# Development of Enhanced Key-Aggregate Cryptosystem for Secure Cloud Storage

Goldie Lee Joe
Mtech, Cyber Security,Dept Of IT
Student Member, IEEE
Nehru College of Engineering and Research Centre
Pampady,Thrissur,Kerala,India

Dr. N. K. Sakthivel
Vice Principal
Nehru College of Engineering and Research Centre
Pampady, Thrissur, Keral,India.

***Abstract*— Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users' physical possession of their outsourced data, which inevitably poses new security risks towards the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, key management and key sharing plays the main role in the data sharing concept of cloud computing. Aggregate key cryptosystem is a concept in which key is generated by means of various attributes of data in various cipher text classes and its associated keys. Aggregate key consist of various derivations of identity and attribute based classes of respective data owner of cloud. Irrespective of traditional cryptographic key generation and derivations, this technique possesses complex, unique and robust cryptographic key aggregate cryptosystem which is optimal for secure cloud and privacy preserving key generation process. We define access level policy in IAAS structure such as Public and Private Access level to enhance the data access mechanism in the data sharing cloud mechanism process. We also enhance the data security algorithm such as Blowfish to AES as Blowfish algorithm results in higher security and faster execution when compared to AES (Advanced Encryption standard) and DES (Data Encryption Standard). Besides the blowfish algorithm is unpatented and no license is required. Even though the algorithm has been subjected to a number of cryptanalysis, the full blowfish algorithm has never been broken. Proposed approach proves more secure and efficient cryptographic scheme in which we have an effective derivation of secret key generation and key management for the outsourced Cloud data.***

***Keywords—Cloudcomputing,IAAS cloud,Access level mechanisms.***

## I. INTRODUCTION

Cloud has become the best storage mechanism for all the users who access online resources and is gaining high popularity recently. Cloud storage plays a vital role in many personal applications as it the core technology for its existence. Many users are accessing the cloud space since Google Drive, One cloud etc are providing access to the common user to make them aware about the convenience of the cloud storage and its access. When the wireless technology joined the hands with cloud it has turned to a miracle fulfilling every needs of the user from any corner of the world.

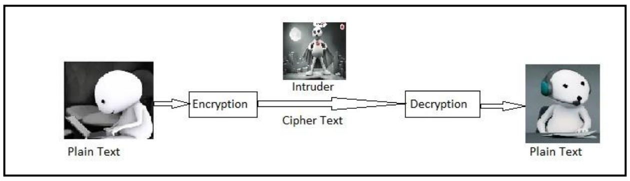

Fig 1.1 Encryption and Decryption

The safekeeping of the data which is being amassed onto the cloud has become one of the key concerns. The trust of a cloud user cannot be relied blindly on a cloud provider completely. The confidentiality and integrity of the data cannot be assured if it is uploaded as such to the cloud. We depend on many cryptographic schemes to overcome this issue. Cryptographic schemes don't assure complete security but prevent the absolute revealing of the secret data. The major limitation arrives when the user needs to share the access to other on fine-grained level. One method is that the user has to provide the permission to access the complete data since the selected data permission can't be granted. Another method is that separate encryption has to be done the selected data one-by-one separately and send the private keys to the one who request. This is practically impossible when we consider the time, cost, complexity etc.

Data can be so shared by encrypting all the selected data with its attributes and secret key converting it to a single aggregate key(private key) and this key can be sent over any communication channel like email, message etc. This mechanism not only saves the space, but also the execution time, cost, complexity etc.

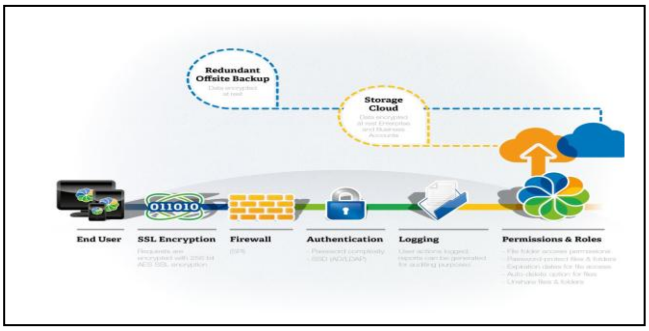
Redundant Offsite Backup
Data encrypted at rest
Storage Cloud
Data encrypted at rest Enterprise and Business Accounts
011010
End User
SSL Encryption
Requests are encrypted with 256 bit AES SSL encryption
Firewall
(IDS)
Authentication
- Password complexity
- SSO (AD/LDAP)
Logging
User actions logged, reports can be generated for auditing purposes
Permissions & Roles
- File folder access permissions
- Password-protect files & folders
- Expiration dates for file access
- Auto-delete option for files
- Unshare files & folders

Fig1.2 Security Architecture of Cloud

The aggregate key can be used only to decrypt the data with which it was encrypted which means all the other data outside this set remains safe and hidden to the one to whom the aggregate key is being sent.

