IJCSET |September 2012 | Vol 2, Issue 9, 1420-1426 www.ijcset.net | ISSN:2231-0711

Implementation of Secure and Efficient Model to Control Over Data Sharing in the Cloud Computing

M. Sri Pavani,
Dept of Computer Science & Engineering
Avanthi Institute of Engineering & Technology, Vizianagaram
vspavani80@rediffmail.com

K.Ravindra
Asst prof, Dept of CSE
Avanthi Institute of Engineering & Technology, Vizianagaram
komali.ravindra@gmail.com

Y. Ramesh Babu
HoD, Dept of CSE
Avanthi Institute of Engineering & Technology, Vizianagaram
javaramesh143@gmail.com

Abstract— Cloud computing is a rapidly growing segment of the IT industry that will bring new service opportunities with significant cost reduction in IT capital expenditures and operating costs, on-demand capacity, and pay-per-use pricing models for IT service providers. Among these services are Software-as-a-Service, Platform-as-a-Service, Infrastructure-as–a-Service, Communication-as-a-Service, Monitoring-as-a-Service, and Storage-as-a-Service. Storage-as-a-Service provides data owners a cost effective service to store massive data and handles efficient routine data backup by utilizing the vast storage capacity offered by a cloud computing infrastructure. However, shifting data storage to cloud computing infrastructure introduces several security threats to data as cloud providers may have complete control on the computing infrastructure that underpins the services. These security threats include unauthorized data access, compromise data integrity and confidentiality, and less direct control over data for data owner. The current literatures propose several approaches for storing and sharing data in the cloud environments. However, these approaches are either applicable to specific data formats or encryption techniques. In this paper, unlike previous studies, we introduce a secure and efficient model that allows the data owners to have full control over data sharing in the cloud environment. In addition, it prevents cloud providers from revealing data to unauthorized users. The proposed model can be used in different IT areas, with different data and encryption techniques, to provide secure data sharing for fixed and mobile computing devices.

Keywords- cloud computing; cloud storage; data sharing model; data access control; data owner full control, cloud storage as a service; data encryption

I. INTRODUCTION

Cloud computing acting as a hot topic in IT industry. Cloud computing is internet based development and is used in computer technology. Cloud computing manages and schedules the computing resources through network, and constitutes a large computing resources pool which can provide service to users on their demand [1]. The network is called “cloud”. Resources in cloud is seems that can be extended unlimitedly, got anytime, used on-demand and paid according to apply. It dynamically delivers everything as a service over the internet based on user demand, such as network, operating system, storage, hardware, software, and resources. These services are classified into three types: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (Saas).Cloud computing is a rapidly growing segment of the IT industry that will bring new service opportunities with significant cost reduction and increased operating efficiency for IT vendors. Cloud computing includes three major models: Software-as-a-Service, Platform-as-a-Service, and Infrastructure-as-a-Service [1]. Additional models are evolving as the concept of cloud computing develops new services such as Storage-as-a-Service, Communication-as-a-Service, and Monitoring-as-a-Service.

II. RELATED WORK

An important characteristic of cloud computing is pay-per-use [2]. Customers pay for cloud services only when they use them. Several cloud services are available to the public such as the Google App Engine [3] and Microsoft Live Mesh [4]. Storage-as-a-Service, such as Amazon simple storage service [5], gives data owners a cost effective service to store massive data and handles efficient routine data backup by utilizing the vast storage capacity offered by a cloud computing infrastructure. In addition, it gives customers the ability to expand and reduce IT resources as needed. However, with the development of cloud computing, deployment of IT systems and data storage is shifted to off-premises third-party IT infrastructures, i.e., cloud computing platforms. Shifting data storage to cloud computing infrastructure introduces several security threats to data, as cloud providers may have complete control on the computing infrastructure that underpins the services. These security threats include unauthorized data access, compromised data integrity and confidentiality, and less direct control over data for data owners. To overcome these threats, we present a

