Data Sharing using Key Aggregate in Cloud Storage

A Sowranika Devi[1] I MadhaviLatha[2]
[1]Student [2]Assistant Professor
[1,2]Department of Computer Applications
[1,2]KMM Institute of PG Studies, Tirupati, India

Abstract— Distributed storage is a capacity of data on-line in cloud that is available from various and associated assets. Distributed storage is the capacity that offers keen openness and constancy, tough security, catastrophe recuperation, and most reduced cost. Distributed storage certainly has critical common sense for example safely, imparting data to other people, with incredible effectiveness. New public key cryptography is presented that is named as Key aggregate cryptosystem (KAC). Key-aggregate cryptography system yield steady size ciphertexts with the goal that dependable surrender of decoding rights for a readied cluster of ciphertexts is attainable. Any arrangement of mystery keys might be mass made and structure a solitary key that includes intensity of all the keys being mass formed. This blend key might be sent to the others over a verified channel, and remaining encoded records are immaculate and stay secret. The framework can possibly use verified utilization of cloud framework.

Keywords: Data Sharing, Key Aggregation, and Java Key Store, Private Key Cryptography, Encryption, and Decryption

I. INTRODUCTION

Distributed storage is today entirely in vogue stockpiling framework. Distributed storage is putting away of learning off-site to the physical stockpiling that is kept up by outsider. Distributed storage is sparing of computerized data in intelligent pool and physical stockpiling conveying loads on different servers that are oversee by outsider. Outsider is subject for keeping data available and open and physical air should be ensured and running in the scarcest degree time. Instead of putting away data to the circle drive or the other local stockpiling, we will in general spare data to remote stockpiling which is open from wherever and whenever. It diminishes endeavors of conveying physical capacity to everyplace. By abuse distributed storage we can get to information from any pc through web that discarded confinement of getting to information from same pc where it's kept. While thinking about data (content) security and insurance, goals is to encipher data before transferring to the server with user's claim key. Data sharing is yet again essential usefulness of distributed storage, because of client will share data from wherever and whenever to anybody. For example, association may give consent to get to a piece of delicate data to their staff. Anyway troublesome errand is to share scrambled data safely. Customary way is client will exchange the scrambled data from capacity, decode that data and send it to impart to other people; anyway it loses the significance of distributed storage.

Cryptography technique is connected in a 2 noteworthy ways one is even key cryptography and other one is uneven key encryption. In even key cryptography, same keys are utilized for cryptography and coding. Against this, in uneven key encryption entirely unexpected keys are utilized, open key for cryptography and individual key for coding. Misuse uneven key cryptography is moreover adaptable for our methodology. This could be delineated by following model. Assume Alice places all data on demo.com and she or he doesn't have to demonstrate her data to everyone. Because of data overflowing prospects she doesn't trust on security instrument given by demo.com, subsequently she encipher all data before transferring to the server. On the off chance that Bob raise her to share some data, at that point Alice use share works of demo.com. Anyway downside sense's the best approach to share scrambled data.

II. RELATED WORK

A. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data

As more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. One drawback of encrypting data, is that it can be selectively shared only at a coarse-grained level (i.e., giving another party your private key). We develop a new cryptosystem for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE). In our cryptosystem, ciphertexts are labeled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user is able to decrypt. We demonstrate the applicability of our construction to sharing of audit-log information and broadcast encryption. Our construction supports delegation of private keys which subsumes Hierarchical Identity-Based Encryption (HIBE).

B. Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records

We explore the challenge of preserving patients' privacy in electronic health record systems. We argue that security in such systems should be enforced via encryption as well as access control. Furthermore, we argue for approaches that enable patients to generate and store encryption keys, so that the patients' privacy is protected should the host data center be compromised. The standard argument against such an approach is that encryption would interfere with the functionality of the system. However, we show that we can build an efficient system that allows patients both to share partial access rights with others, and to perform searches over their records. We formalize the requirements of a Patient Controlled Encryption scheme, and give several instantiations, based on existing cryptographic primitives and protocols, each achieving a different set of properties.

C. Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage

Data sharing is an important functionality in cloud storage. In this paper, we show how to securely, efficiently, and flexibly share data with others in cloud storage. We describe new

public-key cryptosystems that produce constant-size ciphertexts such that efficient delegation of decryption rights for any set of ciphertexts are possible. The novelty is that one can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. In other words, the secret key holder can release a constant-size aggregate key for flexible choices of ciphertext set in cloud storage, but the other encrypted files outside the set remain confidential. This compact aggregate key can be conveniently sent to others or be stored in a smart card with very limited secure storage. We provide formal security analysis of our schemes in the standard model. We also describe other application of our schemes. In particular, our schemes give the first public-key patient-controlled encryption for flexible hierarchy, which was yet to be known.

III. PROPOSED SYSTEM

In proposed system, we're going to consciousness how we will make private key cryptography secure to proportion encrypted records amongst customers. It is very critical now not to proportion a users' personal key to different customers. So we have to percentage the encrypted records to users without sending the personal key of the information owner .The most important gain of AES is that it handles encryption of huge facts successfully and the overall performance is much better than compare to different encryption algorithms. The major advantage of AES is that it handles encryption of large data efficiently and the performance is much higher than compare to other encryption algorithms. We introduce a new private key cryptography algorithm where the size of the secret key remains constant. The new private key cryptography algorithm is called as Key Aggregation cryptography. It provides us efficiently and securely data sharing mechanism in our cloud storage.

IV. PROPOSED ALGORITHMS

A. Blowfish Algorithm

- Blowfish was developed by Bruce Schneier. It is very strong symmetric key cryptographic algorithm.

B. Features of Blowfish

- Fast: Blowfish encryption state on 32 bit microprocessors is 26 clock cycles per byte
- Compact: Blowfish can execute is less than 5KB memory
- Simple: Blowfish uses only primitive operations such as addition, XOR and table look up making its design and manipulation simple
- Secure: Blowfish has a variable key length up to a maximum of 448 long, making it both flexible and secure

C. Operations: (Blowfish encrypts 64-bit block with a variable length key)

1) Sub key Generation

This process covert the key up to 448 bit long to sub keys totaling 4168 bits

2) Data Encryption

This process involves the iteration of a simple function 16 times. Each round contains a key dependent permutation and key and data substitution

- Blowfish is a very fast algorithm which takes 64 bit input as plaintext and generates 64 bit output cipher text
- It uses the concept of P-array which use of 21 bit and there are 18 P-arrays P1 to P18P1 to P18
- Blowfish Algorithm runs 16 times i.e. 16 rounds

D. Processes:

1) Sub key Generation

- Key Size is variable but blowfish algorithm generates very large sub-keys .The key size is in the range of 32 bits to 448 bits or 14 words.
- Concept of P-array consists of 18, 32 bit sub-keys
- There are 4 S-boxes containing 256 entries of 32 bits
- P-array is initialized first then four s boxes with fixed string
- Then P-arryas are XORed with subkeys ie from P1 to P18P1 to P18. Once the sub keys are generated the encryption process begins.

2) Data encryption and decryption:

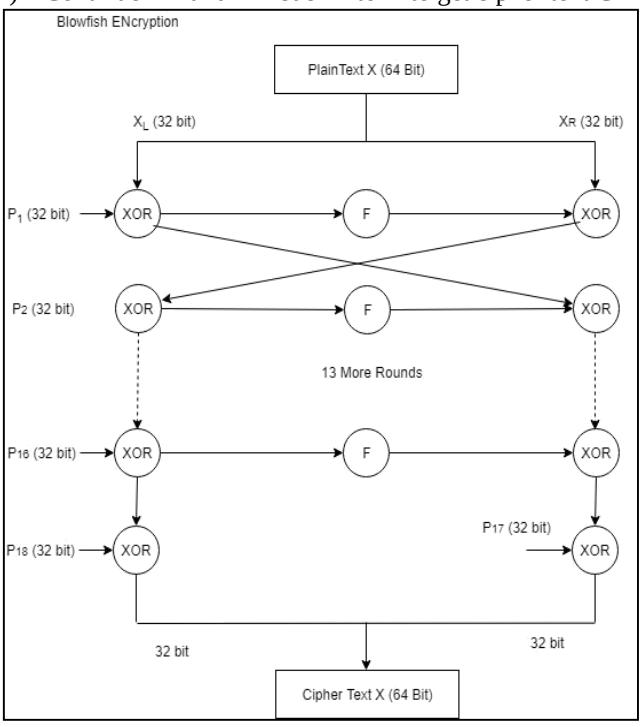
- We use the P arrays and S boxes during this process

E. Algorithm for encryption of 64 bit block

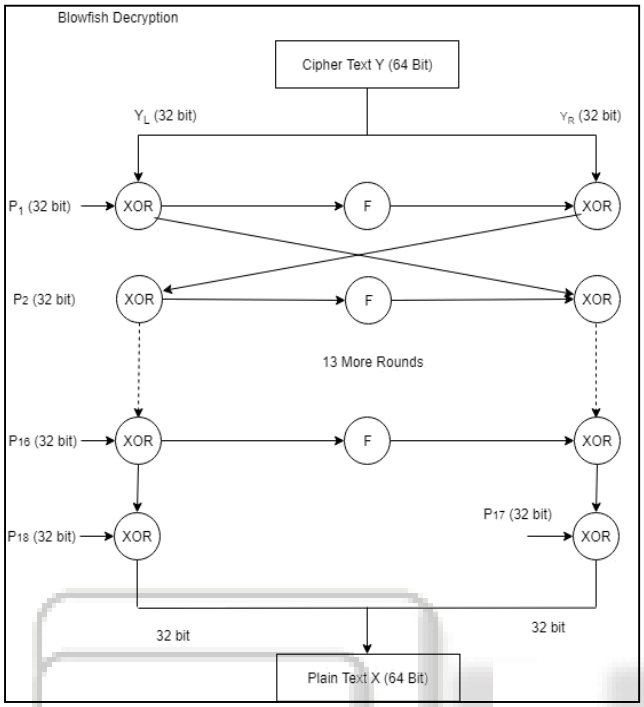
1) Divide X into two blocks CL and XR of equal sizes. Thus both XL and XR will consist of 32 bit each
2) For P=1 to 16

XL = XL XOR PiPi
XR = f(XL) XORXR
Swap XL ,XR
Next i

1) Swap XL, XR XOR P18P18
2) XL = XL XOR P18P18
3) XR = XR XOR P17P17
4) Continue XL and XR back into X to get cipher text CT

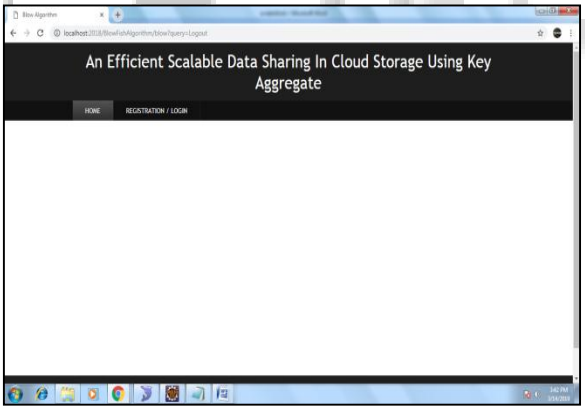


– Function f is as follows
1) Divide the 32 bit XL block into four 8 bit sub blocks named a, b, c, d
2) Compute f(a,b,c,d) = ((S1, a + S2, b) XOR S3, c) XORSc , d
– Function F In Blowfish

