International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 3, Issue 4, April 2014 ISSN 2319 - 4847

Cloud Data Storage Security Enhancement Using Identity Based Encryption

Varsha S.Agme[1], Prof. Archana C.Lomte[2]

[1] Varsha S.Agme BSIOTR,India
[2] Prof .Archana C .Lomte, BSIOTR,Pune,India

Abstract

In cloud computing, data owner can outsource his data on cloud server and only the authorized user can access or request for this data .This service is called as Database-As-A-Service. But there are different issues related to the confidentiality, correctness, query and access permission of outsourced data because Cloud is managed by un-trusted third party. Owners are always suspected about security of the data. So we consider this as basic idea for propose concept. The existing system does not provide security against possible attack of clouds, but the propose system can provide security against Collusion attack, DDOS attack .By using the concept of re-encryption the data get more secure and access permission that who will access the data is decided by only the data owner .Now a days uses of Android based devices has been increasing day by day because it can be easily carried out anywhere and easily inserted in pocket, so we decide to develop a propose system application which will access from Android based devices. In existing system to provide better security data owner has to be online all the time so our propose system will be helpful for data owner by providing notification about request of user.

Keyword: data owner, DDOS, Proxy server, re-encryption key.

1. INTRODUCTION

In Cloud computing technology includes many technologies such as the autonomic computing virtualization, utility computing, service oriented architecture and many. The purpose of these technologies to provide scalable, shared resources- software and hardware and providing services over the network. The cloud term 'as a service' is referred to as providing something as a service over the network. There are 3 types of services provided by cloud as: Software as a Service (SaaS), Platform as a Service (Paas), Infrastructure as a Service (IaaS) and many more. All the provided services are based on policy of on-demand fashion in which users can pay only to for their required usage. Today, many cloud service providers such as Amazon's EC2 and S3, Microsoft's Windows Azure Google's App Engine providing the facility to different users. Users who cannot afford such huge cost to build their own huge infrastructure, so they can have their work done by the help of cloud providers at minimum cost.

As per the type of users and the hosting of environment the cloud architecture can be divided as four types: 1.Public Cloud, 2.Private Cloud, 3.Hybrid Cloud, 4.Community Cloud. Public cloud provides services, are hosted for public usage and anyone can have their data stored and services get done using this cloud. Data security is most important issue here. Private cloud, where the data access and service usage are restricted to one authority only. In Hybrid cloud, it shared by a limited no. of organizations and it combined the features of both private and public cloud. Community cloud is much like the private cloud but in this the data is shared among the same entities of the one organization.

Including this cloud also provides service called as database-as-a-service, in this data owner can outsourced his data/files on cloud server for reducing space cost as well as maintenance cost and only the authorized legitimate user can query /request this data. In this before uploading data on cloud server data owner has to be encrypt his data. Proxy servers can perform some functions on the stored cipher text without knowing anything about the original data/files. Cloud server is untrusted server because it managed by is untrusted third party. Therefore, here issues of confidentiality comes in existence ,data owner mostly concerned on the security from unauthorized access, integrity means correctness of the file/data after outsourcing to the proxy server, as well should not be modified by unauthorized user or even though by the proxy server. So, this is the reason that it becomes major research problem among research community and it growing day by day.

In this paper, we propose a system for security enhancement of cloud computing using identity-based encryption ,it can capture the following properties: (1) without the help of the private key generator (PKG) the file owner can decide the access permission independently; (2) For one request, a receiver can only access one file, instead of all files of the owner; (3) Our schemes are secure against the DDOS and (4)also secure against collusion attacks, namely even if the receiver can compromise the proxy servers, he cannot obtain the owner's secret key. (5) Can get the notification about user request on his android based device which is not possible in existing system. (6) Can get the messages related secret key on android based device which will increase the security level of this system.

1.1 LITERATURE SURVEY

Following are some related existing encryption and re-encryption schemes related to identity based system.

