Key Policy Attribute Based Encryption in Cloud Storage

Sonali Deshpande[1], Dipa Dharmadhikari[2]

[1, 2]CSE, BAMU, DIEMS, Maharashtra 431001, India

Abstract: ***The Key Policy Attribute Based Encryption provides how to maintain integrity, confidentiality and availability to share data with others. It generates encryption, decryption time along with file size and file type. Proposed work explains encryption with Attribute Based encryption then file is spitted into chunks. Hash key is generated for every data chunk which is unique. Secure hash algorithm is used for generation of hash key. Proposed algorithm gives confidentiality to work as it is a public key encryption algorithm of cryptography. It is also asymmetric i.e. key set is used for data encryption and decryption. Data is stored on cloud means it is available to everyone through Cloud space. Key generated for encryption and decryption is runtime. Key policy attribute based encryption reduces encryption time by 35% and decryption time by 65%.***

Keywords: KAC, Key policy attribute based encryption, Integrity, Confidentiality, Availability

1. Introduction

Cloud computing means storing and accessing data and programs over the Internet instead of your computer's hard drive. The cloud is just a metaphor for the Internet. It goes back to the days of flowcharts and presentations that would represent the gigantic server-farm infrastructure of the Internet as nothing but a puffy, white cumulus cloud, accepting connections and doling out information as it floats [1].

For the business in the cloud, they have following requirements: User can upload data from client side. Uploaded data can be encrypted in server side along with algorithm. Encryption time is calculated with respect to file size. Decryption time is calculated with respect to file size .File after encryption is stored on cloud which is available to all.

The objective of developing proposed system is it calculates encryption and decryption time of file type with respect to file size. The proposed system is developed for the following reason. It is more secure for data encryption and decryption .It is along with secure hashing algorithm gives integrity to system. Key Policy Attribute Based Encryption calculates time for encryption and decryption on different files. Key Policy Attribute Based Encryption makes data available on cloud. Key Policy Attribute Based Encryption gives confidentiality as it is an encryption technique of cryptography.

2. Related Work

2.1 Identity Based Encryption

Guoel tried to build IBE with key aggregation. One of their schemes assumes random oracles but another does not. In their schemes, key aggregation is constrained in the sense that all keys to be aggregated must come from different “identity divisions”. While there are an exponential number of identities and thus secret keys, only a polynomial number of them can be aggregated. Most importantly, their key-aggregation comes at the expense of $O(n)$ sizes for both cipher texts and the public parameter, where n is the number of secret keys which can be aggregated into a constant size one. This greatly increases the costs of storing and transmitting cipher texts, which is impractical in many situations such as shared cloud storage.

2.2 Key Aggregate Cryptosystem

Cloud storage is gaining popularity recently. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Nowadays, it is easy to apply for free accounts for email, photo album, and file sharing and/or remote access, with storage size more than 25GB (or a few dollars for more than 1TB). Together with the current wireless technology, users can access almost all of their files and emails by a mobile phone in any corner of the world. Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication, which means any unexpected privilege escalation will expose all data. In a shared tenancy Cloud Computing environment, things become even worse.

Data from different clients can be hosted on separate virtual machines but reside on a single physical machine. Data in target virtual machines could be stolen by instantiating another virtual machines co-resident with the target one. Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third party auditor to check the availability of files on behalf of the data owner without leaking anything about the data, or without compromising the data owner’s anonymity. Likewise, Cloud users probably will not hold the strong belief that the Cloud server is doing a good job in terms of confidentiality. Fig 1 describes KAC.

