International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 2, April 2016, pp. 770~777
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i2.8919

HASBE Access Control Model with Secure Key Distribution and Efficient Domain Hierarchy for Cloud Computing

Rajanikanth Aluvalu*, Vanraj Kamliya, Laksmi Muddana*****
* Department of Computer Science & Engineering, GITAM University, India
** Department of computer Science, Glorious College of Computer Science, India
*** Department of Information Technology, GITAM University, India

Article Info

Article history:

Received Aug 28, 2015
Revised Dec 25, 2015
Accepted Jan 9, 2016

Keyword:

Access control
Attribute-set
Data Security
Decryption
Encryption

ABSTRACT

Cloud computing refers to the utility computing model, where virtualized resources are provided on demand over internet. It is a distributed commodity system and provides access to authorized users. Cloud computing virtualizes system by pooling resources from commodity hardware and supports multi tenancy. Cloud consists of user's confidential data. Cloud computing should ensure security for user data on cloud by providing fine grained access control. Traditional access control models are not sufficient to cater applications running on cloud due to its dynamic nature.Various access control models are proposed for cloud computing using attribute based encryption (ABE). All the proposed models suffered from various drawbacks. Among the proposed models HASBE proved as best in terms of flexibility, scalability and fine-grained access control. Howevere HASBE fails in supporting hierarchical domain structure. In this paper, we had proposed improved "Hierarchical attribute-set-based encryption" (HASBE) access control with a hierarchical assembly of roles with respect to their attribute values in flexible domain hierarchy structure and with predefined Secure key distribution policy.

Corresponding Author:

Rajani Kanth Aluvalu,
Research Scholar,Department of Computer Science & Engineering,
School of Technology, GITAM University, Hyderabad, India
Email: rajanik.rkcet@gmail.com

1. INTRODUCTION

Cloud computing has promptly become an extensively accepted model for providing services on demand over the internet. Cloud computing supports multi tenancy and cloud user looses data ownership once stored on cloud. Cloud service provider has to ensure reliability and security of user's confidential data. Huge amount of such data is stored on the cloud. To ensure security of the stored data, Access control models are widely used. [1]. Organizatios are generating huge data from their day to day transactions. Hugedata storge and maintainance is burden to organizations. They can store their data on cloud using services like Amazon s3, which provides storage service on cloud. Cloud based storage services support multi tenancy and users loose their data ownership once stored on cloud. Data security on such multi tenant storage services can be provided by using access control models. Access control means restricting access to resource, node etc. The act of restriction means, approval to access a resource is required, we call it as authorization. The traditional access control models like DAC ("Discretionary Access Control"), MAC ("Mandatory Access Control"), RBAC ("Role based access control") and ABAC ("Attribute Based Access Control") are not sufficient for required security levels. Later various attribute based encryption schemes are proposed [2]. Attribute based encryption (ABE) models are proposed by Sahai and Waters in 2005 [3] [4]. ABE allows users to encrypt and decrypt data using their attributes. User's secret key and cipher-text are adjunct on attributes. User can decrypt cipher text only if the set of attributes of user key are equivalent to the attributes

of the cipher text. In "attribute based encryption" (ABE) scheme the data owner has to use user's PK ("Public Key") to encrypt data. In later time various ABE based access control schemes have been proposed to overcome using user's public key for encrypting data.

Key Policy Attribute Based Encryption (KP-ABE): KP-ABE was developed by Goyal et al in 2006 [2] which is an enhanced model of ABE. In KP-ABE cipher text is associated with a set of attributes and user's decryption key is associated with a monotonic tree access structure [2]. In this model user can decrypt cipher text, only when the attributes associated with the cipher texts satisfy the tree access structure.

Cipher text Policy Attribute Based Encryption (CP-ABE): This is an alternative model of ABE invented by Sahai [3]. By using CP-ABE we can store data in encrypted form on untrusted server and maintain data confidentiality [5] .Data owner encrypts data and stores it on cloud with associated acess structure over attributes. To decrypt the cipher text, the data consumer's attributes has to qualify the cipher-text's access structure.