1) *Identity based proxy encryption*

Identity-based proxy encryption (IBPE) was first proposed by Ivan and Dodis [1] in which they presented security model for both unidirectional and bidirectional IBPE schemes. In this schemes, the master secret key, used to extract secret keys, split into two parts. One is sent to the proxy server and the other is sent to the user. The user can decrypt a cipher text for him with the help of the proxy server. But disadvantage of this system is that it is not collision safe.

2) *Identity based proxy re- encryption*

The first identity-based proxy re-encryption (IBPRE) was developed by Green and Ateniese [3] where the proxy server can transfer a cipher text for the owner to a cipher text for the user after he gets a re-encryption key from the former. The IBPRE schemes divides into the following two types based on the generation of the re-encryption key:

a) The re-encryption key can be computed by the owner

In this paper [3], to decrypt the cipher text, the owner selects a random number and computes a re-encryption key by randomizing his secret key. Then, he encrypts the selected random number under the user's identity. Finally, he sends the re-encryption key and the cipher text to the proxy server. Using the re-encryption key, the proxy server can transfer a cipher text for the owner to a cipher text for the user.

The user decrypts the cipher text using his secret key and obtains the random number selected by the owner. Then, he can decrypt the re-encrypted cipher text by the random number. Unfortunately, these schemes are safe to the collusion attacks. If the user can compromise the proxy Server, they can decrypt the cipher text, obtain the random number selected by the owner and compute the secret key of the owner.

b) The re-encryption key can be computed by the Private Key Generator

This system proposed by L. Wang, M. Mambo, and E. Okamoto [6], in this system, the PKG computes the reencryption key by checking the secret keys of the owner and the user.

3) *Identity-based Secure Distributed Data Storage*

This system, proposed by J.Han, Willy Susilo, and Yi Mu [5], a user's identity can be an arbitrary string and two parties can communicate with each other without checking the public key certificates. At first, the file owner encrypts his files under his identity prior to outsourcing them to servers. Then, he sends the cipher texts to the proxy servers.

Consequently, the proxy servers can transfer a cipher text encrypted under the identity of the owner to a cipher text encrypted under the identity of the` receiver after they has obtained an access permission (re-encryption key) from the owner.

To provide confidentiality for the outsourced data, propose scheme provides the following properties:

1. Unidirectional: After receiving access permission that who will accessing which data, the proxy server can transfer a cipher text for Alice to a cipher text for Bob while he cannot transfer a cipher text for Bob to a cipher text for Alice.
2. Non-interactive: The access permission can be created by the file owner only without any trusted third party and interaction with him.
3. Key optimal: The size of the secret key of the receiver is constant and independent of the delegations which he accepts.
4. Collusion-safe: The secret key of the file owner is secure from malicious user even if the receiver can compromise the proxy server.
5. Non-transitive: Receiving the access permissions computed by A for B and B for C, the proxy server cannot transfer a cipher text for A to a cipher text for C.
6. File-based access: For one query/request, the receiver can only access only one file. This can improve the security of the outsourced files and is desirable to maintain the access record.

For provide better security, to accept the user request, in this system data owner has to be online all the time. So in our propose system we provide a facility to data owner in which he can get the notification regarding the user request/query on his android based device.

2. METHODOLOGY

2.1 PROPOSED WORK

In this paper, we propose a system for cloud computing where, for one query, the receiver can only access one file, instead of all files. In other words, access permission (re-encryption key) is bound not only to the identity of the receiver but also the file. The access permission can be decided by the data owner, instead of the trusted party (PKG).

Furthermore, our schemes are secure against the collusion attack and will also detect the distributed DOS attack which possible on proxy server in which server can't proceed to legitimate work and this system will available to owner on his/her android based device. And can get notification also .To enhance the security level we provide secret key on android based device.

Our ideas can be summarized as follows

1. The access permission is decided by the data owner therefore the propose work secure against collusion attack.
2. The existing scheme is not detect the Distributed Denial of service(DDOS) attack which possible on proxy server to make the server busy and stop to respond to authorized user by sending request repeatedly .So propose work makes the system DDOS free.
3. Our propose work provides the facility to system users to access the system and can get the request notification on android based device.

