Enhancing Cloud Security Based On Group Signature

Arumugam Sakthivel
Department of Computer Science and Engineering, Kalasalingam University, India

Abstract: *Using the eccentric of truncated preservation, cloud computing gives a reasonable and proficient result for distributing cluster resources among cloud clients. Regrettably, distributing data in a multi user fashion whereas maintaining data and individuality privacy from an unfaith cloud is quiet a puzzling concern, because of the recurrent change of the participation. The proposed system focuses a protected multi user data distributing method, for active clusters in the cloud. Using group signature and active broadcast encryption methods, any cloud client can secretly distribute data among others. Provisionally, the storage load and encryption calculation cost of the proposed method is liberated from the amount of repealed clients. Additionally, the security and performance analysis of the proposed method shows that, much more efficient and secure than all other existing methods.*

Keywords: *Active broadcast encryption, cloud, data distribution, group signature*.

Received September 12, 2014; accepted June 18, 2015

1. Introduction

Today, the people in the world are affected with so many health related problems and there are some health problems which are unknown to the doctors. In that situations the doctors need to know how to treat the patients, to cure this type of health problems. Cloud allows the doctor to share the patient health record to several doctors and ask the treatment which was known by other doctors. Patient Health record and cure method shared in a cloud should be secure. Only authorized doctors are allowed to access the data. Group of doctors those are specialized in specific domain are registered with cloud and use the cloud. Doctor who doesn’t know the treatment for a sick can share the patient record to the other doctors in a cloud. The patient record should be in an encrypted form. The authorized doctors can get the patient record and specify the method, prescription and dosage level to cure the sick.

The main tricky issue in a cloud is to provide a security because of the following concerns.

1. Recognizing privacy is the most important problem in a cloud computing [9].
2. Any member in a group should be capable to store and share the data in a cloud.

Groups are generally dynamic in nature. The changes of membership make secure data sharing trickier [10]. The system dares new users to know the content of data stored before their membership, because it is an unattainable for new users to contact with data owners, and get the decryption keys [4]. Cloud service providers are not fully trusted by the users. To avoid the security and privacy issues, the data stored in a cloud should be in a non-readable form and only allow authorized user to access the data. For that, many algorithms are proposed. Those are Attribute Based Encryption (ABE), fine-grained access control, asymmetric encryption, Identity Based Encryption (IBE), Message Authentication Code (MAC), Homomorphic Linear Authentication (HLA) and etc., in those approaches, data owner store the data in an encrypted form and deliver the decryption keys only to authorized users. So the unauthorized users and third party CSP cannot know the data stored in a cloud.

Yu *et al.* [18] presented a secure, scalable fine grained access control in a cloud computing based on an attribute based encryption. But this scheme requires high computation. Lu *et al.* [11] presented a secure provenance by cipher text policy attribute based encryption which allows any user to share data with others. This scheme is failed to support the user revocation efficiently. Boneh and Franklin [3] presented an Identity Based Encryption from the Weil Pairing which provides the security against chosen ciphertext attack. This scheme has a problem to build chosen cipher text secure IB systems. Goyal *et al.* [6] presented a scheme called Attribute based encryption for fine grained access control of encrypted data which provides a security against chosen ciphertext attack, but has problem to hide the set of attributes. Erway *et al.* [5] presents a dynamic data possession maintains provable updates to stored data but this system slowdown the performance.

To overcome the demerits listed above, propose a secure data sharing scheme in a cloud. The main contributions of this scheme include:

1. This scheme allows user in a group can share secret

data with others.

2. This scheme efficiently supports the dynamic group. New users can read the data in a cloud without asking permission from data owners. Revocation of user can be done through the revocation list generated by the group manager.
3. Private keys of the remaining users won't be changed.

The remainder of this paper is organized as follows: section 2 gives the survey of literature. Section 3 gives the proposed system. Section 4 has the experimental setup. Performance was analyzed in section 5. Section 6 has the conclusion.

