



An Efficient File Hierarchy Attribute Based Encryption Scheme in Cloud Computing

Shital Panchal¹, Kamthane A.N.², Shital Gaikwad³

P.G. Student, Department of Computer Science & Engineering, MPGI School of Engineering, Nanded,
Maharashtra, India¹

HOD, Department of Computer Science & Engineering, MPGI School of Engineering, Nanded, Maharashtra, India²

Assistant Professor, Department of Computer Science & Engineering, MPGI School of Engineering, Nanded,
Maharashtra, India³

ABSTRACT: Cloud computing enables the users to remotely store their data in a server and provide services on-demand. Cloud storage is the best way to handle our information securely. Data security and privacy are the critical issues. There are many encryption technologies used to share data securely. One of the best ways is CP-ABE is to solve challenging problem of secure data sharing scheme in cloud computing. Only authorized users are able to encrypt and decrypt data. Each user has the set of attributes. The layered access structure are integrated into single access structure. The hierarchical files are encrypted with integrated access structure. Ciphertext time cost of encryption is saved. Final Implementation gives that linearly increasing the encryption and decryption time as number of attributes are increased. In this paper an efficient file hierarchy attribute based encryption scheme is proposed.

KEYWORDS: Cloud Computing, Data sharing, file hierarchy ciphertext policy, attribute based encryption

I. INTRODUCTION

Cloud Computing offers new ways to provide useful services on demand at a much cheaper make-up. The technology is ever developing and there are many cases of ongoing research to further improve this technology which inevitably will change the way businesses operate forever and provide many new opportunities for organizations alike [1].

Online data sharing has become a new "pet", such as Facebook, MySpace, and Badoo. Meanwhile, cloud computing [1]–[5] is one of the most promising application platforms to solve the explosive expanding of data sharing. In cloud computing, to protect data from leaking, users need to encrypt their data before being shared. In the cloud computing environment, the Cloud Storage Providers (CSPs) offer paid storage space on its infrastructure to store customers' data.

Here let us take the personal health record (PHR) for example [6]. To securely share the PHR information in cloud computing, a patient divides his PHR information M into two parts: personal information m_1 that may contain the patient's name, social security number, telephone number, home address, etc. The medical record m_2 which does not contain sensitive personal information, such as medical test results, treatment protocols, and operation notes. Then the patient adopts CP-ABE scheme to encrypt the information m_1 and m_2 by different access policies based on the actual need. For example, an attending physician needs to access both the patient's name and his medical record in order to make a diagnosis, and medical researcher only needs to access some medical test results for academic purpose in the related area, where a doctor must be a medical researcher, and the converse is not necessarily true.

The main purpose of attribute based encryption is to protect data from leaking, users need to encrypt their data before shared. An efficient algorithm is developed to encrypt and decrypt file. This algorithm extracts encryption time and decryption time. The scope of this project is to encrypt and decrypt the file and observe the respective time. This project can work as a base for future improvements in the field of attribute based encryption in cloud computing.



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirce.com

Vol. 6, Issue 3, March 2018

II. RELATED WORK

In [7] Sahai and Waters proposed fuzzy Identity-Based Encryption (IBE) in 2005, which was the prototype of ABE. Latterly, a variant of ABE named CP-ABE [8], [9], [10], [11] was proposed. In these schemes, a ciphertext is associated with an access policy and the user secret key is associated with a set of attributes. A secret key holder can decrypt the ciphertext if the attributes associated with his secret key satisfy the access policy associated with the ciphertext. In [12] author proposed the first notion of hierarchical encryption scheme, many hierarchical CP-ABE schemes have been proposed. For example in [13] author proposed a hierarchical ABE scheme by combining the hierarchical IBE [12] and CP-ABE. In [14] author proposed hierarchical ABE scheme. Later, in [15] author gave a hierarchical ABE scheme, while the length of secret key is linear with the order of the attribute set. A ciphertext policy hierarchical ABE scheme with short ciphertext is also studied in [16]. In [17] author proposed an online/offline ABE scheme to improve the speed of key generation and encryption, where each computation work in the two processes is split into two phases: offline phase (a preparation phase) and online phase.

III. BACKGROUND

A. Definitions

1. Access structure

Let $\{P_1, \dots, P_n\}$ be a set of parties. A collection $A \subseteq 2^{\{P_1, \dots, P_n\}}$ is monotone if $\square B, C: \text{if } B \subseteq A \text{ and } B \subseteq C \text{ then } C \subseteq A$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) A of nonempty subsets of $\{P_1, \dots, P_n\}$, i.e., $A \subseteq 2^{\{P_1, \dots, P_n\}} \setminus \{\emptyset\}$. The sets in A are called the authorized sets, otherwise, the sets are called the unauthorized sets.