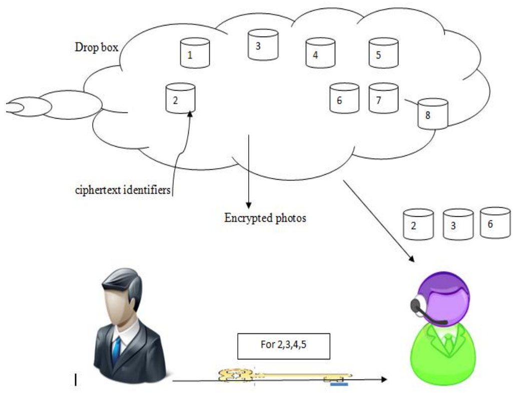

Figure 1: Key Aggregate Cryptosystem

3. Proposed System

In key policy Attribute Based Encryption first user upload data from client side then that uploaded file is stored on server side. On server side that respective file is encrypted by using attribute based encryption with public key. After encryption that file get stored on cloud in four chunks. At the time of decryption that particular file is downloaded then merged, decrypted in client side with private key which is generated runtime. Decryption is done in the client side. Below figure describes whole architecture of key policy attribute based encryption. Fig 2 decribes proposed system diagram.

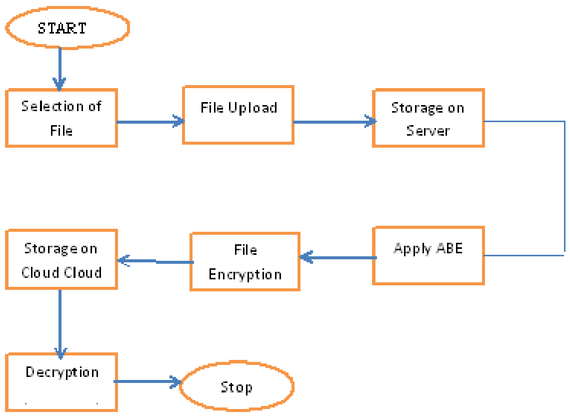

Figure 2: Proposed System Architecture

Key Policy Attribute Based Encryption, encryption scheme is divided into three modules Client module, Server module, Storage module. In client side data is uploaded, In server side data is encrypted, In storage side data is stored. All these steps described in brief below.

- **Selection of File:**User can select any file such as pdf, ppt, excel, mp3, video, image for uploading.
- **File Upload:** File is uploaded from client side.
- **Storage on server:** File is stored on server after uploading. on server side algorithm is selected for encryption.
- **Apply ABE algorithm:** Algorithm ABE is selected for encryption with public key.
- **File encryption:** File is encrypted with algorithm means file is taken afterword's it is converted into binary form then particular algorithm is applied with specific key.
- **Storage on cloud:** After encryption with proposed algorithm key policy attributes based encryption file is stored on cloud in equal chunks.
- **File decryption**: File is decrypted from client side firstly it is downloaded then merged into single file and afterword's with private key it is decrypted.

4. System Modules

1) **Client Side**: In client side user is going to login, if new one then first registration will be done and then file is uploaded then user will logout. After uploading file is redirected to server side for encryption then for decryption it is downloaded then merged and decrypted from client side again. keys for encryption and decryption are different separate. Fig 3 describes client side.

- Client side has following parts.
  a) Login
  b) Register
  c) File upload
  d) Logout

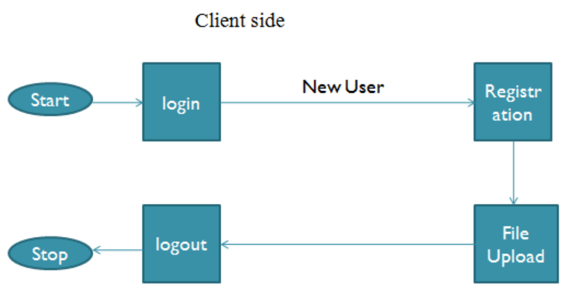

Figure 3: Client Side

2) **Server Side:** In server side there are two parts such as master and analysis. In master part user has again two parts file upload on server side and users data. In analysis crystal report is generated as per algorithm file type, average time is also calculated. File is encrypted with attribute based encryption then after encryption data is stored on cloud. Fig 4 describes Server side.