2. Survey

Goyal *et al.* [6] proposed an attribute based encryption for fine grained access control of encrypted data that develops a cryptosystem called Key Policy Attribute Based Encryption (KP-ABE). In that system ciphertexts are marked with the set of attributes and user private keys are related with the access control that specifies which part of encrypted data is able decrypt by the specified users. This system uses the audit log information and broadcast encryption. It supports the allocation of private keys which includes the Hierarchical Identity Based Encryption (HIBE) and it leaves the open problem to hide the set of attributes.

Yu *et al.* [18] proposed a secure, scalable, and fine grained data access control in cloud computing. This system defines and enforces access policies based on attributes. Allow the data owner to hand over most of the computation tasks to the cloud without revealing the data contents. This system uses the KP-ABE for achieving fine grained access control and new user membership. This system combines the proxy re-encryption and lazy re-encryption for user revocation. This scheme achieves confidentiality and accountability. This system requires high computation overhead of cloud.

Lu *et al.* [11] proposed a new secure provenance the essential bread and butter of data forensics in cloud. Secure provenance is a method to trace the ownership and process record of data objects [12]. It uses the bilinear pairing techniques and also provides the confidentiality for the user data. But computation overhead is high because it requires handling multiple keys.

Wang *et al.* [16] proposed a privacy preserving public auditing for secure cloud storage. This paper is to enable the auditability for ensuring the integrity of data in a cloud by using the Third Party Auditor (TPA). TPA should not learn the data in a cloud. For that homomorphic linear authenticator and masking is used. This method is safe and proficient at single user setting. It failed to support multi user environment. Wang *et al.* [14] proposed a system knox privacy preserving auditing for shared data with large groups in the cloud. In a cloud the data is stored and exclusively shared with multiple users in a group. The quantity of data and time taken by the TPA are not concerned with the number of users in a group. But the computation cost of this system is higher.

Limitations in a cloud security can be overcome, by proposing a method which uses group signature. This successfully removes the necessity to rely on the storage server for preventing unauthorized access and this scheme efficiently supports the user revocation. Storage and encryption overhead are free from the amount of revoked users.

The proposed approach is partially related to several recent works in the cloud. Ateniese *et al.* [1] proposed Provable Data Possession (PDP), which permits a user to confirm the integrity of data stored at cloud without reclaiming the entire data. However, this method is only fitting for static data. To develop the competence of verification, Ateniese *et al.* [2] proposed scalable and efficient provable data possession with symmetric keys. Regrettably, this method cannot maintain public verifiability and only suggests each user a limited amount of verification desires.

Juels and Kaliski [8] proposed a model called proofs of retrievability (POR), which is capable to verify the suitability of data on a cloud. The novel data is inserted with a set of casually prized test blocks called sentinels. The confirmer dares the cloud by identifying the location of a group of sentinels, and by raising the cloud to revisit the linked sentinel values. Shacham and Waters [13] proposed two developed POR mechanisms, which are fabricated on BLS signature and pseudo random function. Wang *et al.* [17] used the Merkle hash tree for the construction of a public auditing system with entirely dynamic data.

Hao *et al.* [7] proposed an active public auditing system based on RSA. Erway *et al.* [5] proposed an active PDP founded on the rank based valid dictionary. Zhu *et al.* [20] proposed index based hash tables to maintain entirely active data. To guarantee the rightness of users data stored on several servers, Wang *et al.* [15] proposed homomorphic tokens and cutting codes in the inspection process.

Wang *et al.* [16] utilized data privacy by public inspection in the cloud. In this method, the TPA is capable to verify the reliability of cloud data but cannot get any confidential data. Zhu *et al.* [19] proposed a method to protect the data confidentiality from the TPA. Regrettably, it was not willingly scalable to inspecting the veracity of data distributed among a bulky amount of users in the group.

