ISSN:0975-9646

Development of improved Aggregated Key Cryptosystem for scalable data sharing

Rashmi Khawale, Roshani Ade

D.Y. Patil School of Engineering and Technology, Lohgaon, SPPU, Pune, India

***Abstract* – Cloud plays the vital role in internet world. Cloud provides storages, platforms which improves the functionality. Cloud storage shows how securely and flexibly we can store and share our data. With the help of keys user can easily and securely share their data over cloud. This introduces Key Aggregate Cryptosystem in which an aggregate key is created using which user can share their data partially over cloud and it provides a constant size ciphertext. In spite of traditional cryptographic key generation techniques, this technique possesses unique cryptographic key aggregate cryptosystem which is helpful for secure cloud and privacy preserving key generation process. We propose access level policy structure such as Public and Private Access level to improve the data access mechanism in the data sharing cloud mechanism process. We are using algorithm such as Blowfish algorithm which results in higher security and faster execution when compared to AES (Advanced Encryption standard) and DES (Data Encryption Standard). Also the blowfish algorithm is unpatented and no license is required**

***Keywords*— Virtual machine, Key aggregate encryption, ciphertext, Attribute based Encryption, Aggregate keys, Extraction, Blowfish Algorithm.**

I. INTRODUCTION

Cloud storage is the most popular functionality recently. Cloud-based services include Software-as-a-Service (SaaS) and Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Cloud computing gives us various facilities for data storage and data sharing. User can easily transfer his data using cloud storage in the GB or TB units. Thus cloud storage is advantageous in terms of low cost and high availability of data. So the security of cloud storage is a major concern in the cloud computing environment, as user can store any type of information in the cloud storage.

In cloud computing environment when we share data across cloud , data from different users can be stored on separate virtual machines (VMs) but may reside on a single physical machine. But data in a target VM could be stolen by starting another VM on same physical machine. When we consider traditional ways of data privacy, some depends on the server to enforce the access control after authentication [3] or some allows a third-party auditor to check the availability of files on behalf of the data owner without leaking the data [2]. But cloud user can not fully depend on cloud server for their data security, privacy and confidentiality purpose. Thus users are motivated to encrypt their data with own keys .

Let us consider a condition, user A uploads a set of photos over cloud. But he does not want to share all these photos with everyone. So he need to put some security constraints. With the available cloud security services user A is not satisfied. So he encrypts his photos using his own keys before uploading. Now when user B asks user A to share his photos, user A will send him a single constant size decryption key via secure channel. With this decryption key, user B is allowed to decrypt only those photos which are permitted by user A.

This paper provides the technique using which partial data sharing is possible that is using Key Aggregate Cryptosystem (KAC) [1]. With this solution, user A can simply send user B a single aggregate key via a secure e-mail. Then user B can download the encrypted photos from A’s cloud storage space and then use this aggregate key to decrypt these encrypted photos. The sizes of ciphertext, public-key, master-secret key, and aggregate key in this KAC schemes are all of constant size.

Comparing our basic KAC technique with different techniques of sharing data in cloud storage like Cryptographic Keys for a Predefined Hierarchy, Compact Key in Symmetric-Key Encryption, Compact Key in Identity-Based Encryption (IBE) ,Attribute-based encryption (ABE) , the results are listed in following table-

TABLE I
COMPARISON BETWEEN KAC AND OTHER SCHEMES

	Decryption Key size	Ciphertext size	Encryption Type
Key Assignment Schemes for predefined hierarchy	Non constant	constant	Symmetric key or Public key
Symmetric key Encryption with compact key	constant	constant	Symmetric key
IBE with compact key	constant	Non constant	Public key
Attribute based Encryption	Non constant	constant	Public key
KAC	constant	constant	Public key

II. OUR CONTRIBUTION

Cryptography is an amazing technique using which various user can access data from different users. The Key-Aggregate Cryptosystem (KAC) [1] provides great results 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.

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.

III. Existing System

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): 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.