2. Bilinear maps

Let G_0 and G_T be two groups of prime order p . The generator of G_0 is g . A bilinear mapping $e : G_0 \times G_0 \rightarrow G_T$ satisfies the following properties:

- Bilinearity: For any $u, v \in G_0$ and $a, b \in \mathbb{Z}_p$, it has $e(u^a, v^b) = e(u, v)^{ab}$
- Non-degeneracy: There exists $u, v \in G_0$ such that $e(u, v) \neq 1$.
- Computability: For all $u, v \in G_0$, there is an efficient computation $e(u, v)$.

3. DBDH Assumption

A challenger chooses a group G_0 of prime order p based on the security parameter of system. Let $a, b, c \in \mathbb{Z}_p$ be randomly chosen and g be a generator of G_0 . With (g, g^a, g^b, g^c) , the adversary must distinguish a valid tuple $e(g, g)^{abc} \in G_T$ from a random element $R \in G_T$. An algorithm B that outputs a guess $\mu \in \{0, 1\}$ has advantage ϵ in solving DBDH in G_0 if (1) was satisfied [31].

$$|\Pr[B(g, g^a, g^b, g^c, T = e(g, g)^{abc}) = 0] - \Pr[B(g, g^a, g^b, g^c, T = R) = 0]| \geq \epsilon$$

B. Description of the Proposed Algorithm:

The proposed algorithm is consists of four main steps. The FH-CP-ABE scheme consists of four operations: Setup, KeyGen, Encrypt and Decrypt. It is described as follows:

Step 1: $(PK, MSK) \leftarrow \text{Setup}(1^\kappa)$. The probabilistic operation takes a security parameter κ as input and outputs public key PK and master secret key MSK .

Step 2: $(SK) \leftarrow \text{KeyGen}(PK, MSK, S)$. The operation inputs PK, MSK and a set of attributes S and creates a secret key SK .

Step 3: $(CT) \leftarrow \text{Encrypt}(PK, ck, A)$. The operation inputs $PK, ck = \{ck_1, \dots, ck_k\}$ and a hierarchical access tree A as shown in the Fig. 2. At last, it creates an integrated ciphertext of content keys CT .

Step 4: $(ck_i (i \in [1, k])) \leftarrow \text{Decrypt}(PK, CT, SK)$. The algorithm inputs PK, CT which includes an integrated access structure A, SK described by a set of attributes S . If the S matches part of A , some content keys $ck_i (i \in [1, k])$ can be



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 6, Issue 3, March 2018

decrypted. If it matches the whole A, all the content keys can be decrypted. Then, the corresponding files m_i ($i \in [1, k]$) will be decrypted with the content keys by the symmetric decryption algorithm.

IV. PSEUDO CODE

Let $e : G_0 \times G_0 \rightarrow GT$ be a bilinear map, and G_0 be bilinear group of prime order p with generator g . For any $k \in \mathbb{Z}_p$ and an attribute set $S = \{S_1, S_2, \dots, S_m \in \mathbb{Z}_p\}$, the Lagrange coefficient $\Delta_{k,S} = \prod_{l \in S, l \neq k} (x - l) / (k - l)$. Two hash functions $H1: \{0,1\}^* \rightarrow$ and $H2 : \{0,1\}^* \rightarrow GT$ are used in the proposed scheme. An universe of attribute set is defined as $A = \{a_1, \dots, a_n\}$.

1. Setup(1^k). The authority runs the operation which inputs a security parameter κ and chooses random numbers $\alpha, \beta \in \mathbb{Z}_p$. It outputs PK and MSK as the formulas (2) and (3), respectively.

$$PK = \{G_0, g, h = g^\beta, e(g, g)^\alpha\} \quad (2)$$

$$MSK = \{g^\alpha, \beta\} \quad (3)$$

2. KeyGen(PK, MSK, S). The authority executes the algorithm which inputs a set of attributes $S (S \subseteq A)$ and creates a secret key SK about the set as the formula (4),

Where $r \in \mathbb{Z}_p$ and $r_j \in \mathbb{Z}_p$ are randomly chosen for each user and each attribute $j \in S$.

$$SK = D = g^\alpha \cdot h^r,$$

$$\forall j \in S : D_j = g^r \cdot H1(j)^{r_j}, D'_j = h^{r_j} \quad (4)$$

3. Assume that a data owner shares k files, i.e., $M = \{m_1, \dots, m_k\}$, with k access levels. Then, the corresponding content keys $ck = \{ck_1, \dots, ck_k\}$ are encrypted as the following Encrypt operation. Encrypt(PK, ck, T). The public key PK, content keys $ck = \{ck_1, \dots, ck_k\}$, and a hierarchical access tree T are taken as input. The algorithm outputs an integrated ciphertext CT.

- Data owner sets level nodes $(x_i, y_i) (i = 1, 2, \dots, k)$ in T, and selects k random numbers s_1, \dots, s_k in \mathbb{Z}_p .