1) System architecture

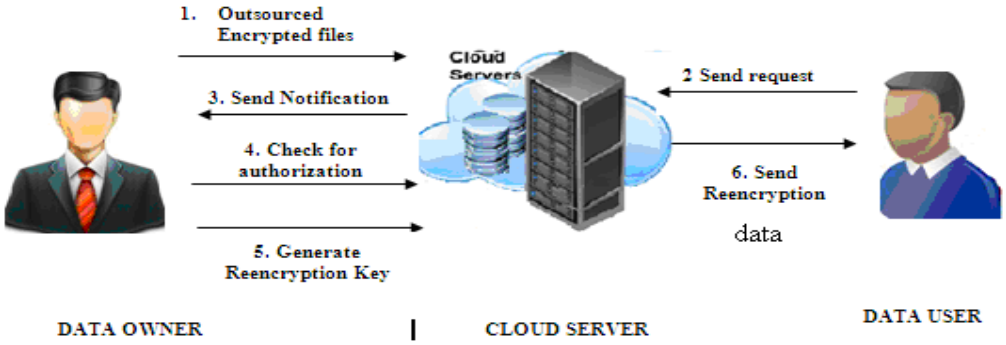

Figure1. System architecture/Block Diagram for Cloud Data Storage Security Enhancement Using Identity Based Encryption

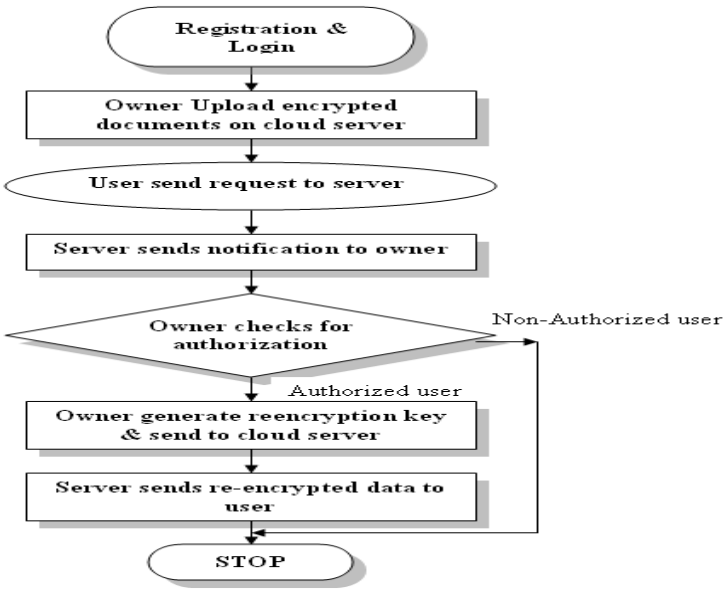

Figure2. System Flowchart for Cloud Data Storage Security Enhancement Using Identity Based Encryption

As shown in fig, the data owner, the proxy server and the receiver. The data owner encrypts his data before outsources to the proxy servers. The data owner check for the authorization of the user if the requested user is authorized then and then only the reencryption key generated and transfer to the proxy server. Proxy servers store the encrypted data and transfer the Cipher text from the owner to the receiver when they obtain access permission (re-encryption key) from the owner.
The receiver authenticates himself to the owner and decrypts the re-encrypted cipher text to obtain the data. This will be the advantage of this scheme to achieve more security that only authorized user can access the system based on identity and the secret key can be obtain on his mobile.

2) Mathematical Module

Let G and Gτ be two multiple cyclic groups with prime order p, and g be a generator of G. A bilinear map e :G x G → Gτ is a map satisfies the following properties:

(1) Bilinearity. For all u, v εG and x, y ε Z_p, e (u^x, v^y) = $e(u, v)^{xy}$.