Hierarchical Attribute Set-Based Encryption (HASBE): model was invented by Wang et al [6]. It is combination of ("HIBE") and "CP-ABE".HASBE model has the hierarchical structure of users. HASBE structure contains a root master at the top, followed by multiple domain masters. Each domain master will have set of users and each user contains set of attributes [7]. To protect sensitive data from rivalaries, the data is stroed in encrypted form on servers, while the decryption keys are disclosed to authorized users only. HASBE scheme also suffering from various short comings, however an efficient key management mechanism is required to distribute decryption keys to authorised users, which is very difficult. This approach lacks scalability and flexibility in terms of user set and domain levels; as the numeral of legal users becomes large, the solution will not be efficient and performance degrades. The data owners are required to be online all the time so as to encrypt or re-encrypt data and distribute Keys to authorize users [8]. In this paper we had Extended HASBE ("Hierarchical Attribute Set-Based Encryption") by proposing secure key distribution with improved domain hierarchy of user roles to access the files stored on cloud [9]. Role based hierarchy helped us in overcoming the problem of data owner and data consumer (problem: data owner must be always online for key distribution). The root authority called as trusted authority will be always online and will distribute keys to the authorized users satisfying the policy in a more secure way. The data owner will store the data on the cloud and will share the access policy with trusted authority. Trusted authority will distribute keys based on data owner's policy that are predefined. For example whenever an authorized user gets logged on to the system and request for file access, trusted authority verifies the requested users attributes captured at the time of user registration with data owners access policy. If requested users attributes qualifies user's access policy the trusted authority will provide access rights by providing specific key for specific file. Here we are combining the HASBE with Role based Access control model [10] [11] [12]. This helps us to reduce the system time and make easy record fetch facility in more flexible manner improving overall performance.

2. EXISTING SYSTEM

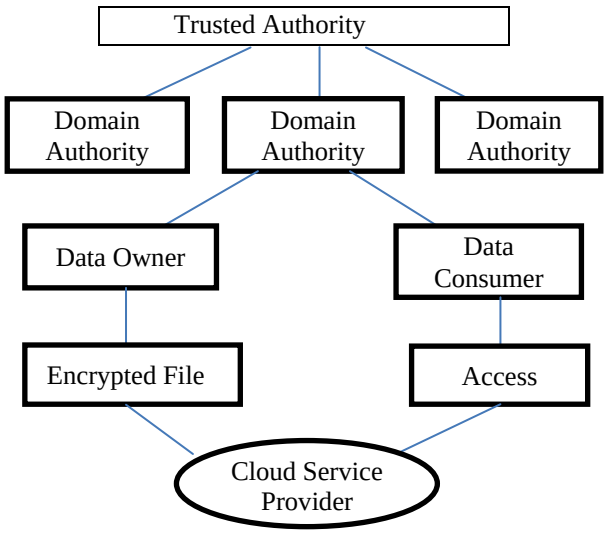

Figure 1. User level hierarchy of HASBE [7]

Major roles in existing system are “Data Owner”, Data Consumer”, “Domain Authority” and “Trusted Authority”. As discussed earlier user stores encrypted data on cloud, which can be regained by decrypting the same using private key provided. This helps us in maintaining the stored data confidential. As shown in Figure 1 higher level authority authorise lower level authorities. The biggest issue in cloud computing is loss of data ownership. Whenever data owner uploads a file, he will generate secret key for each file and data consumer has to request for the key from data owner. Consumer will decrypt the data using the key. The entire hierarchy of the system users is as shown in Figure 2

C = {C1, C2, C3, C4}

Where,
C is cloud
C1 is president.
C2 is vice president.
C3 is the list of superintendants.
C4 is the list of employees.

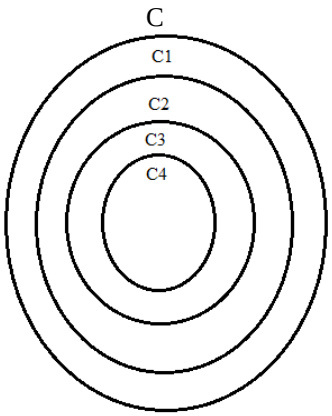

Figure 2. Domain Hierarchy [7]

As shown in Figure 2 user has to register by providing attribute values. Once user registeration is done, C1 (“president”) approves all the information of user and C1 provides userid and password to the user. User will store all data in encrypted format using PK (“public key”) and user can regain decrypted data from the cloud using PK (“public key”) and PRK (“private key”) provided by data owner. Data owner can access his own data by using private key and password provided. When C2 (“vice president”) wants to access employee’s attribute then MK (“Master key”) is used, which is produced by picking the attributes from the available set. If any lower level authority is offline or not responding then higher level authority will respond to all the requests of lower authorities [7]. The C2 (vice-president) will allocate access rights and direct the employees under him. Hence the administration of assignment of tasks to employees should be done in a method that is known to himself by immediate domain authority with the approval of “president” in root domain [7].