Let G_a be an additive cyclic group and G_m be a multiplicative cyclic group of order p. Then bilinear map e can be $G_a \times G_a \rightarrow G_m$. Bilinear map possess the following properties:

1. *Bilinear*: For all m, n $\in$ Z and U, V $\in G_a$, e(mU,

nV)=e(U, V)^{mn}.

2. *Non Degenerate*: There exists a point U such that e(U, U) ≠ 1.
3. *Computable*: For any U, V Є G_a there is an efficient algorithm to compute e(U, V).

- *q-Strong Diffie Hellman Assumption (q-SDH)*: Given (U1, U2, γU2, γ2U2, … , γqU2), it is impossible to calculate 1/(γ+x)U1, where x Є Z.
- *Decision Linear Assumption (DLA)*: Given U1, U2, U3, mU1, nU2, cU3, it is impossible to decide whether m+n=c mod p.
- *Weak Bilinear Diffie Hellman Exponent Assumption (WBDHE)*: For any m Є Z, given D, mD, m2D, …, mlD, U Є Ga, it is impossible to calculate e(D, U)1/m.
- *(t, n) General Diffie Hellman Exponent Assumption (GDHE)*: Let f(H)=Πri=1 (H+xi) and g(H)=Πn-ri=1(H +xi') be two arbitrary univariate polynomials. For any y, γϵ Z, then F0,γF0, …, γt-1F0, γf(γ)F0, U0, …, γt-1U0, yg(γ) P0ϵ Ga and e(F0, P0) f2(γ) g(γ)ϵ Gm, it is impossible to calculate e(F0, P0) yf(γ) g(γ)ϵ Gm.

3. Proposed System

To reach secure data distribution for active groups in the cloud, this paper combines the group signature and active broadcast encryption methods. Specifically, the group signature method permits members to mysteriously use the cloud resources, and also the active broadcast encryption method allows the data owners to safely distribute data files among others as well as new connecting members.

Regrettably, every member has to calculate revocation factors to safe the confidentiality from the revoked members in the active broadcast encryption method, which consequences in both the calculation overhead of the encryption and the amount of the ciphertext raise with the amount of revoked members. Accordingly, the weighty overhead and bulky ciphertext range may hamper the agreement of this encryption method to capacity limited members.

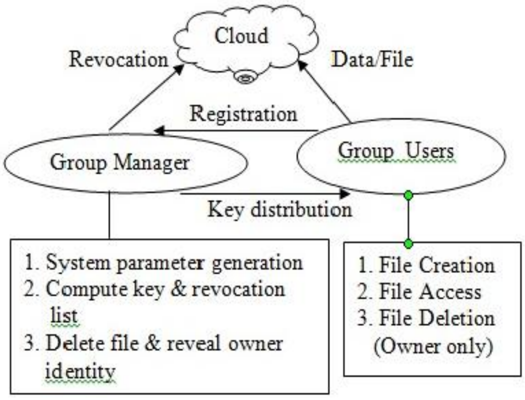
Cloud
Revocation
Data/File
Registration
Group Manager
Group Users
Key distribution
1. System parameter generation
2. Compute key & revocation list
3. Delete file & reveal owner identity
1. File Creation
2. File Access
3. File Deletion (Owner only)

Figure 1. Secure cloud data storage system.

To deal with this tricky issue, the group manager calculates the revocation factor and constructs the outcome openly available by moving that into the cloud. Figure 1 shows that the overall architecture of the proposed system. Group Manager is responsible for new client membership and client repeal.