secure and efficient model that allows the data owners to have full control to grant or deny data sharing in the cloud environment. In addition, the proposed model ensures data integrity and confidentiality, and prevents cloud providers from revealing data to unauthorized users. The proposed model can be used in several applications such as remote file storage, data publication, on-demand data access, and online educational programs. Each application can use its data format and encryption technique to provide secure data sharing in the cloud. In addition, the proposed model uses a low computing power (e.g. symmetric encryption) and a one-authentication step to accept or deny a data access request. Therefore, it can be used with low computing power devices such as mobile devices. The remainder of this paper is organized as follows. In section II, we survey and analyze the related work. Section III describes the details of our proposed model, followed by the security analysis in section IV, and finally, section V concludes the paper. Deployment of storage as a cloud computing service, where data storage is shifted to off-premises third-party infrastructure, introduces special security threats. Therefore, data owners have to establish the following special security requirements to safeguard the data in the midst of un-trusted cloud environments:

There are four main types of cloud:

Public Cloud: The cloud computing resource is shared outside, anyone can use it and some payment maybe need.

Private Cloud: It is opposite to public cloud, private cloud's resource is limit to a group of people, like a staff of a company etc.

Hybrid Cloud: This is a mixture of previous two clouds, some cloud computing resource is shared outside but some don't.

Community Cloud: This is a special cloud to make use of cloud computing features. More than one community shares a cloud to share and reduce the cost of computing system. Data storage in cloud offers so many benefits to users:

(1) It provides unlimited data storage space for storing user's data.

(2) Users can access the data from the cloud provider via internet anywhere in the world not on a single machine.

(3) We do not buy any storage device for storing our data and have no responsibility for local machines to maintain data.[8]

III. ENSURING DATE INTEGRITY AND CONFIDENTIALITY

Within the cloud computing world, the virtual environment lets users' access computing power that exceeds that contained within their own physical worlds. To enter this virtual environment does not require the exact location of their data nor the other sources of the data collectively stored with theirs. To ensure data confidentiality, integrity, and availability (CIA), the storage provider must offer capabilities that, at a minimum, include

1. A tested encryption schema to ensure that the shared storage environment safeguards all data;
2. Stringent access controls to prevent unauthorized access to the data; and
3. Scheduled data backup and safe storage of the backup media.[2]The cloud storage providers should not have the capability of compromising the integrity and confidentiality of the data stored in the cloud. Confidentiality means keeping users' data secret in the cloud systems while data integrity means preserving information integrity, i.e., no data loss or modification by unauthorized users [6].

B. Controlling Data Access and Sharing: The data owner should be the only authority that grants and access to authorized users.

C. Authentication

The Authentication is used to verify the claimed identity of the data owner, user, or other entity [7] such as cloud provider. To meet these security requirements, data owners have to enforce authorization access policies that prevent revealing data information to cloud service providers or unauthorized users. Previous studies proposed several approaches for storing and sharing data in the cloud environments. However, these approaches are either applicable to specific data formats or encryption techniques. For example, the model introduced in [8] applies the publisher policy model presented in [9] to secure storage of Extensible Markup Language (XML) data in the cloud by adding special secure co-process to the stored machine, as part of the cloud infrastructure, to enable efficient encryption to the stored XML documents. Although mechanism published in [8] may enforce owner's policies on XML documents, the cloud providers have access to plain XML data. Reference [10] introduced a model for securing data sharing on the cloud. In that model, data sharing is achieved by re-encrypting the data to the authorized users by the cloud provider. Although model illustrated in [10] can enforce sharing policies, specified by data owners, and preventing unauthorized access to data, the model's idea works only with one encryption technique (progress elliptic curve encryption) and requires the cloud provider to re-encrypt the encrypted data before forwarding it to authorized users. Reference [11] introduced a model to outsource very large blocks of data by encrypting each block of data with a different encryption key. However, the model published in [11] fails to demonstrate how a user will ensure data confidentiality after receiving data from the cloud. In addition, whenever a user's access right is revoked, the data block group needs to be fragmented and several data blocks need to be re-encrypted. Our model is more secure and more efficient than the model presented in [11] and immune to eavesdropping attacks since, in our model, a user is not allowed to communicate with the cloud provider. In summary, our model gives the data owner full control to grant or deny data sharing in the cloud using efficient and secure procedures. In addition, it prevents cloud providers from revealing data contents to unauthorized users. The proposed model can be used in several applications (e.g. remote file storage, data publication, online educational programs), with different data and encryption techniques, to provide secure data sharing for both fixed and mobile computing devices.