**Our contribution:** Cryptography is an amazing technique with which veils the veracity of the message from superfluous users. The Key-Aggregate Cryptosystem (KAC) [1] provides an outstanding performance reducing the computational complexity of the overall algorithm. The KAC aggregates various cipher texts into cipher text classes and every class holds a secret key from which the aggregate key will be generated. This generated aggregate key holds the decryption power of any subset of cipher classes.

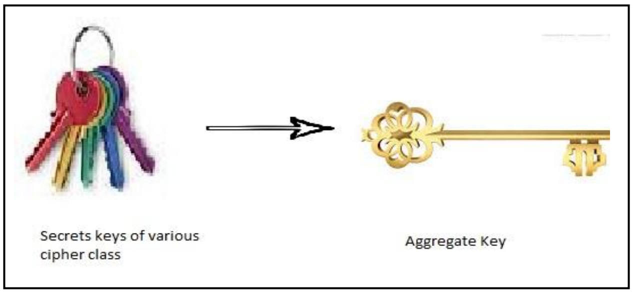

Fig 1.3 Multiple secret keys to single powerful Aggregate Key

Alice can send to bob the aggregate key as an email so that bob can decrypt the set of data which is being encrypted using the aggregate key and the set outside this encryption remain hidden to bob. Another advantage of this scheme is that the size of cipher text, aggregate key and the master secret key remains constant. KAC is a flexible work that the cipher text classes need not establish a relationship between each other[1].

We propose to perform the encryption and decryption process using the blowfish algorithm since Blowfish has a 64-bit block size and a variable key length from 32 bits up to 448 bits and making it ideal for securing data. It is a variable-length key block cipher. It is suitable for applications where the key does not change often, like a communications link or an automatic file encryption. Blowfish is a symmetric block cipher that can be used as a drop-in replacement for DES or IDEA. It has been analyzed considerably, and it is slowly gaining acceptance as a strong encryption algorithm it is much faster when compared to other symmetric algorithms

## II. KEY-AGGREGATE ENCRYPTION

The key-aggregate encryption process comprises of five polynomial-time algorithms as follows.[1]
The data owner generates the public system parameter with the Setup algorithm and engenders a public/master-secret3 key pair through the KeyGen. Encryption of the messages to be stored on to the cloud can be done with the Encrypt algorithm. The master-secret key thus generated can be used to form the aggregate key in the Extract process. The generated aggregate key can be sent to delegatee securely as an email or through portable devices. Finally, any client with an aggregate key can decrypt the data associated with this key received though the process called Decrypt.

1. Setup: This is a randomized algorithm that takes no input other than the implicit security parameter.
2. KeyGen: randomly generate a public/master-secret key pair (pk,msk).
3. Encrypt (pk,i,m): performed by anyone who is the owner of the data. Encrypts the data m using the public key and the index i of the cipher class and outputs C.
4. Extract (msk,S): this process results an aggregate key when we input the set of indices of the cipher class along with the master secret key.
5. Decrypt : decrypt is the process done by the one who receives the aggregate key obtaining the message m iff i ε S.

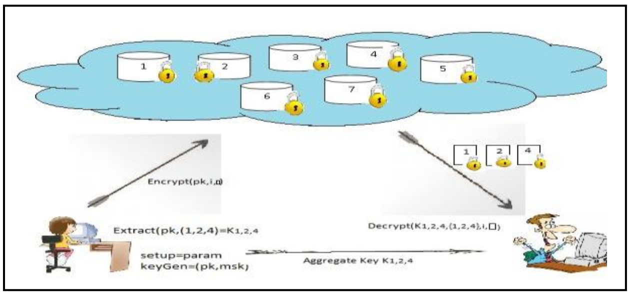

Fig 2.1 Data sharing in clouds

The above figure shows how the data is being shared in the cloud. Suppose a user A wants to share the data 1,3,5 to another user B, then the user A generates an aggregate key using the attributes of 1,3,5 and sends it to the user B. The user B thus decrypts the required data by performing the setup to generate the param, then the keys are generated followed by the encryption process. The extract process generates the aggregate key with which the user B decrypts the data.

In the traditional methods the key assignment will be providing separate keys for every data to be decrypted. This increases the key generation process as well as consumes much larger space.