(2) No-degeneracy. e(g, g) ≠ 1 where 1 is the identity of the group Gτ .
(3) Computability. There exists an efficient algorithm which can compute e(u, v) for all u, v ε G.

a) Key Generation:

Let id is an identity which is an n bit string, id_i be the i^{th} bit of id. I: be the set which Consists of the entire index i with $id_i = 1$. This algorithm takes as input the user's Identity id, and computes secret key- $Kid = (u_0 \prod u_i)(id)$,Where G=denotes a bilinear group, which is a set with an associated binary operation that satisfies the group axioms, g = generator for G, $u_0 \leftarrow^R$G and U= (u_1, u_2 …. u_n) where $u_i \leftarrow^R$G for i=1, 2…n. Z is set of n integers.

b) Encryption:

Suppose there are k messages (M1, M2…Mk).To encrypt the message Mi, the Owner O chooses $s_i \leftarrow^R Zp$ and computes Ci1=Mi.e(g1,ή)si,ci2=g^{s_i} and ($u_0 \prod u_i)^{s_i}$for i=1,2,...k. The cipher text for message Mi is C^{T_i}=(C^{i_1}, C^{i2}, and C^{i_3}).For reencryption he computes $C'i_1$=D2.Ci_1, $C'i_2$=Ci_2, $C'i_3$=Ci_3, $C'i_4$=D1, $C'i_5$=D3 and $C'i_6$=Kid2.

c) Decryption: To decrypt the cipher text C^{T_i}, the user computes - $M_i = C'i_1 \cdot \frac{e(C'i_6, Ci_3)}{e(K, Ci_2)}$.

3) Algorithms used for implementation:

a) Algorithm 1: Encryption and decryption algorithm:

Input: Message or file/cipher text

Output: cipher text/original message or File

(1) Advanced Encryption algorithm (AES)-

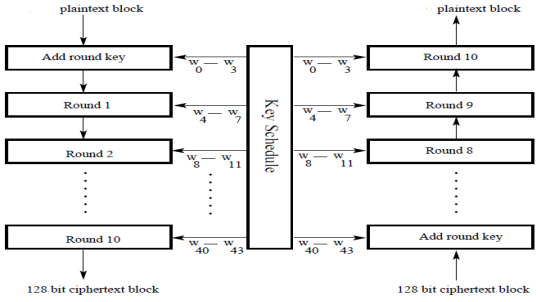

Figure 2 The overall structure of AES algorithm

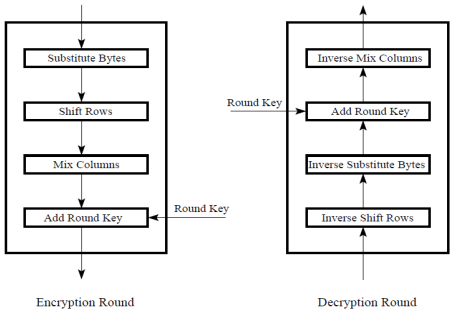

Figure 3 different steps that are carried out in each round

b) Key generation algorithm:

Input: ID of the owner and User

Output: Secret Key i.e. Kid

(2) Base64 Algorithm-

Base64 algorithm is designed to encode any binary data, stream of bytes, into a stream of 64-printable characters. In which, the binary data is transformed to ASCII text, which can be transported in email without problems. On the recipient's end, the data is decoded and the original file is rebuilt.

The Base64 encoding process is to:

Step 1: Divide the input bytes stream into blocks of 3 bytes.
Step 2: b) Divide 24 bits of each 3-byte block into 4 groups of 6 bits.
Step 3: Map each group of 6 bits to 1 printable character, based on the 6-bit value using the Base64 character set map.
Step 4: If the last 3-byte block has only 1 byte of input data, pad 2 bytes of zero (\x0000). After encoding it as a normal block, override the last 2 characters with 2 equal signs (==), so the decoding process knows 2 bytes of zero were padded.
Step 5: If the last 3-byte block has only 2 bytes of input data, pad 1 byte of zero (\x00). After encoding it as a normal block, override the last 1 character with 1 equal signs (=), so the decoding process knows 1 byte of zero was padded.

Step 6: Carriage return (\r) and new line (\n) are inserted into the output character stream. They will be ignored by the decoding process. Here are some examples for Base64 Encoding:

Example 1: Input data, 1 byte, "A". Encoded output, 4 characters, "QQ=="

Input Data	A			
Input Bits	01000001			
Padding		01000001	00000000	00000000
	\	\	\	
Bit Groups	010000	010000	000000	000000
Mapping	Q	Q	A	A
Overriding	Q	Q	=	=

Example 2: Input data, 2 bytes, "AB". Encoded output, 4 characters, "QUI=".