IV. PROPOSED INTERNAL WORKING STEPS TAKEN FOR DATA SECURITY

In this section, we will explain our proposed access model based on a scenario illustrated in Figure 1 and notations listed in Table 1. As shown in Figure 1, a data owner, who stores his encrypted data in the cloud, receives a data access request from a user. After successfully authenticating the user and checking the policies, relevant to the user, the data owner sends a control message to the user and a data access permit to the cloud storage provider.

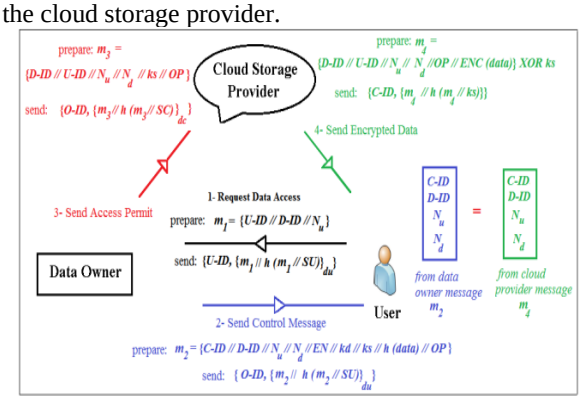

Figure 1. Secure Data Sharing Model with Full Control in the Cloud

TABLE I. MODEL'S NOTATIONS

Notation	Description	Comments
O-ID	Data Owner ID	
C-ID	Cloud storage provider ID	
U-ID	User ID	
D-ID	Shared data ID	
SU	User secret anonymity	Published by data owner
SC	Cloud provider secret anonymity	Published by data owner
du	Secret encryption key for exchanging messages between data owner and the user	Published by data owner
dc	Secret encryption key for exchanging messages between data owner and the cloud provider	Published by data owner
XOR	Logical exclusive or operation	
ks	A one-time session key to be used with XOR operation when transferring message from the cloud provider to the user	Generated by data owner
h (.)	A one-way secure hash function such as SHA-1	
||	A concatenation operator	
$\{.\}_k$	Encryption operator using encryption key, *k*	
EN	Encryption algorithm used for encrypting the shared data	Chosen by the data owner based on data type
ENC{data}	Encrypted data	Sent by cloud provider
kd	Encryption key used for encrypting the shared data	Chosen by the data owner
h(data)	Hash value of the shared data	Calculated at the data owner

The data access permit has relevant information that allows the cloud storage provider to apply data owner's policy and provides specific data to the user. Meanwhile the control message, sent by the data owner, will allow the user to decrypt and authenticate the data that will be granted from the cloud storage provider. As shown in step 4 in Figure 1, the user compares the information received from the data owner with information received from the cloud provider.If there is a match, the user ensures that the received information is valid and authentic. In the proposed model, a cloud storage provider has no knowledge about the data encryption algorithm and decryption key. This way, data owners keep control over data integrity and confidentiality in the cloud. Meanwhile, data owners control user policy access and reveal relevant information that grants users access and protects data against any modification.