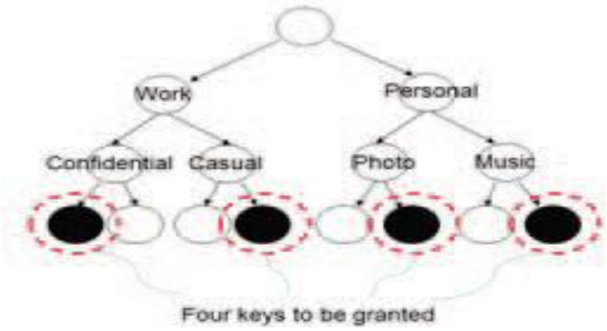

Fig 2.2 Key assignment for traditional cryptographic scheme

Here in the above figure four separate keys are to be granted for the availability of these files. The KAC uses only half the no.of keys than in the traditional cryptographic schemes. For example, consider the figure

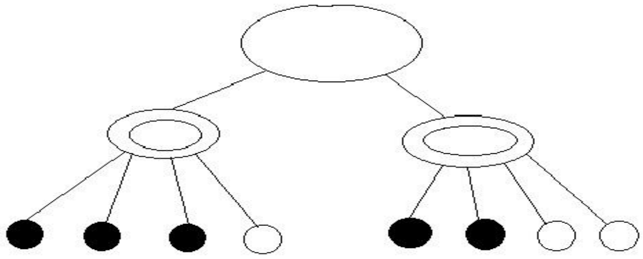

Fig 2.3 Key assignment in KAC

Only two different keys has to be provided for the access of five data. Hence hierarchical level of classification of data has a more advantageous level than the data arrangement in class level.

## III. Identified Problems

The symmetric algorithm used is having many advantages even though it come with certain problems like power consumption, high execution time, cost complexity and very low throughput. The encryption and the decryption process is consuming a large amount of time and hence increases its execution time along with the power consumption. Since the execution time is higher the cost complexity is also higher. Even though the algorithm proposes a compressed data storage mechanism, the cipher text classes occupy a higher space in the cloud network. To over these issues to a limit we come forward with the proposed system.

## IV.PROPOSED SYSTEM

The proposed system is the Blowfish algorithm which was designed in 1993 by a great scientist Bruce Scheier as a swift, substitute to accessible encryption algorithms like AES, 3DES and DES etc.. Blowfish algorithm is a symmetric block encryption scheme intended in contemplation with,

- **Fast:** Data encryption takes place at a rate of 26 clock cycles per byte on 32-bit microprocessor.
- **Compact:** 5K of memory is more and enough to execute efficiently.
- **Simple:** It makes use of XOR, addition, lookup table with 32-bit operands.
- **Secure:** The key length is variable, it can be in the range of 32~448 bits: default 128 bits key length.
- It is apposite for applications where the key does not alter often, like communication link or an automatic file encryptor.
- It does not have a patent and also royalty-free.

**Description of Algorithm:**

Blowfish algorithm is a symmetric block cipher algorithm which encrypts block data of 64-bits at a time. It will pursue the feistel system and this algorithm is mainly divided into two parts.

1. Key-expansion
2. Data Encryption

1.Key Expansion: the key expansion process converts a key of 448 bits into numerous subkey arrays making it to a size of 4168 bytes. Blowfish makes use of a large number of subkeys. These keys will be generated earlier to any data encryption or decryption.

The p-array consists of 18, 32-bit subkeys:
P1,P2,………….,P18
Four 32-bit S-Boxes consists of 256 entries each:
S1,0, S1,1,………..S1,255
S2,0, S2,1,………..S2,255
S3,0, S3,1,………..S3,255
S4,0, S4,1,..............S4,255

2.Data Encryption: Data encryption is having a function to iterate the function 16 times of network. Each separate round consists of a key-dependent transformation and a key and data-dependent changeover. All operations performed are XORs and the additions on the 32-bit words. The only supplementary operations to the above functions are four indexed array data lookup tables for each round.

```
Divide x into two 32-bit halves: xL, xR
    For i = 1to 16:
        xL = XL XOR Pi
            xR = F(XL) XOR xR
        Swap XL and xR
        Swap XL and xR (Undo the last swap.)
        xR = xR XOR P17
        xL = xL XOR P18
        Recombine xL and xR
```

## IV. Performance Analysis

Compared to all other algorithms the blowfish algorithm has made its mark in the cryptographic field. The unbeatable strength of the encryption algorithm is mainly depended upon the key length. Bruce Schneier, originator of the Blowfish encryption algorithm, has calculated that according to what we know of quantum mechanics today, that the entire energy output of the sun is insufficient to break a 197-bit key.