3. PROBLEM DEFINITION

Cloud’s dynamic nature and organizational work demands flexible Access control models with encryption. Data owners are facing a serious risk of corrupting or missing their data because of lack of physical control over their outsourced data. Access control models have to overcome this security risk. However, traditional Access Control mechanisms are based on static policies which make them too rigid to handle the complex situations in dynamic cloud.

The data owner and data consumer will not be always online. Data consumer required secret key to decrypt data file stored on cloud. For secret key data consumer has to wait for data owner to come online.

The another problem in existing system is displaying whole data associated to the demanded query even though the user requests fewer data due to lack of domain level hierarchy. Consider if user want to fetch only the dataof single student from science domain then it is not possible for the system to fetch specific specific data, instead it fetches the whole data of all the domains. Due to this, the time taken to get the data

and query processing is high. The system response time will be delayed, thereby reducing the overall system performance.

4. RESEARCH MODEL

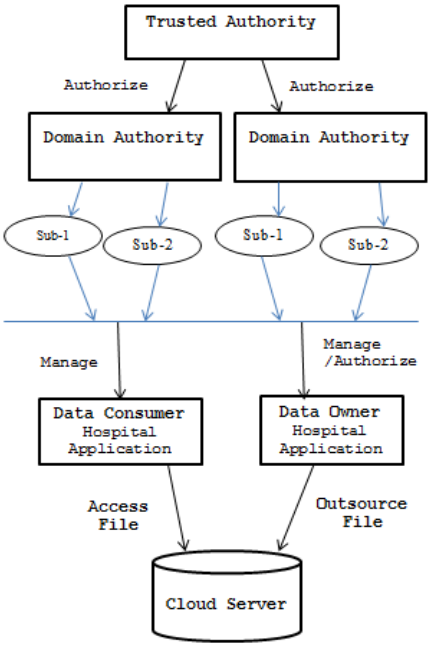

Figure 3. Sub-Domain Level User Hierarchy in Enhanced HASBE acccess control model

4.1. Sub-Domain Creation

As shown in Figure 3, we had proposed and an improved HASBE access control model by creating a sub-domain below the main domain.eg if the domain is engineering then below the Engineering domain, we can create branch specific sub domains like IT, C.E, Mech Engg etc. At the time of user registration, with respect to the data provided, user will be associated with particular sub domain under the main domain. This helps us in resolving the problem of searching whole data associated to the demanded query, instead specific data will be searched i.e. only sub domain data can be displayed instead of entire domain. This model helps us in reducing the time taken to fetch the data, decreasing overall query processing time and finally reduces overall system response time. Sub-domain creation optimizes system performance. Data owner will not be online always; this results in difficulty in key distribution. In our enhanced HASBE, data owner shares secret key, master key and particular access policy with the trusted authority, immediately after uploading file in cloud. The trusted authority will be always online. Whenever data consumer wants to access a particular file sends request for key to the trusted authority. In turn trusted authority will verify requested user attributes with policy, if matches, the user will be allowed to access the file, otherwise the access will be denied. Access policies are defined by data owner. Access policy consists of user attributes combined with conditions. Whenever data consumer request for access, trusted authority verifies access policy provided by the data owner with the attributes of requested user, if consumer satisfies the access policy secret key will be provided to the consumer, by using which he/she can decrypt the file.