For execution of this proposed model, the data owner first needs to complete the following tasks:

a)Issue two secret anonymities, SC and SU, for the cloud service provider and the user.

b) Issue two secret symmetric encryption keys, dc and du, for the cloud service provider and the user.

c) Use a secure channel, such as Diffie-Hellman key agreement [12], to exchange SC and dc with the cloud provider, and submit SU and du to the user In addition, we assume that the data owner encrypts the data with a suitable encryption algorithm, relevant to the data type, and submitted the encrypted data to the cloud service provider though a secure channel. The proposed model has the following five steps:

1. A user Request Data Access from the Data Owner A user who would like to access data, defined by D-ID, generates a nonce, Nu, and prepares a message m1= {U-ID // D-ID // N_u} to be sent to the data owner. The user then sends a request data access message = {U-ID, {m_1 // h (m_1 // SU)}$_{du}$} to the data owner.

2. Data Owner Authenticates and Sends Control Message to the User Upon receiving the data access request from the user, the data owner executes the following steps:
   a)Decrypt the received message, using the symmetric secret key, du, (that is relevant to U-ID) and obtain m1 = (U-ID,D-ID // N_u), and h (m_1 // SU).
   b)Verify the format of U-ID, D-ID from the decrypted message m1. If there is no match, the data owner terminates the connection. Otherwise, the data owner continues.
   c)Compute h (m1 // SU) and check whether it equals the received h (m1 // SU)). If there is a match, the data owner determines the authenticity of the user. After authenticating the user, the data owner generates a nonce, N_d, a one-time session key, ks, and prepares two special messages m2, and m3 to be sent to the user and the cloud provider respectively. The message, m2= {C-ID // D-ID // N_u // N_d // EN // kd // h (data) // ks // OP}, contains the following parameters: cloud provider

identification, C-ID, shared data identification, D-ID, message nonce, Nu and Nd, the encryption algorithm, EN, encryption key, kd, and data hash value, h (data), that are relevant to the data (D-ID), a onetime session key, ks, and optional field, OP. The optional field, OP, could be used to extend the capability of the proposed model. For example, the optional field could have the time when the data should be accessed (e.g. for downloading a test on an online educational program) or special access policy that could be related to Mandatory Access Control **(**MAC) or Role Based Access Controls (RBAC) [13]. After preparing the message, m2, the data owner sends the control message = {O-ID, {m2, h (m2 // SU)}$_{du}$} to the user. Upon receiving the control message, {O-ID, {m2, h (m2 // SU)}$_{du}$}, the user will authenticate and check the integrity of the received message as follows:

a)Decrypt the received message, using the symmetric secret key, du, and obtain m2= {C-ID // D-ID // Nu // Nd // EN // kd // ks // h(data) // OP}, and h (m2 // SU) b)Compare the values of D-ID and Nu, obtained from m2, to those values sent in message m1. If there is a match, the user continues.

c) Compute h (m2, SU) and check whether it equals the received h (m2 // SU)). If there is a match, the user authenticates the data owner.

d)Keep C-ID, ks, and N_d for processing cloud provider message, m4, in step 5.

3. Data Owner Sends a Data Access Permit to the Cloud Provider In addition to sending the control message to the user, the data owner prepares a message m3 = {D-ID // U-ID // Nu // Nd // ks // OP} and sends a permit data access message = {O-ID, {m3 // h (m3 // SC)dc}} to the cloud provider
4. Cloude Provider Sends the Encrypted Data to the User Upon receiving the grant data access message, {O-ID, {m3 // h (m3 // SC)}$_{dc}$}, the cloud provider executes the following steps:

a)Decrypt the received message, using the symmetric secret key, dc, (that is relevant to O-ID) and obtain m3 = {D-ID, U-ID // Nu // Nd // ks // OP}, and h (m3 // SC).

b)Verify the format of D-ID from the decrypted message m3.If there is no match, the cloud provider terminates the connection. Otherwise, the cloud provider continues.

c)Compute h (m3 // SC)) and checks whether it equals the received h (m3 // SC)). If there is a match, the cloud provider ensures the authenticity of the data owner.

d)Extract ks from m3 and prepare a message m4 = {D-ID, UID // N_u // N_d // OP // ENC {data}} XOR ks.

e) Send a message = {C-ID, m4 // h (m4 // ks)} to the user defined by U-ID, obtained from message m3, as shown in Figure 1.