Then, it computes \tilde{C}_i and C'_i for all $i = 1, 2, \dots, k$ as the formula (5).

$$\tilde{C}_i = ck_i e(g, g)^{as_i}, C'_i = g^{s_i} \quad (5)$$

Then, data owner computes $C(x, y)$ and $C'(x, y)$ for all nodes (x, y) in the set of Y as the formulas (6)

And (7)

$$C(x, y) = h^q_{(x,y)}(0) \quad (6)$$

$$C'(x, y) = H1(att(x, y))^q_{(x,y)}(0) \quad (7)$$

- In T, let X be the set of transport nodes, and $TN-CT(x, y)$ be the threshold gate set of transport node (x, y) 's children, where $TN-CT(x, y) = \{child_1, \dots, child_j, \dots\}$. Then, data owner computes $\hat{C}(x, y)_j$ for each node (x, y) in the set of X and all $j = 1, 2, \dots$ as the formula (8).

$$\hat{C}(x, y)_j = e(g, g)^{a \cdot q(x, y)(0) + q_{child_j}(0)}$$

$$H2(e(g, g)^{aq(x, y)(0)}) \quad (8)$$

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 6, Issue 3, March 2018

- Data owner outputs the integrated ciphertext CT as the formula (9).

$$CT = \{ T, C_i^-, C_i^+, C(x,y), C^-(x,y), C^+(x,y), j \} \quad (9)$$

4. **Decrypt(PK, CT, SK).** A user needs the public key PK and SK described by S to decrypt CT.

- If (x, y) is a leaf node, we let $i = \text{att}(x, y)$ and define $\text{DecryptNode}(CT, SK, (x, y))$ as below. If $i \notin S$, $\text{DecryptNode}(CT, SK, (x, y)) = \text{null}$. Otherwise, the operation $\text{DecryptNode}(CT, SK, (x, y))$ is obtained by the formula (10).

$$\begin{aligned} \text{DecryptNode}(CT, SK, (x, y)) &= e(D_i, C(x,y)) / e(D_i, C^-(x,y)) \\ &= e(g^r H1(i)^{r_i}, h^{q(x,y)(0)}) / e(h^{r_i}, H1(\text{att}(x, y)^{q(x,y)(0)})) \\ &= e(g, g)^{r\beta q(x,y)(0)} \end{aligned} \quad (10)$$

- If (x, y) is a non-leaf node, $\text{DecryptNode}(CT, SK, (x, y))$ is defined as below. For all nodes z that are children of (x, y), it runs $\text{DecryptNode}(CT, SK, z)$ and stores the output as F_z . Let $S(x,y)$ be an arbitrary $k(x,y)$ - sized child nodes set z, and then $F_z = \text{null}$. If the set does not exist, $F_z = \text{null}$. Otherwise, $F(x,y)$ is computed as the formula (11),

where $S(x,y) = \{\text{index}(z) : z \in S(x,y)\}$, $i = \text{index}(z)$.

$$\begin{aligned} F_{(x,y)} &= \prod_{z \in S(x,y)} F_z^{\Delta_{i,S'(x,y)}(0)} \\ &= \prod_{z \in S(x,y)} (e(g, g)^{r \cdot \beta q_z(0)})^{\Delta_{i,S'(x,y)}(0)} \\ &= \prod_{z \in S(x,y)} (e(g, g)^{r \cdot \beta q(x,y)(i)})^{\Delta_{i,S'(x,y)}(0)} \\ &= e(g, g)^{r \cdot \beta q(x,y)(0)} \\ &= e(g, g)^{r \cdot \beta q(x,y)(0)} \end{aligned} \quad (11)$$

Then, the procedures of decryption algorithm are described as follows:

- If the attribute set S satisfies part or the whole T, that is, S satisfies part or the whole level nodes, $e(g, g)^{r\beta si}$ ($i \in [1, k]$) can be obtained by the recursive operation of the formula (12).

$$\begin{aligned} A_i &= \text{DecryptNode}(CT, SK, (x_i, y_i)) \\ &= e(g, g)^{r\beta q(x_i, y_i)(0)} \\ &= e(g, g)^{r\beta si} \quad (i \in [1, k]) \end{aligned} \quad (12)$$

- Next, $e(g, g)^{asi}$ can be computed by the formula (13).

$$\begin{aligned} F_i &= e(C_i^-, D) / A_i \\ &= e(g^{si}, g^\alpha \cdot g^{\beta r}) / e(g, g)^{r\beta si} \end{aligned}$$



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 6, Issue 3, March 2018

$$= e(g,g)^{asi} \quad (i \in [1,k]) \tag{13}$$

- Based on the hierarchical nodes, if S includes the lower authorization nodes, we can recursively calculate all of the authorization's level nodes with the values of transport nodes $\hat{C}(x,y,j)$ ($j = 1,2,\dots$) by using the formula (14). Therefore, $F_{(i+1),j}, \dots, F_{k,j}$ are obtained sequentially. That is, the values $e(g,g)^{asi}$, $e(g,g)^{asi+1}, \dots, e(g,g)^{ask}$ are got.