Input Data	A	B		
Input Bits	01000001	01000010		
Padding	01000001	01000010	00000000	
	\	\	\	
Bit Groups	010000	010100	001000	000000
Mapping	Q	U	I	A
Overriding	Q	U	I	=

Example 3: Input data, 3 bytes, "ABC". Encoded output, 4 characters, "QUJD".

Input Data	A	B	C	
Input Bits	01000001		01000010	01000011
	\	\	\	
Bit Groups	010000	010100	001001	000011
Mapping	Q	U	J	D

Base64 Encoding Table:

Table 1:Base64 Encoding Table

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

4) *Modules*

a) Registration /Login Module:

In this, separate modules for data owner who will work as administrator and the data users.

b) Key Generation Module:

In this module secret key can generate for each user; user can decrypt the cipher texts and obtain the original files if he knows the identity used to encrypt the files.

c) Encrypted file upload Module:

For security purpose at first, the file owner encrypts his files prior to outsourcing them to proxy servers.

d) Reencryption key Module:

The data owner checks the authorization of data user as per received AI from proxy server. If the AI is correct, the owner computes access permission (re-encryption key) and sends it to the proxy server. If the user is not authorized then the access is denied.

e) Document Retrieval Module:

The proxy server transfers the intended cipher text to the receiver using the received access permission. Finally, the receiver can decrypt the re-encrypted cipher text by his secret key and obtains the original file.

f) Notification Module:
In owner will get the notification of request on his android based device. The use of android based device has been increased day by day therefore we provide this facility.
g) DDOS Module:
As this scheme is used for Cloud computing therefore there may be the chances of distributed denial of service attack on server, so IBSEDS scheme will resist to the DDOS attack in this module.

3. CONCLUSION
Cloud computing is a distributed system in, where different users in different domains can share data among each other. Different Identity-based proxy re-encryption schemes have been proposed to outsource sensitive data from the owner to an external party. Nevertheless, they cannot be employed in cloud computing. As security of data storage is important, also the security of data transfer is important. We enhance the security of data transfer by introducing the identity based secure encryption and reencryption. It will provide many advantages like collusion-resistance over the previous schemes and will get the notification of user request on android based device and will also provide security against Distributed Denial of Service attack and it provides secure model of cloud storage with safe data forwarding**.**

REFERENCES
[1] Ivan and Y. Dodis, "Proxy cryptography revisited," in Proc.*Network and Distributed System Security Symposium - NDSS'03, (San Diego, California, USA), pp. 1–20, The Internet Society, Feb. 2003*
[2] Armbrust M, Fox A, Grith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. *A view of cloud computing. Communications of the ACM 2010*
[3] Bouganim L, Pucheral P. Chip-secured data access: Confidential data on untrusted servers. In: Proceedings: *International Conference on Very Large Data Bases: Morgan Kaufmann 2002: 131-142.*
[4] Fernando N, Loke SW, Rahayu W. Mobile cloud computing: A survey. *Future Generation Computer Systems 2013; 29(1): 84-106.*
[5] Han, J., Susilo, W. & Mu, Y. (2013). Identity-based data storage in cloud computing. Future Generation Computer Systems*: international journal of grid computing: theory, methods and applications, 673-681.*
[6] L. Wang, L. Wang, M. Mambo, and E. Okamoto, "New identity based proxy re-encryption schemes to prevent collusion attacks," in Proc. *Pairing-Based Cryptography - Pairing'10, vol. 6487r, Dec. 2010.*
[7] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in Proc. *Applied Cryptography and Network Security - ACNS'07 , vol. 4521*, pp. 288–306, Springer, Jun. 2007.
[8] Qin Liu, Guojun Wang and Jie Wu, "Efficient Sharing of Secure Cloud Storage Services," *10th IEEE International Conference on Computer and Information Technology, 2010*