- *System Initialization*: The group manager computers the system factors and performs the system initialization as follows:
    - Creating a bilinear map system M=(p,G_a,G_m, e(.,.)).
    - Choosing two arbitrary elements P, P_0 Є G_a beside with two arbitrary numbers €$_1$, €$_2$ Є Z, and calculating A= €$_1^{-1}$ P & B= €$_2^{-1}$ PЄ G_a such that €$_1$.A = €$_2$.B = P. Additionally, the group manager P_1= € $_1$ P_0 and P_2=€$_2$$P_0$ϵ G_a.
    - Arbitrarily selecting two elements U, FЄ G_a and a number γ Є Z, and calculating E = γ.U, D=γ. F and X=e(F,U), correspondingly.
    - Publishing the system factors including (M, U, P, P_0, P_1, P_2, A, B, E, D, X, f, f_1, Enc()), where f is a one way hash function {0, 1}*→*Z*; f_1 is hash function {0, 1}*→G_a and Enc() is symmetric encryption algorithm.

(*γ*, *€*$_1$, *€*$_2$, *F*) is a group manager master key that will be kept secret.

- *Registration*: For member i with identity ID_i the group manager arbitrarily chooses a number x_i ϵ Z and calculates I_i, J_i by the following equation 1 and 2:

$$I_i = \left[1/(\gamma + x_i)\right] U \varepsilon G_a \tag{1}$$

$$J_i = \left[x_i / (\gamma + x_i)\right] F \varepsilon G_a \tag{2}$$

Subsequently, the group manager puts (I_i, x_i, ID_i) into the member list. Then member *i* get a (x_i, I_i, J_i) as a private key.

- *Revocation*: Revocation operation is carried out by the group manager through a publicly available Revocation List (RL), founded on which group users can encrypt the files and guarantee the privacy against the revoked members. Table 1 shows the format of revocation list.

Table 1. Revocation list.

ID_{group}	I_1	x_1	t_1	U_1	Z_r	t_{RL}	Sig(RL)
	I_2	x_2	t_2	U_2			
	.	.	.	.			
	I_r	x_r	t_r	U_r			

ID_{group} is a group identity, I_i is a partial private key of user i, t is a revoked time, t_{RL} represent the freshness of the RL, and sig(RL) = γf_1(RL). Hence, the group manager moves the RL to cloud.

- *Data Generation*: The group user performs the following activities to store and distribute the data in a cloud:

- User sends the IDgroup to the cloud. The cloud considers it a request for RL, and then cloud sends the RL to user.
- User can check the validity of RL by ensuring signature and date of RL. If the RL is worthless then user ignores the scheme.
- Choose the unique data identity IDdata. The key can be selected by the two ways:

  1. If no revoked member in RL as shown in Equations 3 and 4:

  $$C_1 = k.D\text{Є}G_a,\ C_2 = k.U\text{Є}G_a \tag{3}$$

  $$K = X^k\text{Є}G_m \tag{4}$$

  2. If r revoked members in RL as shown in Equations 5 and 6:

  $$C_1 = k.D\ \text{Є}G_a,\ C_2 = k.U_r\text{Є}G_a \tag{5}$$

  $$K=X_r^{\,k}\ \text{Є}G_m \tag{6}$$

- Encrypt the data M. and choose the arbitrary number г and calculating f(г). Adds (IDdata, г) into local storage.
- Upload the encrypted data to cloud by signing the encrypted data.

Algorithm 1. Signature Generation:

```
Input: Private key (A, x) system parameter (U, A, B, P, E) and data M.
Output: Group signature on M.

Begin
Choose arbitrary numbers $\alpha,\beta,r_\alpha,r_\beta,r_x,r_{\delta 1},r_{\delta 2}$ Є Z
Put $\delta 1 = x_\alpha,\ \delta 2 = x_\beta$
Computes the following values
        $T_1= \alpha.A,\ T_2 =\beta.B,\ T_3 = I_i + (\alpha + \beta).P$
        $R_1= r_\alpha.A,\ R_2= r_\beta.B$
        $R_3=e(T_3,U)^{tx}\ e(P,E)^{-\gamma\alpha-\gamma\beta}\ e(H,P)^{-\gamma\delta 1-\gamma\ \delta}$
        $R_4=r_x.T_1-r_{\delta 1}.A,\ R_5= r_x.T_2-r_{\delta 2}.B$
        Put $c = f(M,T_1,T_2,T_3,R_1,R_2,R_3,R_4,R_5)$ Then
        $S_\alpha=r_\alpha + c\alpha,\ S_\beta=r_\beta + c\beta,\ S_x=r_x + c_x$
      $S_{\delta 1} = r_{\delta 1} + c,\ S_{\delta 2} = r_{\delta 2} + c$
      Return $_\sigma = (T_1,T_2,T_3,c,S_\alpha,S_\beta,S_x,S,S_{\delta 1},S_{\delta 2})$
  End
```