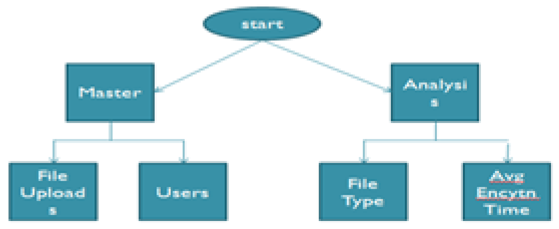

Figure 4: Server Side

3) **Storage Side:** In storage side, the data is stored on cloud after encryption in four chunks then for decryption the encrypted file is searched on cloud and then decrypted. Fig 5 describes storage side.

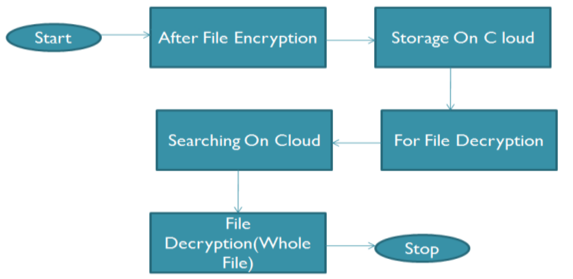

Figure 5: Storage Side

5. Proposed System Framework

In key policy attribute based encryption there are five steps global set up, authority set up, keygen, encryption, and decryption. In global set up security level parameter l is generated. In authority set up authority is given. In keygen public key is generated which is used for encryption, after encryption data is spitted in four chunks by using secure hash algorithm it is assigned a hash key for every data chunk which is unique and then for decryption that particular file is searched on cloud then it is downloaded then merged, decrypted using private key.

System Model

1. Global Setup (1^l) $\rightarrow$ params. This algorithm takes as input a security parameter *l* and outputs the system parameters params.
2. Authority Setup (1^l) $\rightarrow$ *(SK_i, PK_i, Ai)*.Each authority Ai generates his secret-public key pair KG *(1^l)* $\rightarrow$ *(SKi; PKi)* and an access structure *Ai*, for i=1, 2; . . .N.
3. Keygen (*Ski,GID,Ai* $_{GID}$) *Sk_u^i* Each authority Ai takes as input his secret key SKi, a global identifier GID and a set of attributes *Ai* GID, and outputs the secret keys *SKiU*, where $Ai_{GID} = A_{GID} \cap A_i$,A_{GID} and *Ai* denote the attributes corresponding to the GID and monitored by Ai, respectively.
4. Encryption (params, *M*, *Ac*). This algorithm takes as input the system parameters params, a message M and a set of attributes *AC*, and outputs the cipher text CT, where $A_c=\{ A_c^1,A_c^2.....A_c^N\}$ $\tilde{}A_c^i \cap A_i$.
5. Decryption: This algorithm takes as input a GID the secret keys cipher text CT and outputs the message M, where I_c is the index set of the authorities A_i such that $A^i_c \neq \{ \}$

Steps of algorithm
Step I: Initialization ()

```
1) Procedure Initialization ( )
2) Initialize cipher index classes with its file size
3) Generate public key by using list of cipher index class,
   generate random index
4) for each i=0 where i<bytes1.length
5) String j= cipher index class name + random (i);
6) String str= Integer.toBinaryString (j);
7) increment i;
8) end for
9) end Procedure
```

Step II Key generation

```
1) Procedure Summation Keygen ( )
2) Masking public key & byte format
3) for each i=0 upto i<byptes12.lenght
4) int j=bytes12 [i];
5) String s3=Integer.toBinaryString ( j );
6) String temp= temp + Integer.parseInt (s3);
7) S3=toBinaryString (temp);
8) end for
9) end Procedure
```

Step III: Encryption of file

```
1) Procedure Encryption (b)
2) key initialization
3) Takes whole file as msg
4) FOS=new FileoutputStream (out)
5) byte [] b=new byte [8];
6) int i-cis.read (b);
7) while i! = -1 do
8) fos.write (b, 0, i);
9) i=cis.read (b);
10)end while
11)end Procedure 37
```