5. User Verifies the Received Data from the Cloud Provider

Upon receiving a message {C-ID, m4 // h (m4 // ks)} from the cloud provider, the user retrieves the one session key, ks, received from the data owner in m2, and executes the following steps:

a)Compute m4 XOR ks and obtain m4 = {D-ID, U-ID // N_u // N_d // OP // ENC {data}}.

b)Compute h (m4 // ks) and compare it with the received h (m4 // ks). If there is a match, the user continues.

c)Compare the values of C-ID, D-ID, Nu ,and N_d, received from cloud provider, to those values obtained from message m2, received from the data owner. If there is a match, the user authenticates the received message.

d)Encode the received encrypted data, ENC {data}, with the encoding key, kd, received from the data owner in m2.

e) Compute h (data) and compare it with h (data) obtained from the data owner in message m2. If there is a match, the user ensures the integrity and confidentiality of the received data.

V .SECURITY AND RESPONSIBILITY

Within the grid computing world, virtual environment lets user access computing power that exceeds that contained within their own physical worlds. To enter this virtual environment requires them to transfer data throughout the grid. Consequently, several data storage concerns can arise. Typically, users will know neither the exact location of their data nor the other sources of the data collectively stored with theirs. To ensure data confidentiality, integrity, and availability (CIA), the storage provider must offer capabilities that, at a minimum, include a tested encryption schema to ensure that the shared storage environment safeguards all data; stringent access controls to prevent unauthorized access to the data; and scheduled data backup and safe storage of the backup media. Legal issues arise, such as e-discovery. regulatory compliance (including privacy), and auditing .The range of these legal concerns reflects the range of interests that are currently using or could use grid computing. These issues and their yet-to-be-determined answers provide significant insight into how security plays a vital role in grid computing continued growth and development.

VI. SECURITY ANALYSIS OF THE PROPOSED MODEL

This section illustrates how the proposed model achieves security requirements for storing data in cloud environments and how it offers enhanced resiliency to security threats.

A. SECURITY REQUIREMENT ACHIEVED

1) Ensuring data integrity and confidentiality In the proposed model, since the data is stored in encrypted form on the cloud and the data owner keeps the encryption key and algorithm information, the cloud storage provider does not have the capability of compromising the integrity and confidentiality of the data stored in the cloud infrastructure.

2) Controlling data access and sharing

In the proposed model, since the data owner is the only authority that authenticates the user and issues the data

encryption information (algorithm and key) to authorized users, cloud providers cannot grant data access to unauthorized users.0

3) Authentication

Authentication is the act of establishing or confirming claims made by or about the subject are true and authentic [14]. In the proposed model, authentication is achieved by using a hash code that contains a secret anonymity *SU* or *SC* and encrypt by a secret encryption key (*du* or *dc*) as shown in Figure 1.

Hash Function

1. **Declare character 'str' of unsigned long type.**
2. **Declare and initialize hash of unsigned integer type**
3. **Unsigned hit hash = 0;**

Int q;

While (q=str+1)

hash =hash + q;

B. Represents this digest as an integer m between 0 and n-1

C.) Uses her private key (n. d) to compute the signature , s = m power d mod n.

D.) Sends this signature s to the recipient, B.

For example, the data owner appends a secret user's anonymity, *SU*, to the exchanged message, *m*2, before computing its hash code, *h* (*m2* // *SU*). The data owner then encrypts the exchanged message, {*m2* // *h* (*m2* // *SU*)} by the secret symmetric key (*du*) and sends it to the user.

