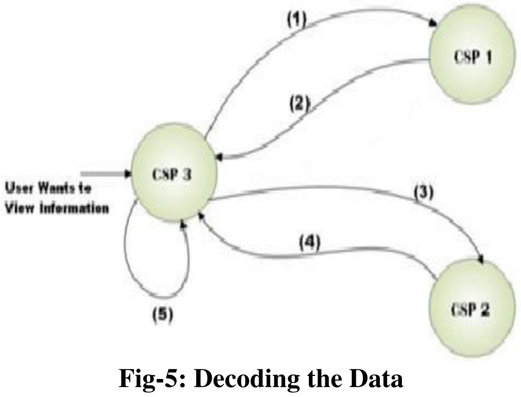

Fig-5: Decoding the Data

The temporary file will be deleted when the user want to log out from the system.

C. Design Goal

Encrypt confidential data into another form is called cryptography. In our proposed model we have taken Plaintext which is an original file.

- Correctness: Data of file will be Encode correctly and stored into the database.
- Availability: The data can be retrieved from encoded data file by an authorized user when it is required.

- Protection: Very hard to detect the secret key for decrypt the encoded data file.

D. *Notation and Preliminaries*

- **F_{en}:** This is a text file which contents the encoded data stored in CSP-1and we have to decrypt this file in order to get decoded file.
- **F_{temp}:** It is a temporary file which will be used during data processing time (i.e. retrieval operation and will be free). This file sometimes used as decoded file as well as user defined file. Any update (text Delete, Update & Addition) can be done on F_{temp}.
- **F_s:** It is a Secret key file generated by Key_Gen() method stored in **CSP**-2.
- **Key_Gen():** It is a Key generation function which uses a secure key generation algorithm.
- **Encrypt():** It is a encrypt function which uses a secure encryption algorithm to encode file.
- **Decrypt();** It is a decrypt function which uses a secure decryption algorithm to decode file.

IV. APPROACH TO ENSURE DATA STORAGE SECURITY

In cloud computing, we cannot process the data locally. The data will process remotely, so the security of data must be guaranteed and distributed server is needed. The major concern regarding security of data in cloud computing is that, as data are available in remote servers in raw format. So, it can be easily accessible and can be manipulate by unauthorized users. So, our main aim is to ensure data security in cloud storage such that it can't be detect by any malicious users. We have implemented the following methodologies.

A. *Encoded and Decoded File*

In this section deals with the Key generation algorithm and also deals with the Encoded and Decoded files which are to be stored in cloud data storage and Deliver toe the user respectively. The Encryption algorithm encodes the User given Plaintext in to Encoded File and Decryption Algorithm Decodes the Encoded file in to Original File or Plaintext.

Algorithm 1: Key_Gen()

This algorithm describes the procedures to generate secret key to encrypt data as well as decrypt data.

```
1. procedure
2. P & Q be two successive small prime numbers where ( $\gcd(P,Q)=1$ )
3. Number of prime number sets $m=(P+1)(Q+1)$
    4. Let F_i be the sum of two successive prime numbers
          (P_i,Q_i) (starting from smallest prime number)
          $F_i=(P_i+Q_i), P_i \neq Q_i$ where i =1 to m;
5. Sum of all prime number sets up to m $N_{sop}=\sum_{i=1}^{m} F_i$
   where N_sop is a positive Integer
6. Now average value of sum of all prime set
   $N_{avg\_sop}=\frac{\sum_{i=1}^{m} N_{sop}}{m}$ Where N_avg_sop is a positive Integer
7. Find a smallest random value R. Where
   $R \in Z^{+}_{N_{avg\_sop}}\{R \mid 1<R \le N_{avg\_sop}, \gcd(R,N_{avg\_sop})=1\}$
8. Let the cloud has K number of users. Users sets
   $\{U_1,U_2,U_3,\ldots,U_K\}$ Where $K \ge 1$
9. Select two small odd numbers B_i ,B_j from
   $F_1<B_i<N_{avg\_sop}, F_1<B_j<N_{avg\_sop}$ respectively
   where $\gcd(B_i,B_j)=1$
10. $\phi(B)=B_i+B_j$
11. Secret key $F_s=R^{K(\phi(B) \bmod N_{sop})} \bmod N_{sop}$
12. Return F_s
13. end procedure
```

Algorithm 2: Encrypt ()

This algorithm deals the pre-requisite requirements like Secret Key File (F_s) and Plain Text file (F_{temp}). With the help of S the user defined plaintext P converted in to Cipher Text file (F_{en}) or Encoded File that is stored in to 3rd party server (CSP-1)

```
1. procedure
2. String F_temp,F_en;
3. F_s=Key_Gen();
4. F_en=(F_temp)+F_s;
5. Store (F_en);
6. end procedure.
```

Algorithm 3: Decrypt ()

This algorithm deals the pre-requisite requirements like Secret Key file (F_s) and Cipher Text file (F_{en}).With the help of S the Cipher text or Encoded File (F_{en}) is Decoded in to Plain Text file (F_{temp}) or Decoded File that is delivered to the user.