Step IV: Decryption of file

```
1)  Procedure Decryption (b)
2)  Encrypt. nit (cipher.DecryptMode, Secret key);
3)  cis=new fileOutputStream (fis, encrypt);
4)  fos=new fileOutputStream (dec);
5)  byte [ ] b=new byte [8];
6)  inti=cis.read (b);
7)  while i! =-1 do
8)  fos.write (b, 0, i);
9)  i=cis.read (b);
10) end while
11) end Procedure
```

Steps of SHA-512

1) Initial hash values are set by considering fractional parts of first eight prime numbers.
2) Message is padded into blocks based on size it should be a multiple of 1024 bits.
3) Parse the message into 1024 bit blocks.
4) SHA-512 compression function is used to update registers such as a, b, c, d,e.
5) Number of rounds used are 80.
6) Expandable message block is generated via SHA-512 message schedule.
7) Words are created which is a fractional cube root of first eighty prime numbers.
8) Hash keys are words created over their 16 bit long.
9) Hash key is nothing but a hexadecimal number created by SHA-512.

How SHA-512 is incorporated in proposed system?

In key policy attribute based encryption, data is encrypted after words divided into chunks and stored on cloud. Stored data is having unique hash key for every data chunk generated by SHA-512.hash key is a hexadecimal number generated by SHA-512.

6. Result and Discussion

Following attributes are taken for comparison.

- **File Type**
  File type such as word, pdf, ppt, mp3, video, image are taken for comparison. on file type user can compute

encryption, decryption time. file is selected from client side then it is redirected to server side.

- **File Size**
  File size such as 2MB pdf, 2MB word document, 2MB mp3, 2MB video selected and as per proposed algorithm encryption and decryption time is calculated. user can select file up to 5 MB.
- **Encryption Time**
  Encryption time is time required to encrypt that particular file. Encryption time required for such type of files pdf, ppt, word, mp3, video is calculated as per Identity based encryption and then with proposed key policy attribute based encryption. Time is different for both algorithms and then both systems are compared.
- **Decryption Time**
  Encryption time required for such type of files pdf, ppt, word, mp3, and video is calculated as per Identity based encryption and proposed attribute based encryption then both Systems are compared. In table 1 all values are given file type, file size, encryption time, decryption time algorithms.

Table 1: File type; file size, encryption time, decryption time

File Type	File size	ABE		IBE	
		Encr time	Decr time	Encr time	Decr time
pdf File	3MB	434	361	640	655
ppt File	3MB	391	489	680	500
Mp3 File	3MB	86	76	800	753

Evaluation of Algorithms
The following graphs shows file type and average encryption and decryption time needed to calculate. Both with IBE, ABE graphs are generated.

Evaluation of IBE
Graphical representation of file type for IBE describes time required for encryption and decryption time by using identity based encryption. In the following graph X-axis represents file type for 3 MB data and Y-axis represents time in seconds It shows Pdf of 3MB and encryption and decryption time required by it. table 1 describes evaluation of algorithms.

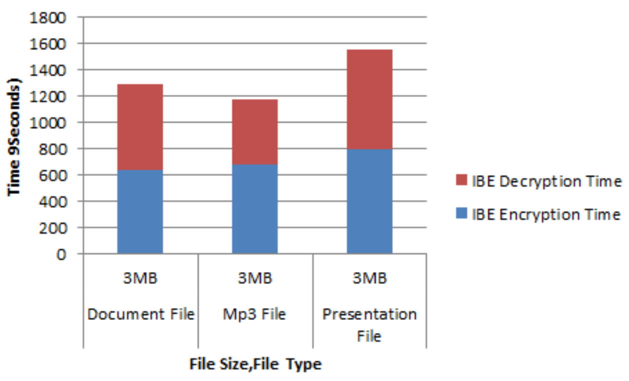

Figure 6: Evaluation of IBE

Evaluation of ABE
Graphical representation of file type for ABE describes time required for encryption and decryption time by using attribute based encryption. In following graph X-axis represents file type for 3 MB data and Y-axis represents time in seconds.