Algorithm 2. signature verification.

```
Input: System factors (U, A, B, P, E, M) and signature $_\sigma = (T_1,T_2,T_3,c,S_\alpha,S_\beta,S_x,S,S_{\delta 1},S_{\delta 2})$
Output: True or False.
Begin
        Compute the following values
        $R_1 = S_\alpha.A - c.T_1,\ R_2 = S_\beta.B - c.\ T_2$
        $R_3= (e(T_3,E)/e(U,U))^c\ e(T_3,P)^{Sx}\ e(H,W)^{-S\alpha-S\beta}$
        $R_4 = S_{x.}\ T_1-S_{\delta 1}.\ A,\ R_5 = S_{x.}\ T_2-S_{\delta 2}\ .\ B$
        If $c=f\ (M,T_1,T_2,T_3,R_1,R_2,R_3,R_4,R_5)$
        Return True
        else
        Return False
  End
```

Algorithm 3. Revocation verification.

```
Input: System factor ($P_0, P_1, P_2$), a group signature $\sigma$, and a set of revocation keys $I_1,...,I_r$
Output: legal or Illegal.
Begin
        Set temp = $e(T_1, P_1)e(T_2, P_2)$
        for i = 1 to n
                if $e(T_3 - I_i, P_0)$ = temp
                Return legal
                end if
        end for
        Return Illegal
  End
```

Algorithm 4. parameter computing.

```
Input: The revoked user factors $(U_1, x_1)... (U_r, x_r)$, and user partial private key (I, x).
Output: $I_{r,r}$ or Empty.
Begin
Set temp = I
for $\mu$ = 1 to r
        if $x = x\ \mu$
        return Empty
        else
        set temp =1/ $(x- x\mu)\ (U\mu$ - temp)
        return temp
End
```

- *Data Deletion*: Either data owner or group manager can delete the file in a cloud. The data owner gets the (ID_{data}, г) from the local storage. Call the group signature algorithm to calculate the signature on (ID_{data}, г) and send the signature to a cloud for deletion request. Then cloud check the signature and compute the f(г). If both the hash values are equal then cloud deletes the file. Group manager can delete the file by calculating signature $\gamma f_1(ID_{data})$, then send the signature with the ID_{data} to cloud. The cloud check the signature by equating $e(\gamma f_1(ID_{data}).U) = e(E,f_1(ID_{data}))$. If both are equal then cloud will delete the file.
- *Data Access*: To access the data stored in a cloud, user performs the following:
- User uses the partial private key (I, x) and compute the signature σu on (IDgroup, IDdata,t) and send the (IDgroup, IDdata,t,σu) to the cloud then cloud sends the requested data after verifying the validity of signature.
- The user verifies the validity of RL.
- Check the validity of data and compute the key without asking to the data owner. It includes the three cases:

  a) If ($t_{data}< t_1$) no user revoked before the data was uploaded. Then key can be $K= e(C_1, I)\ e(C_2, J)$.

  b) If ($t_i< t_{data}< t_{i+1}$) i users revoked before the data was uploaded and Key $K = e(C_1, I_{i,r})\ e(C_2, J)$.

  $$I_{i,r} = 1/[(\gamma + x)\coprod_{\lambda=1} i\,(\gamma + x_\lambda)].U$$

  c) If ($t_r< t_{data}$) r users are revoked before the data file was uploaded and Key $K=e(C_1,I_{r,r})\ e(C_2, J)$.