**Speed Comparisons of Block Ciphers**

| Algorithm | Clock cycles per round | # of rounds | # of clock cycles per byte encrypted | Notes |
|---|---|---|---|---|
| Blowfish | 9 | 19 | 18 | Free, Not patented |
| Khufu/Khafre | 5 | 32 | 20 | Patented by Xerox |
| RC5 | 12 | 16 | 23 | Patented by RSA Data Security |
| DES | 18 | 16 | 45 | 56-bit key |
| IDEA | 50 | 8 | 50 | Patented by Ascom-Systec |
| Triple-DES | 18 | 48 | | |

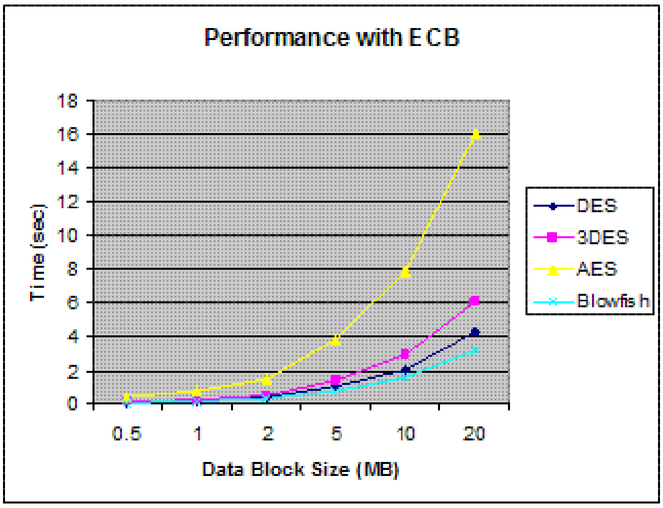

Fig 5.1. Encryption performance comparison with ECB

The performance speed of the algorithm is also exciting. Chances are high to think that a 448 bit key length is too much. However, when the scrutinizing of the algorithm is done, the effectual throughput of the Blowfish algorithm, we see that even large key lengths result in much faster performance than other encryption algorithms.

Blowfish makes use of a memory size of just over 4 kilobytes of RAM foe its execution. This constriction is not a crisis even for the very old type of desktops and laptops, though it does avert use in the tiny embedded systems such as untimely smartcards.

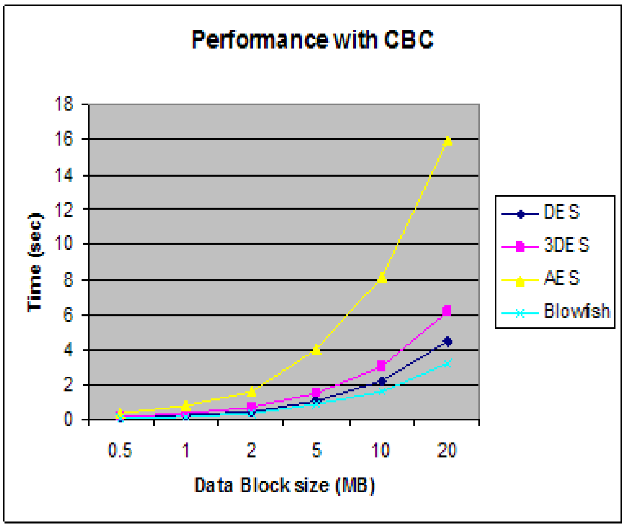

Fig 5.2 Encryption performance comparison with CBC

## V. Conclusion

The above stated results declare that Blowfish has much more advantages when compared to the performance of many other algorithms. Blowfish can be marked as an excellent encryption algorithm since no weak points about the algorithm hasn't been revealed so far. The performance of the AES algorithm is much poor when compared to other algorithms sic it is demanding much processing time and power. Use of the CBC mode has added further processing time, but taken as a whole it was relatively trifling particularly for convinced application that entails more sheltered encryption to a moderately huge data blocks.

As encryption algorithms progress to convene the ever-increasing pace of systems intended to "fissure" them, we will struggle to slot in these improved algorithms in all of our products. But awaiting the next cohort of encryption is developed, rest guaranteed that Blowfish will put forward considerable protection for many years to arrive and will persist to be the preferred encryption algorithm used by many corporations and banking institutions worldwide.

## VI. Refernces

[1] 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 Transactions on Parallel and Distributed Systems, vol. 25, no. 2, February 2014.

[2] Milind Mathur, Ayush Kesarwani, "comparison between DES , 3DES , RC2 , RC6 , Blowfish and AES," Proceedings of National Conference on New Horizons in IT - NCNHIT 2013.

[3] Vipul Goyal, Omkant Pandeyy, Amit Sahaiz, Brent Waters," Attribute-Based Encryption for Fine- grained Access Control of Encrypted Data.

[4] MD Asif Mushtaque , Harsh Dhiman , Shahnawaz Hussain , Shivangi Maheshwari "Evaluation of DES, TDES, AES, Blowfish and Two fish Encryption Algorithm: Based on Space Complexity," International Journal of Engineering Research & Technology (IJERT),ISSN: 2278-0181, Vol. 3 Issue 4, April – 2014