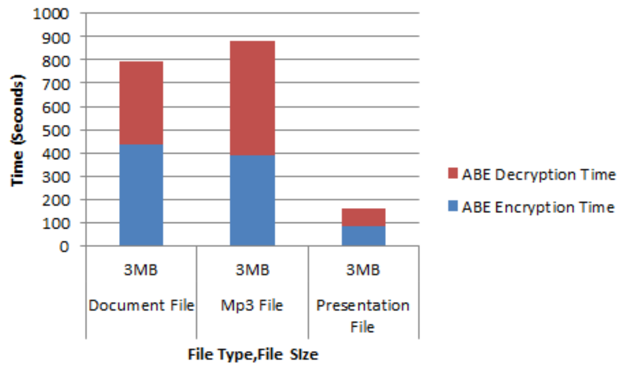

Figure 7: Evaluation of ABE

Analysis of Portable Document Format File
Table gives information about file type, file size, encryption and decryption time required by both algorithms identity based encryption and key policy attribute based encryption it calculates encryption time with respect to file size, file type. Table 2 describes analysis of portable document file format with IBE and ABE.

Table 2: Analysis of Portable Document Format File

File Type	File size	ABE		IBE	
		Encr time	Decr time	Encr time	Decr time
pdf file	6MB	990	1645	987	1089
pdf file	5MB	320	478	879	123
pdf file	3MB	670	567	234	765
pdf file	2 MB	261	458	340	290

Analysis of IBE on Portable Document Format File
The following graph gives user information about file type along with file size and encryption, decryption time required by Identity based encryption algorithm. analysis gives time with file size like 6 MB,5MB ,3MB,2MB and so on .Fig 8 describes IBE.

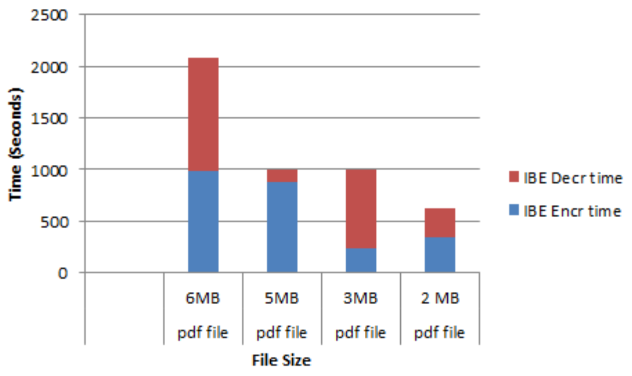

Figure 8: Analysis of IBE on Portable Document Format File

Analysis of ABE on Portable Document Format File
The following graph gives user information about file type along with file size and encryption, decryption time required by Attribute based encryption algorithm. Fig 9 describes ABE on portable document file format.

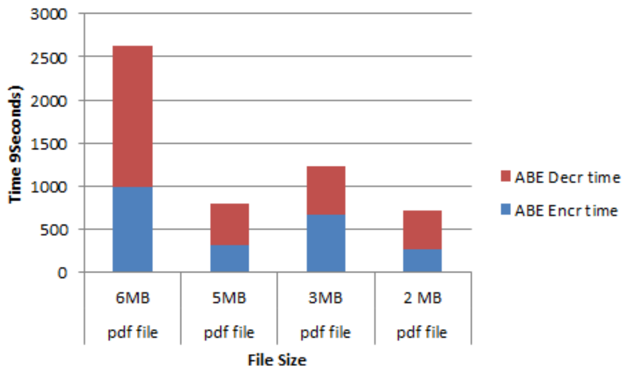

Figure 9: Analysis of ABE on Portable Document Format File

Integrity
Data integrity is defined as the accuracy and consistency of stored data, in absence of any alteration to the data between two updates of a file or record. Cloud services should ensure Integrity is achieved through Secure Hash Algorithm 512.suppose user is sending ppt file then it is as it after encryption. SHA 512 divides the file into equal chunks. for every data chunk hash key is generated which is a hexadecimal number. Hexadecimal number is going to be unique for every data chunk.ch. So SHA shows file is in the same form as it was before encryption.