$$\begin{aligned}
 F_{(i+1),j} &= \frac{\hat{C}_{(x_i,y_i),j}}{F_i \cdot H_2(F_i)} \\
 &= \frac{e(g,g)^{\alpha(s_i+q_{child_j}^{(0)})} \cdot H_2(e(g,g)^{\alpha s_i})}{e(g,g)^{\alpha s_i} \cdot H_2(e(g,g)^{\alpha s_i})} \\
 &= e(g,g)^{\alpha q_{child_j}^{(0)}} \quad (j = 1, 2, \dots) \tag{14}
 \end{aligned}$$

- Then, the corresponding content keys $\{ck_i, \dots, ck_k\}$ are decrypted by executing the formula (15) repeatedly.

$$\frac{\tilde{C}_i}{F_i} = \frac{ck_i e(g,g)^{\alpha s_i}}{e(g,g)^{\alpha s_i}} = ck_i \quad (i \in [1, k]) \tag{15}$$

- At last, the authorized files $\{m_i, \dots, m_k\}$ are decrypted with $\{ck_i, \dots, ck_k\}$, using symmetric decryption algorithm.

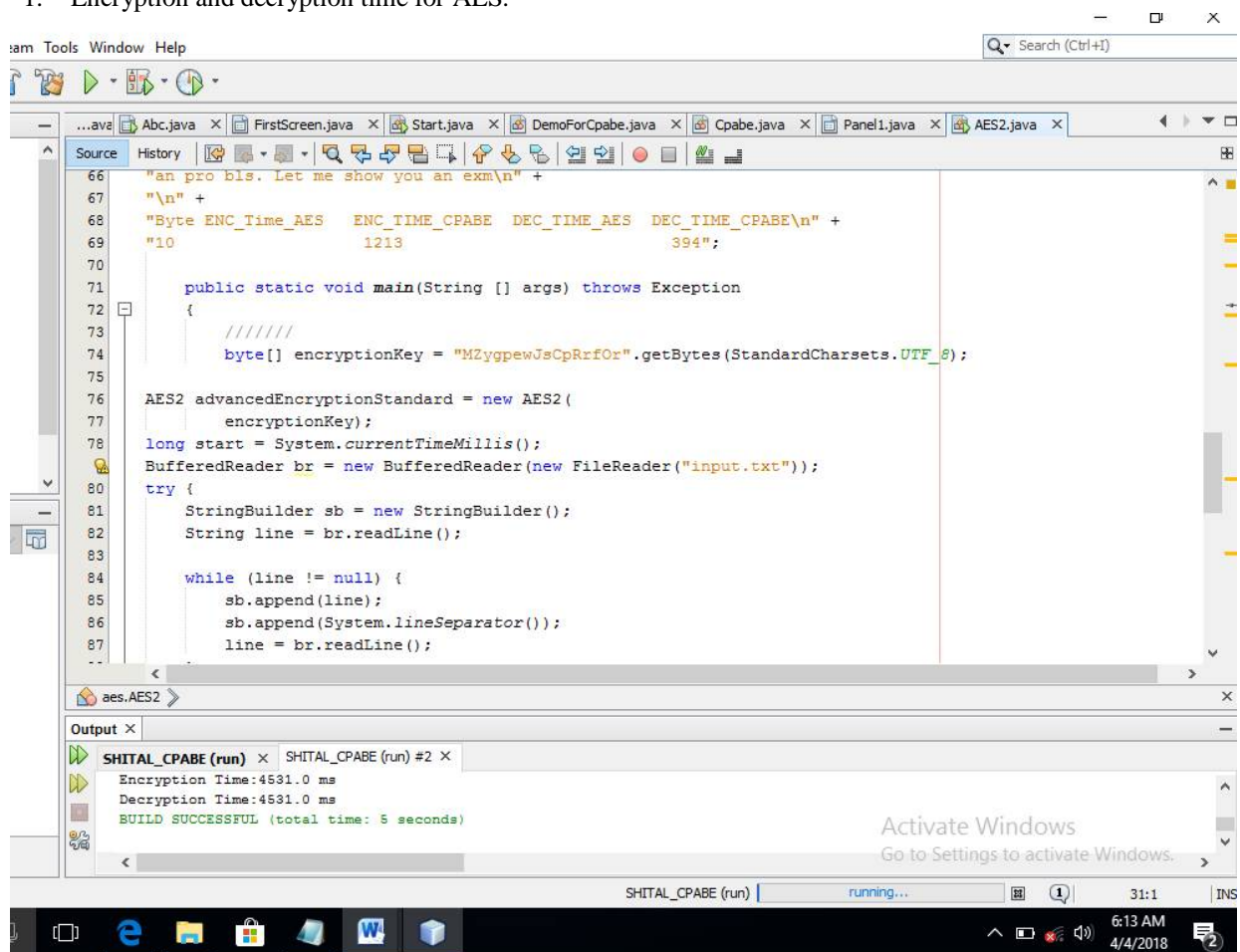
V. SIMULATION RESULTS

To validate theoretical analysis presented in previous subsection, we implement FH-CP-ABE scheme based on the cpabe toolkit and the Java Pairing-Based Cryptography library (JPBC)[18]. The implementation uses a 160-bit elliptic curve group based on the super singular curve $y^2 = x^3 + x$ over a 512-bit finite field. Meanwhile, to compare experimental results of the encryption and decryption, we also simulate the typical CP-ABE system. In addition, the following experiments are conducted by using Java on the system with Intel Core processor at 2.79 GHz and 1.96GB RAM running Windows XP SP 3. And all of the results are averages of 10 trials. In the simulation, the FH-CP-ABE scheme's implementation adopts the improved encryption algorithm in encryption operation. In a CP-ABE scheme[14], the complexity of access policy associated with ciphertext impacts two aspects. The one is the time cost of encryption and decryption. The other is the storage cost of ciphertext. To illustrate this, we assume that a patient sets his PHR's access policy with k access levels in the form of $\{(a_1, a_2, \dots, a_i, i \text{ of } i) \text{ AND } a_{i+1} \dots \text{ AND } a_N\}$ (i.e., the worst situation over the policy), where each a_i ($i \in [1, N]$) denotes an attribute. Meanwhile, the patient generates k policies based the above form for using in CP-ABE scheme. For example, assume that the patient shares three files, i.e., $M = \{m_1, m_2, m_3\}$, with three access levels, the access policy is designed as $\{(a_1, a_2, \dots, a_i, i \text{ of } i) \text{ AND } a_{i+1} \text{ AND } a_{i+2}\}$ in FH-CP-ABE