  $$I_{r,r} = 1/[(\gamma + x)\coprod_{\lambda=1} \gamma(\gamma + x_\lambda)].U$$

4. Experimental Setup

The test setup uses the 512 MB RAM, 80 GB hard disk and 2 GHz processor. Java language with miracl library is used in windows OS and java with PBC library is used in OS Ubuntu. Cloudsim is used for creating a cloud environment. The setup is simulated by using java programming language with GMP Library, Miracl Library, and PBC Library. The simulation contains three factors:

1. Client side.
2. Manager side.
3. Cloud side.

Together client and manager Progressions are behavioral on a laptop. The cloud process is implemented on a machine that equipped with dual Core 2.3 GHz, DDR3 RAM 2G. U

For simulation, elliptic curve algorithm with 160 bit is used, which delivers a reasonable security level with 1,024 bit RSA algorithm. By using the PBC Library to produce a bilinear map for system initialization. Especially, use the pbc_test.h header file with function pbc_demo and parameters a.param in the subdirectory of the PBC Library to set TypeA coupling factors. TypeA couplings express the symmetric bilinear couplings that are built on the curve y2 = x3 + x over the field Fp for some prime number q = 3 mod 4. The entrenching degree k is 2, and G_m is a subgroup of Fp2. The order p is certain prime element of q + 1. For protected usage, initialize p=160 bit and q=512 bit, respectively. The functions f and f1 are the hash functions constructed by the element_from_hash in the system factors. Furthermore, use the AES encryption algorithm to designate the Enc() symmetric encryption. For suitability, a distributed file factor.txt is used to store the other factors including U,P,P_0, and P_1.

Table 2. Comparision of client computation cost among ABE, ODBE and group signature.

Method	The amount of repealed clients				
	0	20	40	60	80
	File Creation (100 MB)				
ABE	1.62	1.98	2.08	2.17	2.45
ODBE	1.4	1.8	1.85	2.05	2.3
Group Sign	1.403	1.392	1.406	1.402	1.403
ABE	File Access (100 MB)				
	1.8	2.02	2.35	2.54	2.95
ODBE	1.6	1.85	2.2	2.44	2.8
Group Sign	1.579	1.678	1.746	1.824	1.949
ABE	File Deletion (100 MB)				
	1.8	2.02	2.35	2.54	2.95
ODBE	1.6	1.85	2.2	2.44	2.8
Group Sign	1.579	1.678	1.746	1.824	1.949

- *Computation Cost*: Computation cost of cloud and client is considered tolerable, even when the amount of withdrawal clients are enormous. Table 2 shows the computation cost of ABE, ODBE and group signature. From the Table 2, analyzed that computation cost of group signature is very tolerable than ABE and ODBE.

5. Performance Analysis

Performance of the proposed system is analyzed in terms of storage and computation cost.

- *Storage*: Without lack of simplicity, setting p = 160

  and the constituents in G_a and G_m to be 161 and 1,024 bit, correspondingly. Additionally, adopting the size of the data uniqueness is 16 bits, which revenue a group ability of 216 files. Likewise, the size of clients and group originality are also fixed as 16 bits.

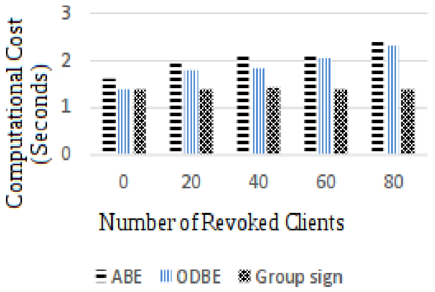

Figure 2. Comparison on client computation cost for file creation among ABE, ODBE and group signature.