B. RESILIENCE AGAINST SECURITY THREATS

This subsection shows how the proposed model is resilient to security threats such as unauthorized data access attack, information disclosure during sharing, and other security attacks.

1) Unauthorized data access attack

Since data owners keep the encryption information (key and algorithm) and check the identity of users, unauthorized data access is not possible in our model. In general, unauthorized data access attacks occur by one of the following methods:

1. *The attacker acquires data from the cloud storage provider. In our model, the user doesn't initiate any messages with the cloud provider to gain data access. Even if the cloud provider sends data to an unauthorized user, the user can't decrypt the received message since the encryption information (key and algorithm) is not known to unauthorized users and to the cloud providers. Therefore, it is not possible for unauthorized users to know the encryption information without the help of the data owner.*
2. *The attacker acquires data access from the data owner. To get data access permission from the data owner, the attacker must have the knowledge of user anonymity, US, and the encryption key, du. It is not possible for the attacker to guess both parameters and access the data.*

2) Information disclosure during sharing attack

Since data is always in its encrypted form, there is no way data can be decrypted before it is delivered to authorized users. This ensures that the entire sharing process will not disclose information to cloud providers and unauthorized users. To acquire data during sharing, an attacker must have the decryption key and algorithm. Since this information is kept with the data owner, cloud storage providers and unauthorized users cannot decrypt the data.

3) Data owner/user's identify guessing attack

As shown in Figure 1 and Figure 2, the user/data owner appends a secret user's anonymity to the exchanged message (*m1*/*m2*) before computing its hash code, and then encrypts the exchanged message by the secret symmetric key, *du*. Both secrets (*SU*, and *du*) are known only to the data owner and the authorized user. At the receiving side, the data owner/user decrypts the message and appends the same secret anonymity, *SU*, to the message before calculating its hash code to check the message's authenticity. Since the hash code provides authentication and the encryption provides confidentiality to the exchanged message between data owner and user, the adversary can't guess the user's anonymity from the exchanged messages and therefore can't imitate user identity to create a new data access request. Similarly, the adversary cannot imitate a data owner and send fake data access to a user.

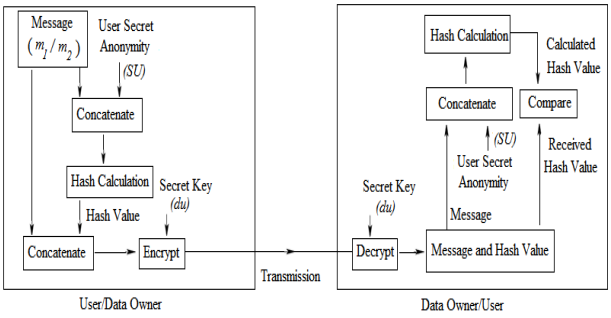

Figure 2. Securing transmission between the data owner and the user

4) Cloud provider's identity guessing attack

As shown in Figure 1 and Figure 3, the data owner uses a cloud provider's anonymity, *SC*, and encryption key, *dc*, to provide authentication, by hash code, and confidentiality, by encryption, when sending messages to the cloud provider. Therefore, the adversary cannot guess the cloud's anonymity from the exchanged messages. Similarly, the adversary cannot imitate a data owner and sends fake data access permit messages, *m3*, to the cloud provider.

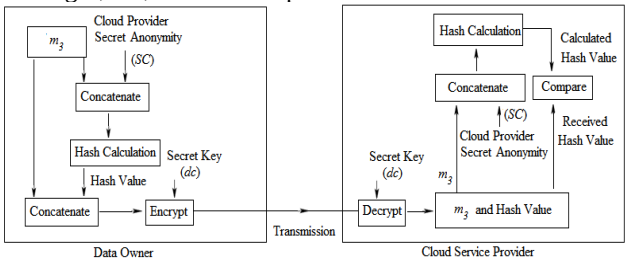

Figure 3. Securing transmission between the data owner and the cloud service provider