scheme. Accordingly, he should construct three access policies for CP-ABE scheme, where the policies are $\{(a_1, a_2, \dots, a_i, i \text{ of } i) \text{ AND } a_{i+1} \text{ AND } a_{i+2}\}$, $\{(a_1, a_2, \dots, a_i, i \text{ of } i) \text{ AND } a_{i+1}\}$, and $\{a_1, a_2, \dots, a_i, i \text{ of } i\}$. The policies only contain AND gate to ensure that all the ciphertext components are computed in decryption algorithm.

When this project is implemented in netbeans it shows encryption time and decryption time for existing and proposed scheme. The following snapshot shows the encryption time and decryption time for existing scheme and proposed scheme.

A. Snapshots

1. Encryption and decryption time for AES.



```
66 "an pro bls. Let me show you an exm\n" +
67 "\n" +
68 "Byte ENC_Time_AES ENC_TIME_CPABE DEC_TIME_AES DEC_TIME_CPABE\n" +
69 "10 1219 394";
70
71 public static void main(String [] args) throws Exception
72 {
73     //
74     byte[] encryptionKey = "MZygpewJsCpRrfOr".getBytes(StandardCharsets.UTF_8);
75
76     AES2 advancedEncryptionStandard = new AES2(
77         encryptionKey);
78     long start = System.currentTimeMillis();
79     BufferedReader br = new BufferedReader(new FileReader("input.txt"));
80     try {
81         StringBuilder sb = new StringBuilder();
82         String line = br.readLine();
83
84         while (line != null) {
85             sb.append(line);
86             sb.append(System.lineSeparator());
87             line = br.readLine();
88         }
89     } catch (IOException e) {
90         e.printStackTrace();
91     }
92     br.close();
93     String encryptedText = sb.toString();
94     String decryptedText = advancedEncryptionStandard.decrypt(encryptedText);
95     System.out.println("Decrypted Text: " + decryptedText);
96 }
```

Output

```
SHITAL_CPABE (run) x SHITAL_CPABE (run) #2 x
Encryption Time:4531.0 ms
Decryption Time:4531.0 ms
BUILD SUCCESSFUL (total time: 5 seconds)
```

Fig1: Pseudo code for encryption and decryption time for AES

Above figure shows the encryption and decryption time for existing scheme. In this program advanced Encryption Standards algorithm is used, which is 128-bit. After run this program , input.txt file encrypted and it requires encryption time in millisecond. It decrypted in millisecond. It will take any type of file for example PNG, PDF, etc.

