Patient Controlled Encryption using Key Aggregation with Blowfish Algorithm

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

RoshaniRaut(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

1. 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 Aggregation (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.

The rest of the article is organized as follows: section II provides the related work of the paper. Section III provides steps of the KAC technique and system architecture. Section IV provides the result analysis of basic KAC technique. Section V provides introduction of new patient controlled encryption system. Section VI concludes the article. Section VII gives the acknowledgment.

2. RELATED WORK
Introducing new Key Aggregation which allows user to partially share their data partially with constant size decryption key. They have compared this method with other methods and shown result. They also implemented this method using PCE system. A key-aggregate encryption system basically includes five algorithmic steps as follows-

The data owner establishes the public system parameter by using **Setup** and generates a public/master-secret key pair by using **KeyGen**. Messages can be encrypted using **Encrypt** by anyone who also decides what ciphertext class is associated with the plaintext message to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of ciphertext classes by **Extract**. The generated keys can be passed to Receivers securely via secure e-mails.

Finally, any user with an aggregate key can decrypt any ciphertext provided that the ciphertext's class is contained in the aggregate key via **Decrypt** [1].

Data security becomes important functionality while uploading data over cloud. Thus considering traditional ways of data privacy, some allows a third-party auditor to check the availability of files on behalf of the data owner without leaking the data [2] or some depends on the server to enforce the access control after authentication [3] .

Moving to electronic health records is important to the modernization of healthcare system. But computerized medical records are vulnerable to cyber attacks. Also patient may need to share their data partially with some users. Thus designing Patient Controlled Encryption (PCE) provides solution to secure and private storage of patients' medical records.In PCE, the health record is decomposed into a hierarchical tree structure based on the use of different ontologies, and patient is the one who generate and store secret keys. So whenever there is a need to access part of the record, a patient will release the secret key for the concerned part of the record. Thus any patient can either define his own hierarchy according to his need, or follow the set of categories suggested by the electronic medical record system, such as disease, x-rays, doctors, allergies, medications, and so on. When the patient wishes to give access rights to her doctor, he can choose any subset of these categories and provide a single key, from which keys for all these categories can be computed. Thus, this cryptosystem helps user to securely and partially share the data over cloud. [4].

We compare our basic KAC technique with existing solutions of sharing data in cloud storage-

This all comparison can be summarized in following table

Table 1: 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

3. SYSTEM ARCHITECTURE

A key-aggregate encryption system basically includes five algorithmic steps as follows-

- Setup(1^{λ} , n): Data owner executes Setup to create an account on an untrusted server. With input as security level parameter 1^{λ} and the number of ciphertext classes n , it outputs the public system parameter param.
- KeyGen: Data owner executes KeyGen to randomly generate a public/master-secret key pair (pk ,msk)
- Encrypt(pk, i, m): Anyone can execute this step who wants to encrypt data with input a public-key pk, an index i denoting the ciphertext class, and a message m, which outputs a ciphertext C.
- Extract(msk, S): Executed by the data owner to handover the decrypting power for a certain set of ciphertext classes to a Receiver . On input the master-secret key msk and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted by Ks.
- Decrypt(Ks, S, i, C): executed by a Receiver who received an aggregate key Ks generated by Extract.On input Ks, the set S, an index i denoting the ciphertext class the ciphertext C belongs to, and C, it outputs the decrypted result m if i ϵ S.

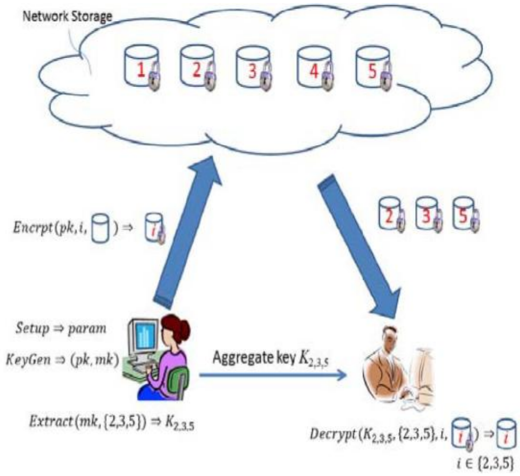

Fig. 1: Data sharing using KAC

As shown in figure first user divides his data into classes in his database. Then setup is created over cloud and keys are created for data. Using these keys classes of data are encrypted and stored over cloud. Now user extracts an aggregate key for a single class of data which he needs to share with another and only this aggregate key is received by customer. Now customer decrypt only allowed class of data with the aggregate key and data is received. Thus using KAC the purpose of partial data sharing over cloud is fulfilled.

