*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661,p-ISSN: 2278-8727,*
*PP 25-29*
*www.iosrjournals.org*

# Extracting Key for Secuire Data Sharing In Cloud Storage

A.B.Bavane[1], Parakh Rakhi Santosh[2], Jindam Gayatri Ramesh[3], Pawar Shubham Anil[4] Dixit Shubham Balkrishna[5].
[12345](*P.D.V.VP.C.O.E,Ahmednagar,M.S.,India.*)

***Abstract**-Data sharing in a cloud storage is an important functionality. So in cloud storage how securely, efficiently, and flexibly we share our data with others is becomes challenging now a days. In this paper we proposed a new cryptosystem for efficient delegation of decryption rights in any set of cipher text possible which produces constant size cipher text .the system produces one such a key which is a aggregate key of all set of secret keys and encompassing the power of all this keys being aggregate. The secrete key holder release that one aggregated key for some choices of encrypted files which he only want to share with others and remaining files are hidden from all and remains confidential throughout the sharing of all set of files among cloud storage. The release of this aggregate key for that one task make available to others by sending conventionally or stored smart card*

*As key holder gives permission for his some set of files and make available for sharing the one who wants that file can use this with the help of available aggregate key. The formal security analysis is also given in this paper to introduce a standard model. In short our system gives the first public-key-patient controlled encryption for hierarchy which is flexible throughout the sharing of data*

***Keywords:** Data Sharing, Cryptosystem, aggregate key, encrypted files etc.*

## I. Introduction

Cloud storage getting popularity recently.it is saving of digital data in physical storage on multiple server. This server managed by third party. This third party responsible for availability, protected, accessible. This data save to remote storage. It is accessible from anywhere, anytime and anyone through internet. For the data privacy we cannot use traditional technics of security system because unexpected privilege escalation and expose all data. For that the solution is to encrypt the data with user's key before uploading it. The use of public-key encryption gives more flexibility for our applications. For example, in enterprise settings, every employee can upload encrypted data on the cloud storage server without the knowledge of the company's master-secret key

In enterprise environment we see the rise in demand for data outsourcing in which that it help to manage the corporate related data in strategic manner it is used as a core technology behind many online services for personal applications and services. To apply for free accounts for emails, photo albums, file sharing and remote access, with storage size more than 25GB (or a few dollars for more than 1TB) become easy in now a days.

Now a days users can access almost all of their files, emails or any other data by mobile phones in any corner of the world by using together with the current wireless technology. In the traditional way for data privacy there is to ensure is it is to depend on the server. Server provide the access control to user after verifying its authentication. But there is chance for any unexpected privilege escalations will get access and open or expose all data. For the data sharing in cloud computing environment data form various clients machine are stored in single physical machine. There are several VM (virtual machine) reside on single machine. Data in one VM may be stolen by another VM with respect to one target at a time. There are various third party auditors are provided in a list of cryptographic schemes according to the availability of files and data. these third party auditors are used to checks avability of files behalf of the owner of that data without leakageing its confidential information during sharing likewise cloud user does not believe in the cloud storage for the sharing because it assumes that cloud storage is not giving its best in terms of confidentiality. As the people are not happy with the security of the VM or the honesty of technical staff, a cryptographic solution with security re-lied on number of theoretical assumptions is more preferable. This type of users encrypt their data before uploading to the related server.as data sharing is very important now a days in a cloud storage for example, the friends or bloggers can view the some private picture of an enterprise who may grant her employees for access a part of sensitive data

## II. Related Work

### 2.1 EASiER-Encryption-Based Access Control InSocial Networks With Efficient Revocation

The trusted approach for minimising the risk in internet social network we have to shift focus from OSN provider to user which uses the encryption. Due to this the challenge of key management is involved in dynamic groups.to overcome the risks involved in OSN the given paper proposed with easier architecture which support good access control policies and dynamic group membership with the help of attribute-based encryption. Creation of a proxy can be takes place that participates in the decryption process. The proxy is trusted cipher text and cannot be decrypted.

Various mathematical assumptions, and some information of CP-ABE be implemented in this paper. EASiER that supports efficient revocation in ABE and access control architecture for internet social network is introduced

### 2.2 Multi-Authority Attribute Based Encryption-

In an identity based encryption, user is identified by using a unique identity word. Where as in attribute based encryption (ABE), each user is identified by a set of attributes, and some function which helps us to determine decryption ability for each cipher text. Giving challenge to a sahai and waters this paper allows any polynomial number of independent authorities for monitoring attributes and distributing secret keys.

In encryption process for each authority, a number dk and a set of attributes can be chosen; If he has at least dk of the given attributes from each authority k then an then he can encrypt a message such that a user can only decrypt it.This paper helps to tolerate an arbitrary number of corrupt authorities. A multi-authority version of the large universe fine grained access control ABE is also included in this scheme. In this scheme, each user must go to the trusted server and he have to prove that he has a certain set of attributes according to this he receive secret keys according to each of those attributes.