4.2. Access Policy Creation

Data owner will define policy for each file and stores encrypted file on shared storage with specific file-id. Defined policy will restrict access of the file to unauthorized users. Access control policies will help data owner in protecting his secure data from unauthorized access. Users register with system by provided their department, role, age, gender information. This registration information will be considered as attribute values. Access rights will be given to users with matching attribute values with respect to the policy defined by data owner. Complex polcies can be defined by combining multiple attributes using “And”& “OR” conditions. Data owner can also use “and” & “OR” conditions together in single policy. Access policies like:

1. Depid=EC & Role =Faculty
2. Depid=EC or Depid=CE or Depid =IT and Role=Student
3. Depid=IT or Depid=CE or Depid-EC or age=>20
4. Depid! =EC & Role =Faculty

In above example the first condition describes that user from EC department and his/her role must be faculty. If user's Department is EC and his/her role is faculty then user can access the file.
In to the second condition Department id must be either IT or CE or EC and role must be student so this is a Combination of "And" and "OR" Condition.
Third condition is based on "OR" and "OR" condition.
In fourth condition if depid is not equal to EC and Role is equal to faculty then user can access the file. Similarly data owner can define policy of his choice using all ("and", "OR","=","")
"! =","> or < or <= or =>".

4.3. Encryption and Decryption

In our proposed system encryption and decryption of data file stored on cloud is done using Blowfish algorithm [13] .User has to first encrypt the file before storing it on cloud. In other hand consumer has to first download the file to the preferred location and perform decryption.

4.3.1. Blow Fish Algorithm

Blowfish is a popular keyed, symmetric cryptographic algorithm designed by Bruce Schneier in 1993 and placed in the public domain [13]. It is ideal for data exporting and has a 64 bit block size with variable key length from 32 bit to 448 bits. Blowfish is included in various encryption based products. Including Splash ID. Blowfish's security is highly proven. As a public domain cipher, Blowfish has been subject to a significant amount of cryptanalysis. Blowfish is also one of the fastest block ciphers in public use. "Each line represents 32 bits. The algorithm maintains two sub-key arrays: the 18-entry P-array and four 256-entry S-boxes. The S-boxes accept 8-bit input and produce 32-bit output. One entry of the P-array is used every round, and after the final round, each half of the data block is XORed with one of the two remaining unused P-entries. Figure 4 represents Blowfish's F-function. The function splits the 32-bit input into four eight-bit quarters, and uses the quarters as input to the S-boxes. The outputs are added Modulo 232 and XORed to produce the final 32-bit output" [13].

4.3.2. Working of Blowfish Algorithm

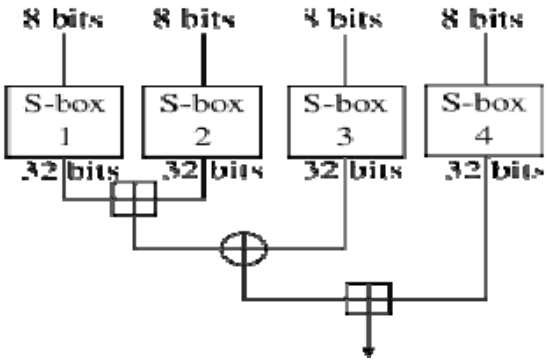

Figure 4. The Feistel structure of Blowfish [13]

As shown in Figure 4, "Blowfish is a Feistel network; it can be inverted simply by XORing P17 and P18 to the cipher textblock, then using the P-entries in reverse order. Blowfish's key schedule starts by initializing the P-array and S-boxes with values derived from the hexadecimal digits of pi, which contain no obvious pattern. The secret key is then XORed with the P-entries in order (cycling the key if necessary). A 64-bit all-zero blocks is then encrypted with the algorithm as it stands. The resultant cipher text replaces P1 and P2. The cipher text is then encrypted again with the new sub keys, and P3 and P4 are replaced by the new cipher text. This continues, replacing the entire P-array and all the S-box entries. In all, the Blowfish encryption algorithm will run 521 times to generate all the sub keys - about 4KB of data is processed" [13].

5. KEY GENERATION

The major functionalities performed in this section are system setup, data owner grant, data user grant, domain hierarchy setup, generating new file, data integrity check, file access, availability check and file deletion.The proposed scheme consists of 3 keys: Private, Public and Master Key. We are following the same key structure of existing system. We use Public key for encrypting the data, Private and public keys are together used to decrypt the data and Master key is used for accessing the data [6].

Setup *(d)*: d represents the depth of key structure. By taking depth parameter d as input. It gives a public key (PK) and master key (MK). ***KeyGen*** *(MK, u, a):* Master key (MK) user identity and attributes of key structure are taken as input to give private key PRK for user u.

Encrypt *(PK, M):* Public key (PK), and a message (M), are taken as an input for giving cipher-text (CT) as an output.