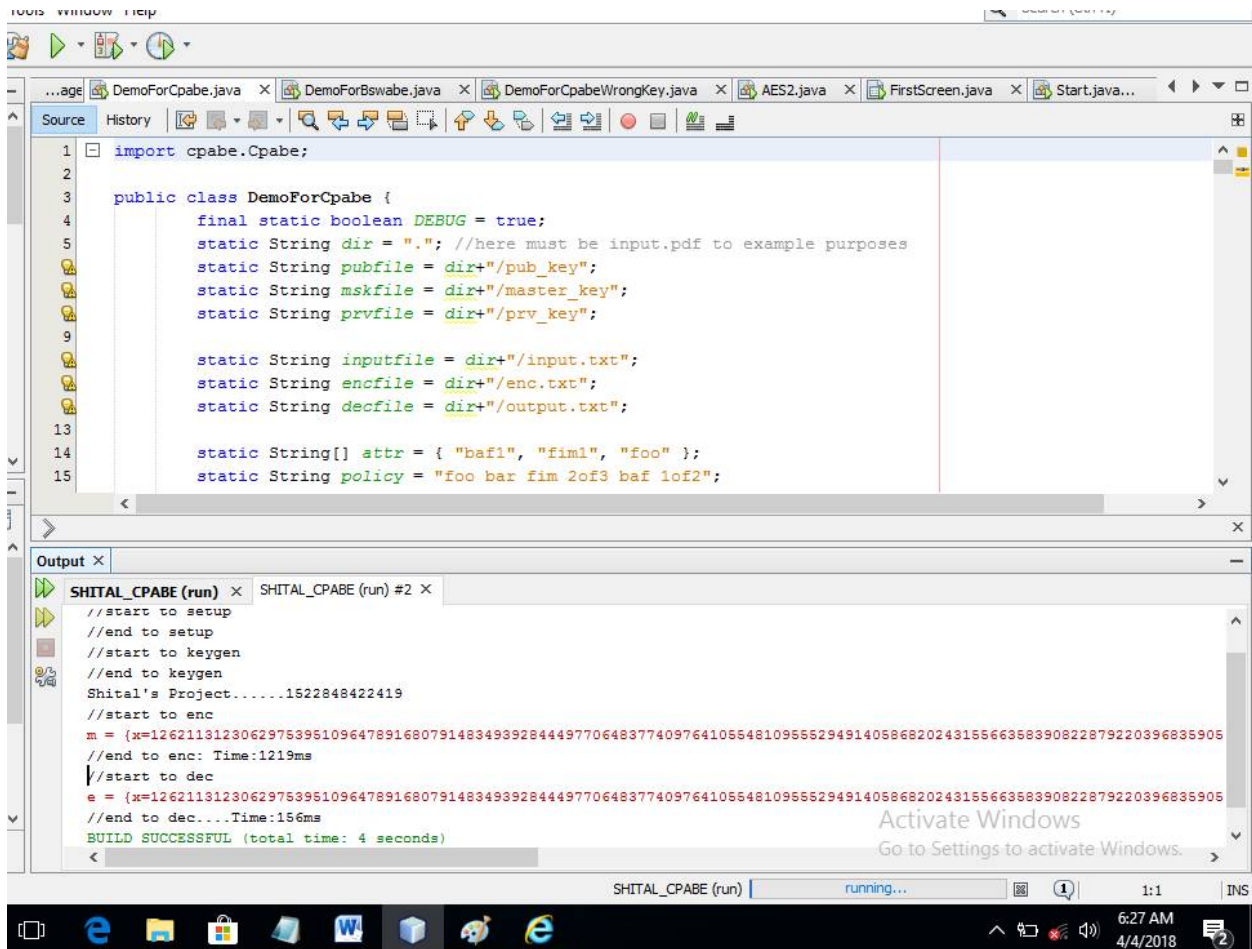
International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 6, Issue 3, March 2018

2. Encryption and decryption time for CP-ABE



```
1 import cpabe.Cpabe;
2
3 public class DemoForCpabe {
4     final static boolean DEBUG = true;
5     static String dir = "."; //here must be input.pdf to example purposes
6     static String pubfile = dir+"/pub_key";
7     static String mskfile = dir+"/master_key";
8     static String prvfile = dir+"/prv_key";
9
10
11     static String inputfile = dir+"/input.txt";
12     static String encfile = dir+"/enc.txt";
13     static String decfile = dir+"/output.txt";
14
15     static String[] attr = { "ba1", "fim1", "foo" };
16     static String policy = "foo bar fim 2of3 ba1 1of2";
17 }
```

```
SHITAL_CPABE (run) x SHITAL_CPABE (run) #2 x
//start to setup
//end to setup
//start to keygen
//end to keygen
Shital's Project.....1522848422419
//start to enc
m = {x=12621131230629753951096478916807914834939284449770648377409764105548109555294914058682024315566358390822879220396835905
//end to enc...Time:1219ms
//start to dec
e = {x=12621131230629753951096478916807914834939284449770648377409764105548109555294914058682024315566358390822879220396835905
//end to dec...Time:156ms
BUILD SUCCESSFUL (total time: 4 seconds)
```

Fig2: Pseudo code for encryption and decryption time for CP-ABE

Above figure shows the encryption and decryption time for proposed scheme. In this program Ciphertext-Policy Attribute Based algorithm is used, which requires policy chain. After run this program, input.txt file encrypted and it requires encryption time in millisecond. It decrypted in millisecond. It will take any type of file for example PNG, PDF, etc.

B. Graphs

After implementation of this project encryption and decryption time is calculated from that result the following graph of existing scheme is drawn

X-axis: No. of bytes

Y axis: Encryption time , Decryption time

From encryption time and decryption time ,the graph is drawn and which is linearly increasing.