## III. Architecture

If data owner puts his private data on drop box and he does not want to share his data with anyone. But due to various data leakage data owner do not relying security provided by drop box data owner will encrypt all his data before uploading. If another user want that data then data owner can share his data. But problem is how to delegate decryption rights. There are two way for data owner to share data with others

1) Data owner encrypt all his data with single key and gives to requested user corresponding key.
2) Data owner encrypt data with distinct keys and send requested user corresponding keys.

Obviously, in first method unchosen data may be leaked to user. For second method there are practical concern on efficiency. The number of such keys as much as number of share data. For transferring these keys required secured channel. Also for storing these keys required expensive secured storage. If number of decryption keys to be share then definitely cost and complexity increases.

The solution for above problem is that data owner encrypt data with distinct public key. But only sends user single decryption key. This decryption key should be send through secure channel. In this project we study how to make decryption key more powerful. We solve this problem by introducing special public key which we call key aggregate cryptosystem.

In KAC, data owner encrypt message with public key and under identification of cipher text called class. Cipher text are categorised into different classes. Key owner holds master-secret called master-secret key. These key used for extract secret key for different classes. These extracted key called aggregated key. Which compact as single class. With these solution data owner can send to requested user aggregated key through secure channel. Requested user will download data from data owners Drop box then decrypt all downloaded data.

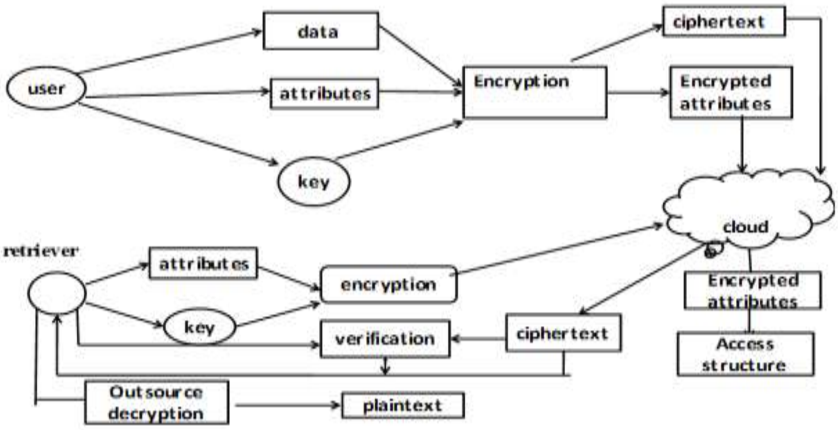

Fig 1.1:Data Sharing And Storage In Cloud Computing

## IV. Framework

The data owner generate the public system parameter through Setup and generates a public/master-secret key pair through KeyGeneration. Data can be encrypted via Encrypt by anyone who also decides what cipher text class is associated with the plaintext message to be encrypted. The data owner can use the master-secret key pair to generate an aggregate decryption key. The generated keys can be passed to delegates securely through secure channel or secure devices. Any user with an aggregate key can decrypt any cipher text provided that the cipher text’s class is contained in the aggregate key via Decrypt. Key aggregate encryption schemes consist of five polynomial time algorithms as follows:

**1. Setup ():**Public system parameter can be initialised by data owner via setup. We gives input as security parameter f, number of cipher text classes as n and it outputs the public system as para.

**2. Encrypt (P, J, M):**The plain text can be converted into the cipher text. The cipher text computed for message m and index as j. It is executed by data owner and for message m and index i, Itcomputes the cipher text as c. It is also executed by data owner

**3. Keygeneration()***:* It can randomly generate a public/ master-secret key pair and executed by the data owner. The aggregate key for a set of classes can be generated which encompasses the power of all secrete keys.

**4. Decrypt ():**It is executed by a key requested user.In this process the given selected cipher text can be converted into plain text.

## V. Algorithm

AES is a block cipher.It allows for three key lengths: 128, 192, or 256 bits, in which 128 bits key mostly used in assumptions. The encryption of 128 bit keys there are 10 rounds,192 bits 12 rounds whereas for 256 bit keys 14 rounds are needed. The 128-bit block are arranged in 4×4 matrix of bytes.Therefore in the 128bit input block first bytes occupy first column in 4×4 matrix of bytes, next occupy the second column and so on.

AES having concept of word. A word consists of four bytes that is nothing but 32 bits. So the four column words of the key matrix are expanded into aschedule of 44 words. For encryption before any round-based processing can begin, input array is XOR with the first four wordsduring decryption the same thing happens — except that now we XOR the cipher text array with the last four words of the key schedule. For encryption, each round consists of the following four steps:

**5.1 substitute bytes**: Substitution take placed byte-by-byte. Each input byte of the substitution is found by lookup table having size of 16 × 16.To find the substitute byte for a given input byte, we divide the input byte into two 4-bit patterns, each holding an integer value between 0 and 15 represented in hex (0 to f). Hex values is used asone for a row indexand other for column index to reach in the 16×16 lookup table.