Confidentiality
Confidentiality refers to only authorized parties or systems having the ability to access protected data sharing. A good example of methods used to ensure confidentiality is an account number or routing number when banking online. Data encryption is a common method of ensuring confidentiality. Confidentiality is achieved through encryption algorithm key policy attribute based encryption. as it is encryption technique user can do encryption with a public key generated runtime.

Availability
Availability refers to the property of a system being accessible and usable upon demand by an authorized entity. When user is encrypting file then after encryption it gets stored on cloud means available to anyone who want it.

7. Conclusion

In key policy Attribute Based Encryption, data integrity is maintained through Secure Hash Algorithm. It creates hash key for every chunk which is unique. Proposed algorithm gives confidentiality to work as it is a public key encryption algorithm of cryptography. It is also asymmetric i.e. key set is used for data encryption and decryption. Data is stored on cloud means it is available to everyone through Cloud space. Key generated for encryption and decryption is runtime. In this work analysis is there. Encryption and decryption time required by Identity Based Encryption and key Policy Attribute Based Encryption is calculated for comparison purpose. Both algorithms user can apply on different file type such as pdf, ppt, doc, mp3, jpg and file size. Key policy attribute based encryption reduces encryption time by 35% and decryption time by 65%.

References

[1] http://www.nist.gov/itl/cloud/ , date: 2 Auguet2015.
[2] Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H.Deng, "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", *IEEE Transaction on distributed and parallel systems*, 2013.
[3] F. Guo, Y. Mu, and Z. Chen, "Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key," *Proc. Pairing-Based Cryptography Conf. (Pairing '07)*, vol. 4575, Springer 2007, pp. 392-406.
[4] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," *Proc. Information Security and Cryptology (Inscrypt '07*), ser. LNCS, vol. 4990, Springer, 2007, pp. 384-398.
[5] Abdulaziz Aljabre, "Cloud Computing for Increased Business Value", *International Journal of Business and Social Science*, Vol. 3 No. 1; January 2012.
[6] Mikhail J. Atallah, Keith B. Frikken, and Marina Blanton, "Dynamic and Efficient Key Management for Access Hierarchies*", CCS* '05, November 2005, pp. 7-11.
[7] Matthew Green, Giuseppe Ateniese, "Identity-Based Proxy Re-Encryption", *in the 12th Annual Network and Distributed System Security Symposium*, Springer 2005,pp. 29–43,
[8] D Boneh, M Franklin , "Identity-based encryption from the Weil pairing", in *SIAM Journal on Computing*, Springer 2003, pp. 11-17 .
[9] F. Guo, Y. Mu, and Z. Chen, "Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key," *Proc. Pairing-Based Cryptography Conf. (Pairing '07)*, vol. 4575, Springer 2007, pp. 392-406.
[10] Guojun Wang, Qin Liu, Jie Wu, "Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services*",* ACM 978-1-4503-0244-9/10/10, 2010.
[11] Peter Mell, Tim Grance, "The NIST Definition of Cloud Computing", Version 15, 10-7-09.
[12] .Ashutosh Kumar, Shreyas Petkar, Prasad Jat, "Securing Military Data Using CP-ABE and Triple-DES with Multi-Authority System*" International Journal of Advanced Research in Science*, Engineering and Technology Issue 11. Vol. 2, November 2015
[13] .Minal Moharir1 and Dr A V Suresh "A Novel Approach using Advanced Encryption Standard to implement Hard Disk Security" in *IJNSA*, Vol 4 No.1, January 2012.
[14] http://searchsecurity.techtarget.com/definition/RSA , date 6 July 2015.
[15] B.Persis Urbana Ivy, Purshotam Mandiwa. Mukesh Kumar **"**A modified RSA cryptosystem based on 'n' prime numbers" *International Journal of Engineering and Computer Science*, ISSN: 2319-7242Volume1 Issue 2 Nov 2012, pp. 63-66.