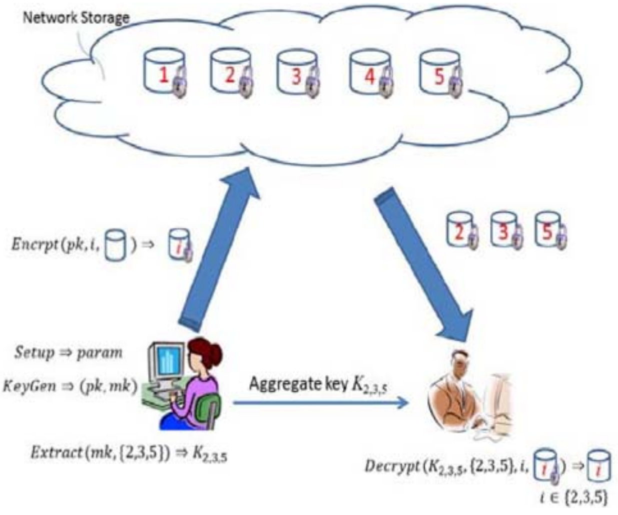
Network Storage
1
2
3
4
5
Encrpt(pk, i,) ⇒ i
2
3
5
Setup ⇒ param
KeyGen ⇒ (pk, mk)
Aggregate key $K_{2,3,5}$
Extract(mk, {2,3,5}) ⇒ $K_{2,3,5}$
Decrypt($K_{2,3,5}$, {2,3,5}, i, i) ⇒ i
i ∈ {2,3,5}

Fig. 1 Data sharing using KAC

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 which provide,

- **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 appropriate 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. 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
```

Decryption is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order.

Implementations of Blowfish require the fastest speed should unroll the loop and ensure that all subkeys are stored in cache

V. Result 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. The results show the impact of changing data load on each algorithm and the impact of Cipher Mode used.

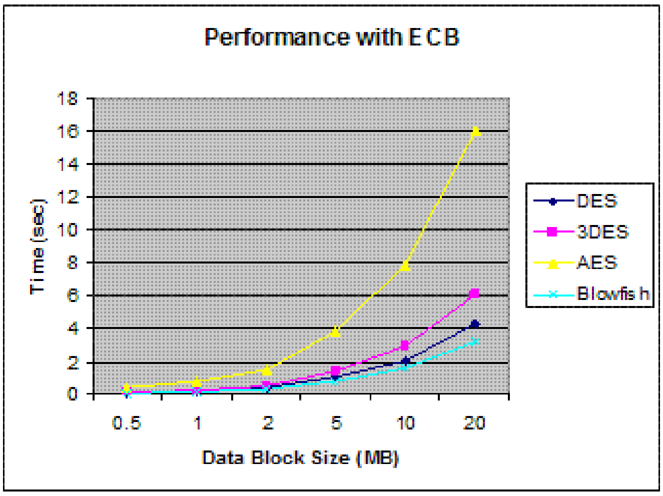

Fig 2 Encryption performance comparison with ECB

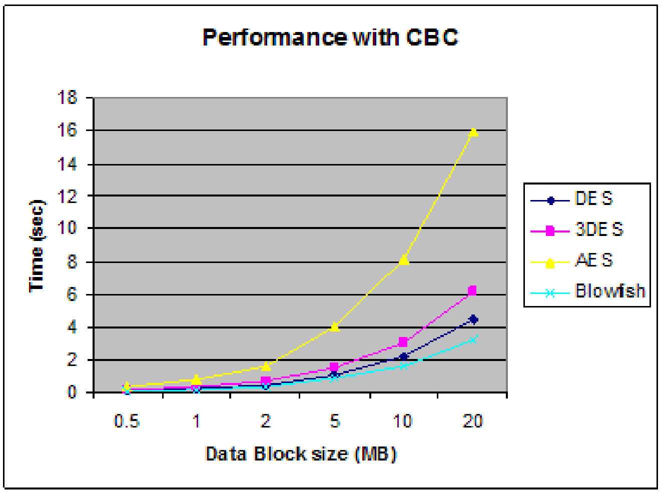

Fig 3 Encryption performance comparison with CBC

VI. Conclusion

Thus the Blowfish algorithm is implemented in given system. The results shows that Blowfish is much more advantages when compared to the performance of many other algorithms. Since Blowfish has not any known security weak points so far, this makes it an excellent candidate to be considered as a standard encryption algorithm. Using this technique we have designed Patient Controlled Encryption which helps user to store their medical records over cloud and partially share their data with desired user.

Acknowledgment

I take this opportunity to thank all in individuals for their guidance, help and timely support .It gives me great pleasure and immense satisfaction to present this paper which result of unwavering, support, expert guidance and focused direction of my guide Prof.Roshani Ade to whom I express my deep sense of gratitude and humble thanks, for valuable guidance throughout the work.

References

[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 Transaction on Parellel and Distributed System, vol. 25, no. 2, February 2014.

[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] 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.

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

[5] 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