Decrypt *(CT, PRK):* Cipher text (CT) and private key of user (PRK) are taken as an input for decrypting the file. It outputs a message (M). If the attributes associated with the user private key (PRK) matches with the access structure of cipher text (CT), then it outputs a message M which is the original correct message. Otherwise, m is null. The modules considered to perform the above operations are DataOwner Module, Data Consumer Module, Cloud Server Module; Attribute based key generation Module [6].

6. PERFORMANCE ANALYSIS AND IMPLEMENTATION

In this section, we first analyze theoretical computation complexity of the proposed scheme in each operation. Then we implement an enhanced HASBE application with sub domain level and conduct a series of experiments to evaluate performance of our proposed scheme.

We analyze the computation complexity for each system operation in our scheme as follows.

System Setup:-When the system is set up, the trusted authority will select a bilinear group and some random numbers. When PK and MK are generated, there will be several Exponentiation operations. So the computation complexity of *System Setup* is **O (1).**

Top-Level Domain Authority Grant"-Process is executed by the TA("Trusted authority") The MK("Master Key") of a DA("Domain Authority") is in the form of **MKi= ("A,D,Di,Dj,D'I,D'j For ai","j Belongs to A","Ei for Ai € A")** where "**A"** is a key structure allied with a "**New Domain authority"**, Ai is the set of A . Let N be the number of attributes in A, and M be the numerous groups in A. Then the computation of MKi consists of two exponentiations for each attribute in "A" and one exponentiations for each group in A. The computation intricacy ("complexity") of "Top-Level Domain Authority" Grant operation is O **(2N+M).**

Sub-Domain Creation: - Similar to DA, process is executed by the TA and the MK of sub-domain is in the form of **MKi= ("A, D, Di, Dj, D'I, D'j For ai","j Belongs to A","Ei for Ai € A")** where "**A"** is a key structure allied with a "**New Domain authority"**, Ai is the set of A. Here creation of the sub-domain level does not increase any kind of complexity as we are not allocating separate keys for the sub-domain inside of the parent's domain. Keys are allocated to only parent's domain.Hence computation complexity of Sub domain authority is **O(2N+M)**

New User/Domain Authority Grant: New user or new domain authority/subdomain is associated with attribute sets, which are the sets of that of the upper level DA ("Domain Authority") the major computation overhead of this module is re-randomizing the key. The computation complexity is **O (2N+M).** Where N is the number of attributes in the set of the new user or domain authority, and M is the number of sets in **A.**

New File storing: The user needs to encrypt datafile using the Blowfish algorithm during file creation. The complexity of encrypting the data file with Blowfish Algorithm depends on the size of the data-file .Encrypting using Blow fish algorithm contains two exponentiations, for every foliage lump in T and one exponentiation for every interpreting lump in T. The Computation Complexity of new file storing is **("2|Y|+|X|").**

Table 1. Comparison of computation complexity

Operations	**Enhanced HASBE**	**HASBE** [6]
System setup	O(1)	O(|Y|)
Top-Level DA Grant	O(2N+M)	
User/DA Grant	O(2N+M)	O(|Y|)
Sub-Domain Grant	O(2N+M)	
File Creation	O(2|Y|+|X|)	O(|1|)
File Deletion	O(1)	O(1)

File Access:-

Here to access file user ahs to perform the "Decrypting" Operation of "Encrypted" data files. As discussed Data owner will "Encrypt" the data file before storing on cloud using Blowfish Algorithm and then "decrypt" data files using Blow fish algorithm. We will discuss the computation complexity of the algorithm. The comlexity of "Decrypting" CipherText differs based on the key used for "Decryption". Even for a particular key, the methods to fulfill the allied access tree may be differs. The algorithm comprises two joint actions for every singlefoliagelump used to fulfill the tree, one pairing for each interpreting lump on the path from the foliage lump used to the root and one exponentiation for each lump on the path from the foliage lump to the origin ("root"). So the computation complexity will be based on the access tree and key structure. It should be noted that the "Decryption" is accomplished at the data consumer side. Hence, its computation complexity has slight effect on the "scalability' of the general system. The computation complexity of file access O(1).

7. CONCLUSION

We had explored various attribute based access control models for cloud computing and developed an enhanced HASBE access control model, which is highly efficient in handling domain hierarchy. We had proved that, the complexity of HASBE can be reduced and efficiency can be improved by increasing the number of levels of domains. In future the system can be enhanced for efficiently handling compound attributes.Hasbe access model can be madeas dynamic access model by combining it with role based risk access control model.