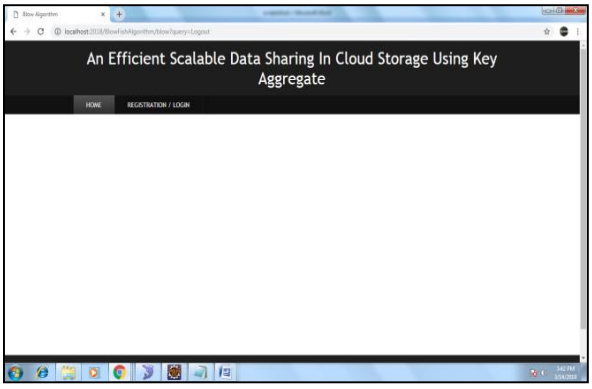


V. RESULTS AND ANALYSIS

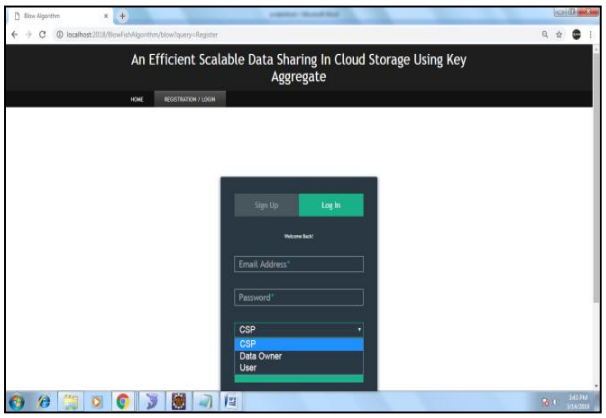
A. *Home*



B. *Registration-Page*

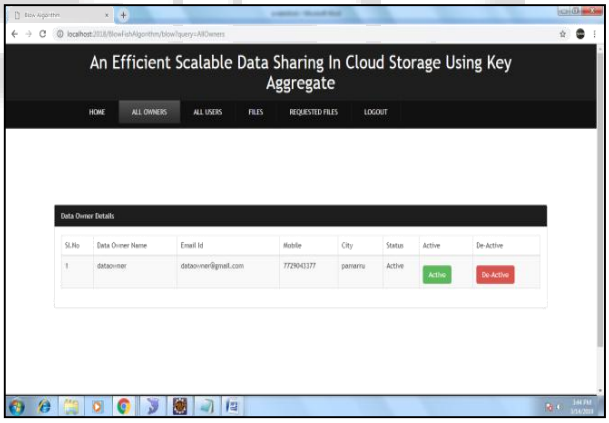


C. *Login-page*

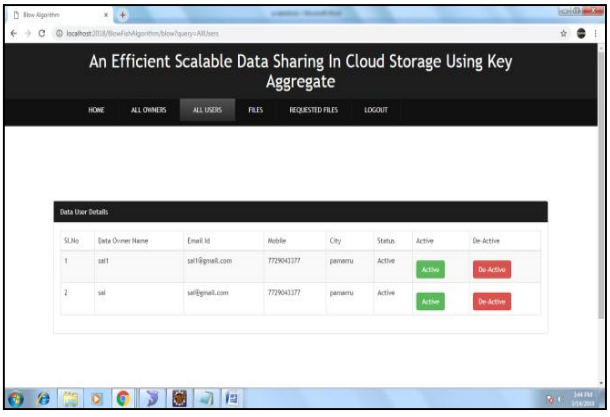


D. *CSP-Home*

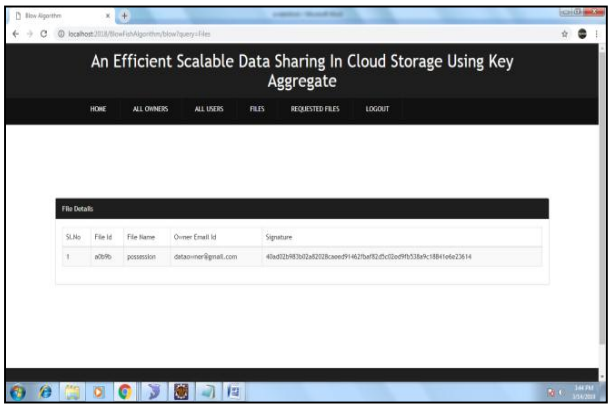