4. PROPOSED WORK

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.

5. NEW PATIENT CONTRLLED ENCRYPTION (PCE)

Moving to electronic health records is important improvement of healthcare system which motivated the study of concept of patient controlled encryption. Thus this Key Aggregation is implemented in preserving patient's privacy in electronic health record systems. Thus patient can easily share their data over cloud and can also decide with which user he needs to share what data. In PCE, the health record is divided into a hierarchical representation based on use of differentontologies, and patients are the parties who generate and store secret keys. When there is a need for a healthcare system to access part of the record, a patient will release the secret key for the concerned part of the record. Any patient can draw his own hierarchy or follow the set suggested by electronic medical record system. When the patient wishes to give access rights to his doctor, he can choose any subset of these categories and issue a single key, from which keys for all these classes can be computed.

6. 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.

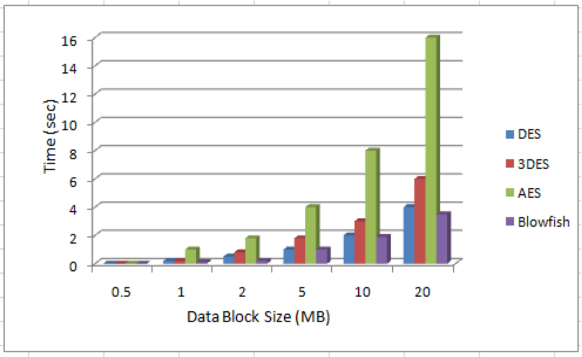

Fig 2 Encryption performance comparison with ECB

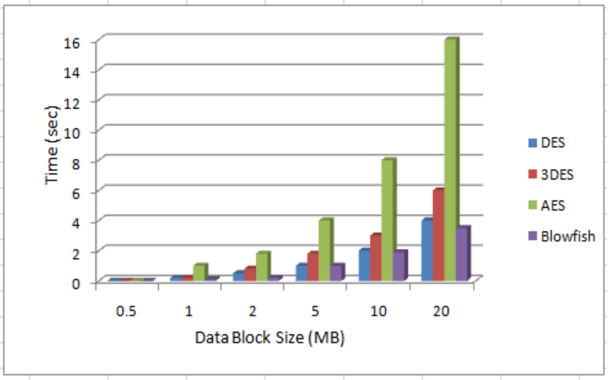

Fig 3 Encryption performance comparison with CBC

7. CONCLUSION

Cloud storage is gaining much popularity in these days. So data sharing and security becomes the crucial area to work. Our technique Key Aggregation helps user to share their data over cloud storage partially. 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. We also compared this technique with previous techniques and given result analysis.

8. 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.RoshaniRaut(Ade) to whom I express my deep sense of gratitude and humble thanks, for valuable guidance throughout the work.

9. REFERENCES

[1] Cheng Kang Chu, Sherman S.M. Chow, Wen-GueyTzeng, 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] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.

[4] 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.

[5] B. Wang, S.S.M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS), 2013.

[6] 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

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

[8] 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.

[9] S.G. Akl and P.D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Trans. Computer Systems,vol. 1, no. 3, pp. 239-248, 1983.

[10] J. Benaloh, "Key Compression and Its Application to Digital Fingerprinting," technical report, Microsoft Research, 2009.

[11] 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,pp. 392-406, 2007.

[12] 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.

[13] S.S.M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions," Proc. ACM Conf. Computer and Comm. Security, pp. 152-161, 2010.

[14] M. Chase and S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proc. ACM Conf. Computer and Comm. Security, pp. 121-130. 2009,

[15] T. Okamoto and K. Takashima, "Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption," Proc. 10th Int'l Conf. Cryptology and Network Security (CANS '11), pp. 138-159, 2011.

[16] R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 185-194, 2007

[17] C.-K. Chu, J. Weng, S.S.M. Chow, J. Zhou, and R.H. Deng, "Conditional Proxy Broadcast Re-Encryption," Proc. 14th Australasian Conf. Information Security and Privacy (ACISP '09), vol. 5594, pp. 327-342, 2009.

[18] S.S.M. Chow, J. Weng, Y. Yang, and R.H. Deng, "Efficient Unidirectional Proxy Re-Encryption," Proc. Progress in Cryptology (AFRICACRYPT '10), vol. 6055, pp. 316-332, 2010.

[19] RafailOstrovksy, Amit Sahai, and Brent Waters. Attribute Based Encryption with Non-Monotonic Access Structures. In CCS, 2007.

[20] D. Boneh, R. Canetti, S. Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption," SIAM J. Computing, vol. 36, no. 5, pp. 1301-1328, 2007.