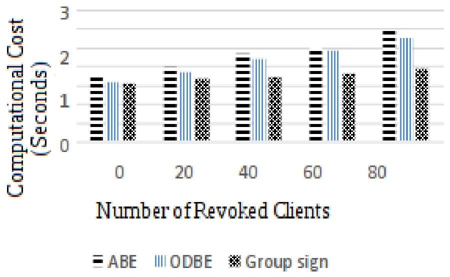

Figure 3. Comparison on client computation cost for file access among ABE, ODBE and group signature.

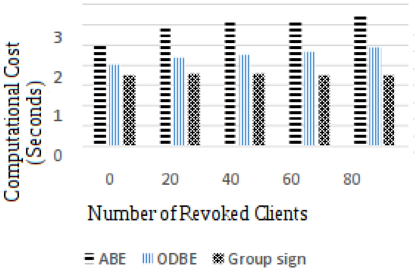

Figure 4. Comparison on client computation cost for file deletion between ODBE and group signature.

- *Group Manager*: In the proposed system, the major private key of the group manager is (γ, ϵ_1, ϵ_2, F) € $G_a \times Z_q$. In addition, the client list and the distributed file list should be deposited at the group manager. Considering the system with 200 clients and assuming that each client distribute averagely 50 files, the total storage capacity of the group manager is (80.125+42.125*200+2*10000)*10^{-3} =28.5 KB, which is very tolerable.
- *Group Members*: Ultimately, each client in a proposed system only requires to store their private

key (I_i, J_i, x_i)€ $G_a^2 \times Z_q$, which is almost 60 bytes. There exists a balance between the storage and the calculation overhead. e.g., the four coupling process containing (e(P, E), e(P, U), e(U, U), e(I_i, U)) ϵ G_m^4 can be precomputed one time and kept in a cloud for the group signature creation and authentication. Hence, the total storage of every clients is almost 572 bytes.

- *The Extra Storage Overhead in the Cloud*: In a proposed system, C is the ciphertext of the data in the symmetric encryption algorithm, the additional overhead to store the data is about 248 bytes, which contains (ID_{group}, ID_{data}, C1, C2, C, f(г), t_{data}, σ).
- *Computation cost*: Assessment on calculation cost of users for file creation process between the proposed system, ABE, and the method that straight using the Original Dynamic Broadcast Encryption algorithm.

From the Figures 2, 3, and 4 perceived that the calculation cost in proposed system is extraneous to the amount of repealed users. In contrast, this cost growths with the amount of repealed clients in ODBE and ABE.

6. Conclusions

The main application in a cloud is healthcare and stock market. This paper focuses a protected data distributing method in a cloud. For the efficient and secure data distribution, group signature and active broadcast encryption method is used. Moreover, this method maintains the proficient client withdrawal and novel client linking. In particularly, competent client withdrawal can be accomplished over a public revocation list, that is free from changing the reserved keys of the remaining clients, and new clients can openly decrypt the data's kept in the cloud before their membership. Additionally, the storage and encryption calculation costs are constant. Broad analysis shows that this proposed method fulfills the preferred security needs and guarantees proficiency as well.

References

[1] Ateniese G., Burns R., Curtmola R., Herring J., Kissner L., Peterson Z., and Song D., "Provable Data Possession at Untrusted Stores," *in Proceeding of 14th ACM Conference on Computer and Communication Security*, Alexandria, pp. 598- 610, 2007.

[2] Ateniese G., Pietro R., Mancini L., and Tsudik G., "Scalable and Efficient Provable Data Possession," *in Proceeding of 4th International Conference on Security and Privacy in Communication Networks*, Istanbul, pp. 1-10, 2008.

[3] Boneh D. and Franklin M., "Identity Based Encryption from the Weil Pairing," *Lecture Notes in Computer Science*, vol. 2139, pp. 213-229, 2001.

[4] Delerablee C., Paillier P., and Pointcheval D., "Fully Collusion Secure Dynamic Broadcast Encryption with Constant Size Ciphertexts or Decryption Keys," *in Proceeding of International Conference on Pairing-Based Cryptography*, Tokyo, pp. 39-59, 2007.