Algorithm 4:

```
1. procedure
2. String F_en,F_temp
3. F_s=Key_Gen();
4. F_temp =( F_en)-F_s;
5. Send(F_temp);
6. end procedure.
```

V. SEQUIRITY ANALISIS AND PERFORMANCE EVALUATION

The security and the efficiency of the proposed model based on SKBE depend on our proposed system architecture and its security model which is defined in section II. We evaluate the performance of our model to encode the data file and store it in to the database.

A. *Security Strength against SKBE*

SKBE which is also known as Symmetric Key Based Encryption technique is a data encoding scheme where the original data file or plaintext is encoded in to another form (Cipher text). This encoded file is completely different form the original data file. Here the propose scheme is based on symmetric key cryptography and that is why secret key is used to encode the original data file in the sender side and similarly on the other hand same secret key is used in the receiver end to decrypt the data file. The secret key generation algorithm which is proposed in this paper is very complex and similarly encoding technique also gets tough due to the secret key. Using key generation algorithm every time we can get unique secret key and large sets of secret key can be obtained from that. Uniqueness of secret key creates problem for the attacker to decode the encoded data file.

B. *Security Strength against CSP-1*

CSP-1 only stores some files. These file contain the encoded data. It also has the set of encoded file previously stored. But an unauthorized user cannot get anything to from the encoded data. CSP-1 does not contain the retrieving algorithm, thus the encoded data are purely safe.

C. *Security Strength against CSP-2*

Secret key generation, Encryption and Decryption mechanism are stored in CSP-2, These will be needed at the time of viewing and encoding the data from CSP-3. CSP-2 does not contain the encoded data, thus knowing only the algorithm will not help the attacker.

D. *Security Strength against CSP-3*

In our proposed model, CSP-3 is responsible for computation i.e. encoded data into another form and retrieve the data from encoded form. All the files will be deleted after the above operations. So, there is nothing to fear from data loss and unauthorized access.

VI. Related Work

Cong Wang et al. [3] use homomorphic token with distributed verification of erasure-coded data. It ensures data storage security and finds the location the server which has been attacked. It support update, delete and append operation on data blocks such. But it is fail to achieve public verifiability and storage correctness. Shacham et al. [7] build a random linear function using homomorphic authenticator which is useful for unlimited number of queries without taking of communication overhead. Jules et al [8] proposed Proofs of Retrievability (POR) for large files. Later Bowers et al. [9] improved POR protocols which generalize both Juels and Shacham's work. In their subsequent work, Bowers et al. [10] used distributed systems to extend the POR model.

Shantanu pal et al. [11] ensures to find location of adversary or the attacking party from its target. It is ensuring a more secure platform for the other virtual machine. It may try to attack them, if adversary knows the location of the other VMs. This may harm the other VMs in between. Flavio Lombardi et al. [12] proposed to check the behavior of cloud resources. Executable system file can be monitored by logging and periodic checking. But it encountered faces system performance. Shah et al. [13] proposed to keep the record of online storage a TPA. It encrypts the data using symmetric-keyed hashes, and then it will be send to the auditor computed. However, this approach is applicable on encrypted files and long-term state is maintained by auditors. Schwarz et al. [14] used file integrity across multiple distributed servers to ensure security using erasure-coding and block-level file integrity checks.

Ateniese et al. [15] proposed the "provable data possession" (PDP) model to ensure possession of file in untrusted storages. This scheme used public key based homomorphic tags to audit the data file and it is providing public verifiability. In their subsequent work, Ateniese et al. [16] proposed a PDP scheme that uses only symmetric key cryptography. This method has only lower-overhead than their previous works. The modification (i.e block updates, deletions and appends to the stored file) of stored file can be done. But this scheme can be used on single server and it is not able to rectify small data corruptions. Distributed scenario and data error recovery issue is unexplored. Curtmola et al. [17] proposed to ensure data possession multiple across the distributed system. It extended PDP which ensure data possession on multiple replicas without encoding each replica separately. It guarantees that multiple copies of data are actually maintained.