REFERENCES

[1] Azeem Sarwar, Muhammad Naeem Khan "A review of trust aspects in cloud computing security" in *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*-ISSN: 2089-3337, Vol. 2, No. 2, April 2013, pp. 116~122.

[2] Rajanikanth aluvalu, lakshmi Muddana," *A Survey on Access Control Models in Cloud Computing"*-in *Springer International Publishing, Advances in Intelligent Systems and Computing* 337, DOI: 10.1007/978-3-319-13728-5_7.

[3] J. Bettencourt, A. Sahai, and B. Waters*"Ciphertext-policy attribute based encryption* "in Proceedings of *IEEE Symposium on Security and Privacy*, pp. 321V334, 2007.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, *"Attibute-based encryption for fine-grained access control of encrypted data,"* in Proc. *ACM Conf. Computer and Communications Security (ACM CCS)*, Alexandria, VA, 2006. International Journal of Computer Applications (0975 – 8887) Volume 112 – No. 7, February 2015 7.

[5] Zhibin Zhou and Dijiang Huang Arizona State University On *"Efficient Ciphertext-Policy Attribute Based Encryption and Broadcast Encryption"*.

[6] Zhiguo Wan, Jun'e Liu, and Robert H. Deng, Senior Member, IEEE, "*HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing*", in *IEEE transaction on information forensic and security*, vol. 7, no. 2, April 2012.

[7] Vanraj kamliya and Rajnikanth Aluvalu *"A Survey on Hierarchical Attribute Set-Based encryption (HASBE) Access Control model For Cloud Computing"*, in *International Journal of Computer Applications* (0975 – 8887) Volume 112 – No. 7, February 2015.

[8] Samy Gerges, Sherif Khattab, Hesham Hassan, Fatma A Omara *"Scalable Multi-Tenant Authorization in Highly-Collaborative Cloud Applications"*, in *International Journal of Cloud Computing and Services Science (IJ-CLOSER))*-ISSN: 2089-3337, Vol. 2, No. 2, April 2013, pp. 106~115.

[9] N.krishna. L.Bhavani," *HASBE: A Hierarchical Attribute Set Based Encryption For Flexible, Scalable And Fine Grained Access Control In Cloud Computing-International Journal of Computer & Organization Trends* –Volume 3 Issue 9 – Oct 2013.

[10] Md.Akram Ali, Ch. Pravallika, P.V.S. Srinivas," *Multi-Attribute Based Access Control Policy Enforcement for File Accesses inCloud"*-in *International Journal of Engineering Science and Innovative Technology (IJESIT)* Volume 2, Issue 5, September 2013.

[11] Sanchal Ramteke, Purvamodi, Apurva Raghojiwar, Vijaya Karad, Prof. P.D. Kale.*"HASBE: Hierarchical Attribute based solution for flexible and scalable access control* in *cloud computing*-in *International Journal of Scientific and Research Publications*, Volume 4, Issue 1, January 2014.

[12] Q. Liu, G. Wang, and J. Wu, *"Time based proxy re-encryption scheme for secure data sharing in a cloud environment," Information Sciences* .In Press, 2012

[13] Jawahar Thakur, Nagesh Kumar*" DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis" International Journal of Emerging Technology and Advanced Engineering (IJEATE)- (ISSN 2250-2459,* Volume 1, Issue 2, December 2011.

BIOGRAPHIES OF AUTHORS

Rajanikanth Aluvalu is Research scholar, currently persuing PhD from GITAM University. Currently working as Associate professor in the Department of Computer Engineering, RK University. He persued M.Tech Computer Science & Engineering from JNTU, Hyd. His research is in the area of cloud computing and his current research interests are in cloud computing security, cloud services and Bigdata Analytics.

Vanraj Kamliya working as assistant professor in department of Computer science at Glorious college of computer science. He persued his Masters in engineering. His research is in the area of cloud computing and his current research interests are in cloud computing security, Network security.

Lakshmi Muddana is working as professor and HOD in the Department of Information technology, GITAM University, Hyderabad Campus. She had persued Ph.D from Osmania University, Hyderabad. Her research is in the area of Data Mining and cloud computing and her current research interests are in cloud computing security, Bigdata Analytics