[5] Erway C., Kupcu A., Papamanthou C., and Tamassia R., "Dynamic Provable Data Possession," *ACM Transactions on Information and System Security*, vol. 17, no. 4, pp. 213-222, 2015.

[6] Goyal V., Pandey O., Sahai A., and Waters B., "Attribute Based Encryption for Fine Grained Access Control of Encrypted Data," *in Proceeding of 14th ACM Conference on Computer and Communication Security*, Alexandria, pp. 89-98, 2006.

[7] Hao Z., Zhong S., and Yu N., "A Privacy Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 9, pp.1432-1437, 2011.

[8] Juels A. and Kaliski B., "Pors: Proofs of Retrievability for Large Files," *In Proceedings of the 14th ACM Conference on Computer and Communications Security*, Virginia, pp 584-597, 2007.

[9] Lakshmanan T and Muthusamy M, "A Novel Secure Hash Algorithm for Public Key Digital Signature Schemes," *The International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 262-267, 2012.

[10] Liu X., Zhang Y., and Yan J., "Mona: Secure Multi Owner data sharing for dynamic groups in the cloud," *IEEE Transactions on parallel and distributed systems*, vol. 24, no. 6, pp. 1182-1191, 2013.

[11] Lu R., Lin X., Liang X., and Shen X., "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," *in Proceeding of the 5th ACM Symposium on Information, Computer and Communications Security*, Beijing, pp. 282-292, 2010.

[12] Naor D., Naor M., and Lotspiech J., "Revocation and Tracing Schemes for Stateless Receivers," *in Proceeding of Annual International Cryptology Conference*, California, pp. 41-62, 2001.

[13] Shacham H. and Waters B., "Compact Proofs of Retrievability," *in Proceeding of International Conference on the Theory and Application of Cryptology and Information Security*, Melbourne, pp. 90-107, 2008.

[14] Wang B., Li B., and Li H., "Knox: Privacy Preserving Auditing for Shared Data with Large Groups in the Cloud," *in Proceeding of International Conference on Applied*

Cryptography and Network Security, Singapore, pp. 507-525, 2012.

[15] Wang C., Wang Q., Ren K., and Lou W., "Ensuring Data Storage Security in Cloud Computing," *in Proceeding of 17th International Workshop on Quality of Service*, Charleston, pp. 1-9, 2009.

[16] Wang C., Wang Q., Ren K., Chow S., and Lou W., "Privacy Preserving Public Auditing for Secure Cloud Storage," *IEEE Transactions on computers*, vol. 62, no. 2, pp. 362-375, 2013.

[17] Wang Q., Wang C., Ren K., and Lou W., "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," *in proceeding of 14th European conference on Research in Computer Security*, Saint-Malo, pp. 355-370, 2009.

[18] Yu S., Wang C., Ren K., and Lou W., "Achieving Secure, Scalable, and Fine Grained Data Access Control in Cloud Computing," *in proceeding of IEEE on Information Communications*, San Diego, pp. 1-9, 2010.

[19] Zhu Y., Hu H., Ahn G., Yau S., "Efficient Audit Service Outsourcing for Data Integrity in Clouds," *Journal of System and Software*, vol. 85, no. 5, pp. 1083-1095, 2012.

[20] Zhu Y., Wang H., Hu Z., Ahn G.J, Hu H., and Yau S., "Dynamic Audit Services for Integrity Verification of Outsourced Storage in Clouds," *IEEE Transactions on Services Computing*, vol. 6, no. 2, pp. 227-238, 2013.

Arumugam Sakthivel is professor of Department of CSE in Kalasalingam University, India. He has 16 years experience in teaching and 7 years in research. He has published 25 papers in National/International Conferences as well as Journals and reviewed three text books. He got three times best reviewer ward from IAJIT. He is also reviewer of IEEE Transactions of Cloud Computing. His area of interest is in Cyber Security and Cyber Forensics.