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 6, Issue 3, March 2018

1. Graph for existing scheme

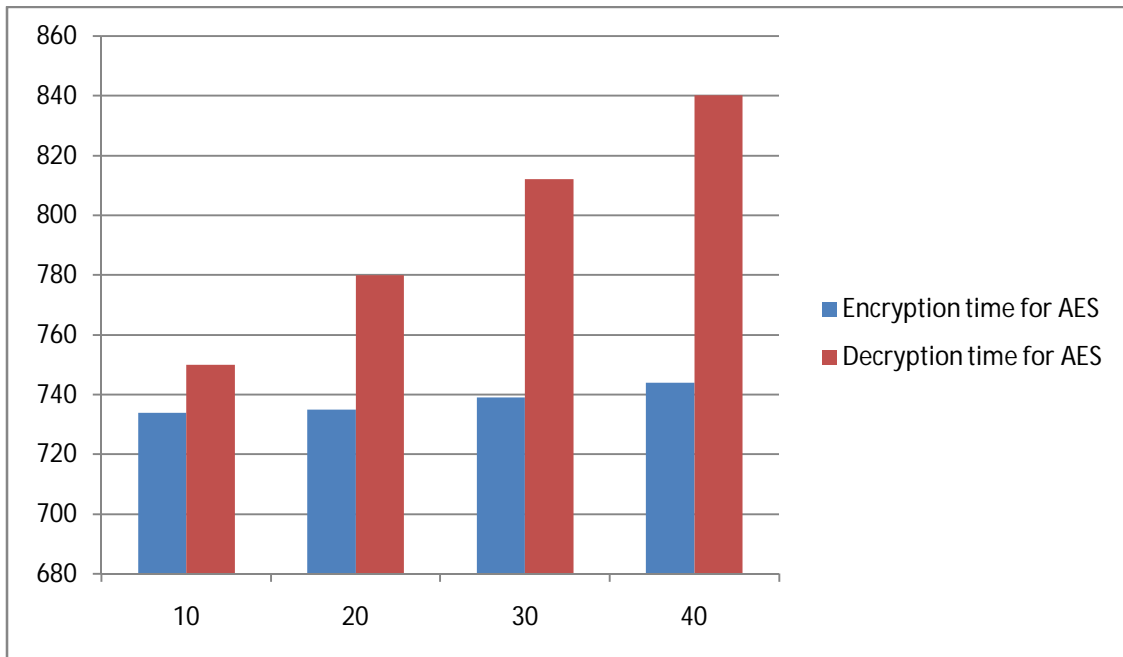


Fig.3: Graph for encryption and decryption time for existing scheme

2. Graph for proposed scheme

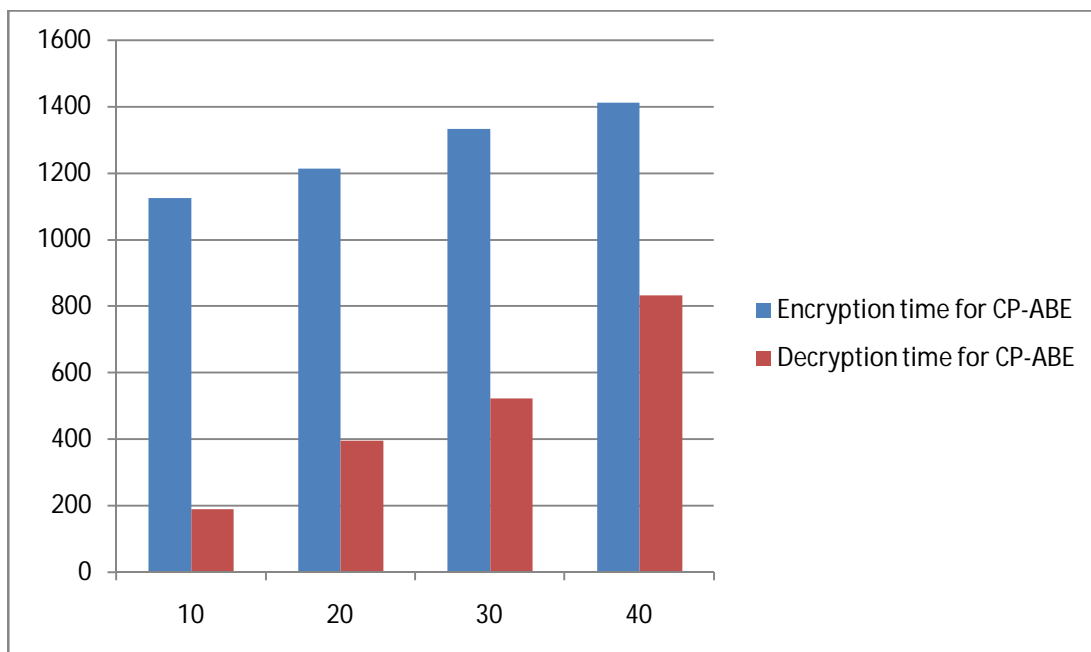


Fig.4: Graph for encryption time and decryption time for proposed scheme



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 6, Issue 3, March 2018

Above graphs proves, after implantation of this code, we can find that the proposed scheme improves the efficiencies of encryption and decryption greatly when two hierarchy files are shared. We can also find that the results are gradually increasing and approximately following a linear relationship with the number of attributes. When the number of files is fixed, the more the number of attributes is used, the more time cost of encryption and decryption in FH-CP-ABE scheme is saved.