VII. Conclusion

The architecture of symmetric key based framework for data security in cloud computing presented in this paper is currently in the prototyping stage. The main aim of this paper is to present an overview of security challenges that are happening in cloud data storage and introduce a new combined encrypting technique using secret key to prevent unauthorized data access in cloud data storage .And to detect intruders within the cloud environment. Through detailed security and performance analysis we show that our approach to find a secret key gives high security of data when it is on rest in the data center of any Cloud Service Provider (CSP).This proposed architecture will be able to provide customer satisfaction to a great level and it will attract more clients in the field of cloud computing for industrial as well as future research firms.

References

[1] Peter Mell, Timothy Grance, "The NIST Definatin of Cloud Computing", Jan, 2011.http://docs.ismgcorp.com/files/external/Draft-SP-800-145_cloud-definition.pdf.

[2] Amazon.com, "Amazon Web Services (AWS)", Online at hppt://aws.amazon.com, 2008.

[3] Con Wang, Qian Wang, Kui Ren, and Wenjng Lou, "Ensuring Data Storage Security in Cloud Computing", 17th International workshop on Quality

of service, USA, pp1-9, 2009, IBSN:978-42443875-4.

[4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Third Edition, Prentice Hall of India, 2010.

[5] B.P Rimal, Choi Eunmi,I.Lumb, "A Taxonomy and Survey of Cloud Computing Sytem", Intl. Joint Conference on INC, IMS and IDC, 2009,pp.44-51, Seoul,Aug, 2009. DOI: 10.1109/ NCM.2009.218.

[6] M. K Sarkar and T. Chatterjee, "Enhancing Data Storage Security in Cloud Computing Through Steganography", ACEEE International Journal on Network Security, ISSN: 2152-5064, Vol. 5, Issue. 1, pp: 13-19, Jan 2014.

[7] H. Shacham and B. Waters, "Compact Proofs of Retrievability", Proc. of Asiacrypt '08, Dec. 2008.

[8] A. Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files", Proc. of CCS '07, pp. 584–597, 2007.

[9] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage", Cryptology ePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/.

[10] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage", Cryptology ePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/.

[11] Shantanu Pal, Sunirmal Khatua, Nabendu Chaki, Sugata Sanyal, "A New Trusted and Collaborative Agent Based Approach for Ensuring Cloud Security", Annals of Faculty Engineering Hunedoara International Journal of Engineering (Archived copy), scheduled for publication in vol. 10, issue 1, January 2012. ISSN: 1584-2665.

[12] Flavio Lombardi, Roberto Di Pietro, "Secure Virtualization for Cloud Computing ", Journal of Network and Computer Application, vol. 34, issue 4, pp 1113-1122, July 2011, Academic Press td London, UK.

[13] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest", Proc. 11th USENIX Workshop on Hot Topics in Operating Systems (HOTOS '07), pp. 1–6, 2007.

[14] S. J. Schwarz and E. L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage", Proc. of ICDCS '06, pp. 12–12, 2006.

[15] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores", Proc. ofCCS '07, pp. 598–609, 2007.

[16] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession", Proc. of SecureComm '08, pp. 1–10, 2008.

[17] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-Replica Provable Data Possession", Proc. of ICDCS '08, pp. 411–420, 2008.

Mrinal Kanti Sarkar received his B.Tech degree in Computer Science & Engineering from Govt. College of Engineering & Ceramic Technology. M.Tech degree in Computer Science & Engineering from Indian Institute of Technology, Kharagpur. Currently he is doing his PhD. from Jaipur National University Jaipur. He served as a Senior Lecturer in The ICFAI University Tripura from 2008 to 2012. Now, he is working as Assistant Professor in University of Engineering and Management, Jaipur, India. His current research interest includes Security in Cloud Computing, Distributed System and Information Security.

Rupayan Das received the B.Tech degree in Information Technology from Academy of Technology. M.Tech degree in Information Technology from Institute of Engineering & Management, Salt Lake. He served as a Lecturer Trainee in Institute of Engineering & Management from 2012-2013.Now, he is working as Assistant Professor in University of Engineering and Management, Jaipur. His current research interest includes Security in Cloud Computing, Distributed System and Information Security.

Dr. V.S. Dhaka received his M.Tech. and Ph.D. in Computer Science from Dr. B.R. Ambedkar University, Agra, India. Currently he is working as a Professor in Department of Computer Science & Engineering, Jaipur National National University, Jaipur, India. He is is also Head, Dept. of Computer Science & Engineering, JNU Jaipur.He has 11 years of experience in the industry and academics. He has more than 32 publications in international journals. He always strives to achieve academic excellence. His current research interest includes Neural Network, Information Security, and Image Processing.