5) Impersonation attack

An impersonation attack involves an adversary who attempts to impersonate a data owner, a user, or a cloud provider.

a) An adversary can't imitate a data owner to grant a user data access without knowing user secrets (SU, du), cloud provider secrets (SC, dc), and data encryption information (encryption algorithm, data encryption key).

b) Without knowing the secrets (SU, du), an adversary cannot imitate a user to decrypt the message m2 and then get data access

c) Since the cloud provider doesn't know the data encryption algorithm, EN, the data encryption key, kd, and the message encryption key, ks, (issued by the data owner to the authorized user), an adversary cannot imitate a cloud provider to provide users with fake data.

6) Replay attack

A replay attack is a method in which an adversary tries to replay messages obtained in previous communications. For example, an adversary might replay the used message m1 to the data owner requesting data access and then receive the message *m2* from data owner. However, the adversary cannot derive correct data information (data ID, data encryption algorithm, and data encryption key) from *m2* since he or she cannot decrypt *m2* without knowing secrets *SU*, and *du*. In addition, the adversary will not be able to decrypt *m4*, received from the cloud service provider, since he or she cannot reveal the one time encryption key, *ks*, issued by data owner in message, *m2*.

VII. CONCLUSION

Among the many IT giants driven by trends in cloud computing has not doubtful. It gives almost everyone has brought good news. For enterprises, cloud computing is worthy of consideration and try to build business systems as a way for businesses in this way can undoubtedly bring about lower costs, higher profits and more choice; for large scale industry, Data security has become the most important issue of cloud computing security.

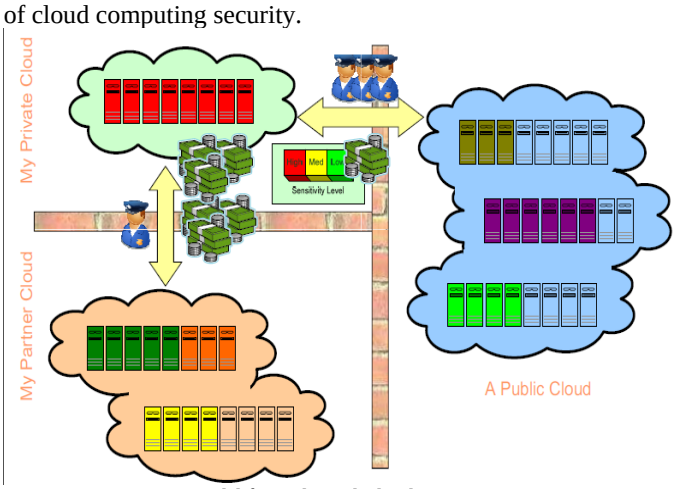

Fig4: Model for Federated Cloud Computing

Though Many solutions have been proposed, many of private and public and partner cloud them only consider one side of security; As in fig4 We proposed that the cloud data security must be considered to analyze the data security risk, the data security requirements, deployment of security functions and the data security process through encryption.The main contribution of this paper is the new view of data security solution with encryption, which is important and can be used as reference for designing the complete security solution.

This paper has introduced a secure and efficient model that offers the data owner full control to grant or deny data sharing in the cloud environment. In addition, it prevents cloud providers from reveling data to unauthorized users. The proposed model can be used in several applications such as remote file storage, data publication, on-demand music access, and online educational programs. Each application can use its own data format and encryption technique to provide secure data sharing in the cloud. In addition, since the proposed model uses low computing power (e.g. symmetric encryption) and a one- authentication step to accept or deny a data access, it can be used with mobile or fixed devices. Security analysis has demonstrated that the proposed model meets cloud security requirements and is resilient to several security threats.