**5.2 shift rows**: This is where the matrix representation of the state array be-comes important. The shift of rows transformation consists of :

- not shifting the first row of the state array at all;

- circularly shifting the second row by one byte to the left;
- circularly shifting the third row by two bytes to the left; and
- circularly shifting the last row by three bytes to the left.

The first four bytes of the block fill the first column of the state array, the next four bytes by the second column and so on and goes shifting the rows in the manner as an output for decryption, The corresponding step shifts the rows in exactly the opposite fashion. The first row is left unchanged the second row is shifted to the right by one byte. The third row to the right by two bytesand the last row to the right by three bytes, all shifts being circular.

**5.3 mix columns:** This step replaces each byte of a column by a function of all the bytes in the same column. The byte in a column is replaced by two times. That byte plus three times the next byte, plus the byte that comes next means multiplication in gf (28) by the bit pattern 00000010 and 00000011.

Operation for the bytes in the first row of the array is given as:

$0,J = (0x02 \times S0,J) \otimes (0x03 \times S1,J) \otimes S2,J \otimes S3,J$

Operation For The Bytes In The Second Row Of The Array Is Given As

$1,J = S0,J \otimes (0x02 \times S1,J) \otimes (0x03 \times S2,J) \otimes S3,J$

Operation For The Bytes In The Third Row Of The Array Is Given As

$2,J = S0,J \otimes S1,J \otimes (0x02 \times S2,J) \otimes (0x03 \times S3,J)$

Operation For The Bytes In The Fourth Row Of The Array, Is Given As

$3,J = (0x03 \times S0,J) \otimes S1,J \otimes S2,J \otimes (0x02 \times S3,J)$

**5.4 Add Round Key**: For The ith Round:Wi Wi+1 Wi+2 Wi+3
i Must Be A Multiple Of 4. These Will Gives Round Key For The (i/4)Th Round.
To Determine the Words: Wi+4 Wi+5 Wi+6 Wi+7
From The Words Wi Wi+1 Wi+2 Wi+3.
Then We Have,

$Wi+5 = Wi+4 \otimes Wi+1$ (1)

$Wi+6 = Wi+5 \otimes Wi+2$ (2)

$Wi+7 = Wi+6 \otimes Wi+3$ (3)

We Have Need Only Wi+4. The Beginning Word Of Each Round Key Is Obtained By:

$Wi+4 = Wi \otimes G(Wi+3)$ (4)

That is, the first word of the new 4-word grouping is to be ob-trained by XOR the first word of the last grouping with what is returned by applying a function g() to the last word of the previous 4-word grouping. The function gf() consists of the following three steps:
Perform a one-byte left circular rotation on the argument 4-byte word.
Perform a byte substitution for each byte of the word returned by the previous step by using the same 16 × 16 lookup table as used in thesub bytes step of the encryption rounds. Is known as a round constant. Therefore, xor with the round constant amounts to xor withjust its leftmost byte. The round constant for the ith round is denoted rcon [i].

rcon[i] = (rc[i], 0x00, 0x00, 0x00)

The only non-zero byte in the round constants, rc[i],obeys the following recursion:

$Rc[1] = 0x01$

$Rc[j] = 0x02 \times rc[j - 1]$

The last step consists of xoring the output of theprevious three steps with four words from the key schedule
.

## VI. Conclusion And Future Work

User data protection, security, privacy is becoming main issue of cloud storage. Using many mathematical models and cryptographic technique multiple keys for single application can be generated. In our system we have used secret keys in public key cryptographic systems to make compact difference of secret key for various cipher text classes in cloud storage management. Without reference of power set of classes the delegate created will always get aggregate key of same size. Aggregate key size can also be constant. Our works more concentrated on hierarchical key assignment which is flexible and which can only save storage space if all the key holder share a same set of classes.

Our future work is to predefine bound of number of many ciphertext classes in cloud storage amount of cipher text gradually increases .so we have to save enough of cipher text privileges for future. We can also expand public key although the parameter can be downloaded with cipher text classes. It will be better if its size is independent

## References

[1] Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng, Senior Member, IEEE

[2] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, Privacy-Preserving Public Auditing for Secure Cloud Storage, IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.

[3] B. Wang, S.S.M. Chow, M. Li, and H. Li, Storing Shared Data on The Cloud via Security- Mediator, Proc. IEEE 33rd Intl Conf.Distributed Computing Systems (ICDCS), 2013.

[4] S.S.M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R.H. Deng, Dynamic Secure Cloud Storage with Provenance, Cryptography and Security, pp. 442-464, Springer, 2012.

[5] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, Proc. 22nd Intl Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT 03), pp. 416-432, 2003.

[6] M.J. Atallah, M. Blanton, N. Fazio, and K.B. Frikken, Dynamic And Efficient Key Management for Access Hierarchies, ACM Trans. Information and System Security, vol. 12, no. 3, pp. 18:1-18:43, 2009.

[7] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records, Proc. ACM Workshop Cloud Computing Security (CCSW 09), pp. 103-114, 2009.

[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters, Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data, Proc. 13th ACM Conf. Computer and Comm. Security (CCS 06), pp. 89-98, 2006.