VI. CONCLUSION AND FUTURE WORK

In this paper, the proposed scheme gives the encryption and decryption time which is linearly increasing as the number of attributes are increasing. Therefore, both ciphertext storage and time cost of encryption are saved. The proposed scheme has an advantage that users can decrypt all authorization files by computing secret key once. Thus, the time cost of decryption is also saved if the user needs to decrypt multiple files. CP-ABE an adjustment of Attribute Based Encryption (ABE) for the reasons for giving certifications towards the provenance the delicate information Our scheme also enables dynamic modification of access policies o supports efficient on-demand user/attribute revocation and break-glass access under emergency scenarios. From the survey we understand that some amount of work has been done in the field of cloud computing for several security issues. It can be applied to achieve scalable, flexible, security, privacy, data confidentiality and fine-grained access control of outsourced data in cloud computing. There is more scope for future research in the field of secure data sharing in the cloud. We proposed a scheme for efficient identity-based user revocation in multi-authority CP-ABE. In the future, our work can be continued in several directions. Securely forwarding the revocation related computations to the CSP (or even to the user). The security of our construction is proved in the generic bilinear group model, although we believe it would be possible to achieve full security by adapting the dual system encryption methodology. This type of work would be interesting even if it resulted in a moderate loss of efficiency from our existing system.

REFERENCES

1. C.-K. Chu, W.-T. Zhu, J. Han, J.-K. Liu, J. Xu, and J. Zhou, "Security concerns in popular cloud storage services," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 50–57, Oct./Dec. 2013.
2. Minu George¹, Dr. C.Suresh Gnanadhas², Saranya.K3, "A Survey on Attribute Based Encryption Scheme in Cloud Computing," *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, Issue 11, November 2013
3. K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloudbased revocable identity-based proxy re-encryption scheme for public clouds data sharing," in *Proc. 19th Eur. Symp. Res. Comput. Secur.*, vol. 8712, Sep. 2014, pp. 257–272.
4. V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Proc. 35th Int. Colloq. Automata, Lang. Program.*, vol. 5126, Jul. 2008, pp. 579–591.
5. Bethencourt, J.; Sahai, A.; Waters, B. (2007-05-01). "Ciphertext-Policy Attribute-Based Encryption". *2007 IEEE Symposium on Security and Privacy (SP '07)*: 321–334. doi:10.1109/SP.2007.11
6. F. Xhafa, J. Wang, X. Chen, J. K. Liu, J. Li, and P. Krause, "An efficient PHR service system supporting fuzzy keyword search and fine-grained access control," *Soft Comput.*, vol. 18, no. 9, pp. 1795–1802, Sep. 2014.
7. A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology*. Berlin, Germany: Springer, May 2005, pp. 457–473.
8. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in *Proc. IEEE Symp. Secur. Privacy*, May 2007, pp. 321–334.
9. L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, Oct. 2007, pp. 456–465.
10. A. Balu and K. Kuppasamy, "An expressive and provably secure ciphertext-policy attribute-based encryption," *Inf. Sci.*, vol. 276, pp. 354–362, Aug. 2014.
11. B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. 14th Int. Conf. Pract. Theory Public Key Cryptogr. (PKC)*, vol. 6571, Mar. 2011, pp. 53–70.
12. C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *Advances in Cryptology*. Berlin, Germany: Springer, Dec. 2002, pp. 548–566.
13. G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, Oct. 2010, pp. 735–737.
14. Z. Wan, J. Liu, and R. H. Deng, "HASBE: A hierarchical attributebased solution for flexible and scalable access control in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 743–754, Apr. 2012.
15. X. Zou, "A hierarchical attribute-based encryption scheme," *Wuhan Univ. J. Natural Sci.*, vol. 18, no. 3, pp. 259–264, Jun. 2013.
16. H. Deng et al., "Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts," *Inf. Sci.*, vol. 275, pp. 370–384, Aug. 2014.
17. S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Proc. 17th Int. Conf. Pract. Theory Public-Key Cryptogr. (PKC)*, vol. 8383, Mar. 2014, pp. 293–310.
18. A. De Caro and V. Iovino, "JPBC: Java pairing based cryptography," in *Proc. IEEE Symp. Comput. Commun.*, Jun./Jul. 2011, pp. 850–855.