VIII. FUTURE EXPECTATION

In future work, we believe that data storage security in Cloud Computing, an area full of challenges and of paramount importance, are still in its infancy now, and many research problems are yet to be identified is to enhance the more security features by using other enhanced techniques of data security through cryptosystems and other techniques.[1][5]

REFERENCES

[1] Uma Somani , Kanika Lakhani ,Manish Mundra , "Implementing digital signature with RSA encryption algorithm to enhance data security of cloud in cloud computing ", PGDC 2010 pp 211- 216

[2] John harauz , Lori M.Kaufman , Bruce potter , "Data security in world of cloud computing " by IEEE computer and reliability societies , jul /Aug 2009 Pp 61-64

[3] Cong Wang, Qian Wang, and Kui Ren, Wenjing Lou," Ensuring Data Storage Security in Cloud Computing" at IEEE (8-1-4244-3876-1/09)

[4] Xiao Zhang, Hong-tao Du ,Jian-quan Chen, Yi Lin, Lei-jie Zeng , "Ensure Data Security in Cloud Storage",

[5] T. Sridhar, "Cloud computing – a primer, Part 1: models and technologies," The Internet Protocol Journal, vol. 12 (3), pp. 2–19, September 2009.

[6] J. W. Rittinghouse and J. F. Ransome, "Cloud computing: implementation, management, and security," CRC Press. Boca Raton, 2010

[7] Google Inc., "Google app engine," 2011, retrieved in March 2011 from http://appengine.google.com

[8] Microsoft Inc., "Microsoft live mesh," 2011, retrieved in March 2011 from http://www.mesh.com

[9] Amazon Inc., "Simple storage service," 2011, retrieved in March 2011 from http://aws.amazon.com/s3

[10] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and Privacy in Cloud Computing: A Survey," Sixth international conference on semantics, knowledge and grids, pp.105-112, 2010.

[11] C. Kaufman, R. Perlman, and M. Speciner, "Network security: private communication in a public world," Upper Saddle River, New Jersey: Prentice Hall Press, 2002

[12] K. Hamlen, M. Kantarcioglu, L. Khan, and B. Thuraisingham, "Security issues for cloud computing," International Journal of Information Security and Privacy , vol. 4 (2), pp. 39-51, 2010.

[13] E. Bertino, B. Carminati, E. Ferrari, B. Thuraisingham, and A. Gupta, "Selective and authentic third party distribution of XML documents," IEEE Transactions on Knowledge and Data Engineering , vol. 16 (10), pp- 1263-1278, 2004.
[14] G. Zhao, C. Rong, J. Li, F. Zhang, and Y. Tang, "Trusted data sharing over untrusted cloud storage providers," 2nd IEEE international conference on cloud computing technology and science, pp- 97-103,2010
[15] W. Wan and Z. Li, "Secure and efficient access to outsourced data," 16th ACM conference on computer and communication security, 2009.
[16] W. Diffie and M. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory , vol. 22 (6), pp- 644-654, 1976
[17] M. Ciampa, "Security Guide to Network Security Fundamentals," Boston, MA: Course Technology, Cengage Learning, 2009
[18] R. Zhang and L. Liu, "Security models and requirements for healthcare application clouds," IEEE 3rd International Conference on Cloud Computing, 2010

M. Sri pavani has completed her MCA from Osmania University. She is pursuing post graduation in M.Tech at JNTU kakinada.

Ravindra has completed his M.Sc (CS) degree from Pydah, Andhra University. He has completed M.Tech (CSE) in Andhra University. He has many years of experience in teaching field, presently working as Assistant Professor, Computer Science & Engineering at Avanthi Institute of Engineering and Technology, Cherukupally, Vizianagaram (Dt), Andhra Pradesh.

Y. Ramesh babu has completed his M.Sc (CS) degree from Andhra University. He has completed M.Tech (CSE) in Andhra University. He has many years of experience in teaching field, presently working as Associate Professor & Head, Department of Computer Science & Engineering at Avanthi Institute of Engineering and Technology, Cherukupally, Vizianagaram (Dt), Andhra Pradesh

.