E. *All-Owners*



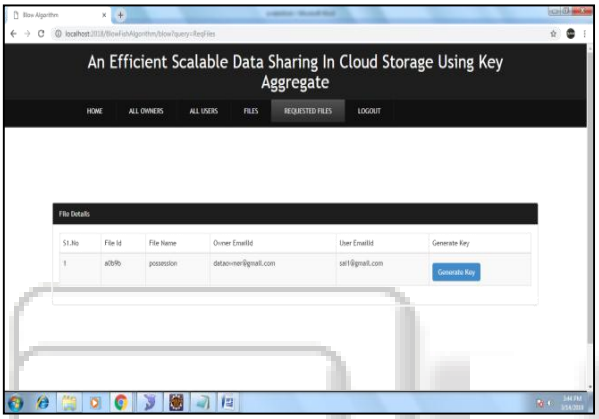
F. *All-Users*



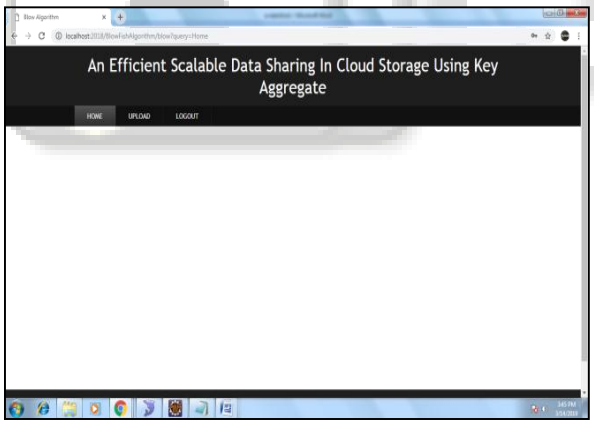
G. Files



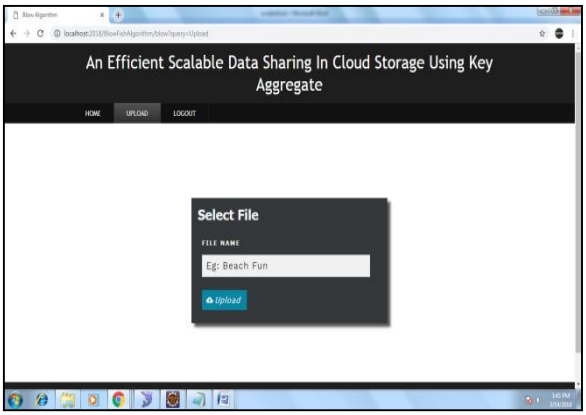
H. Requested-Files



I. Data-Owner-Home

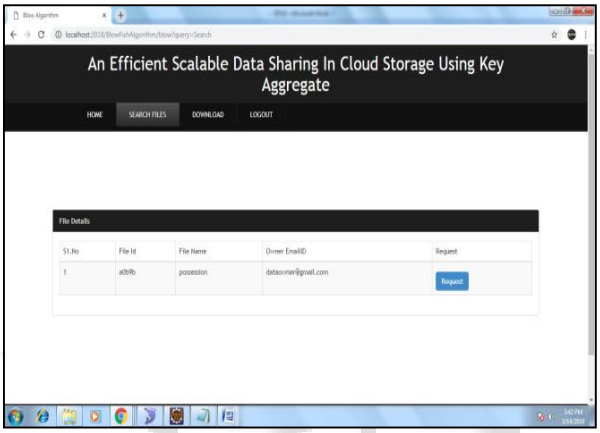


J. Upload

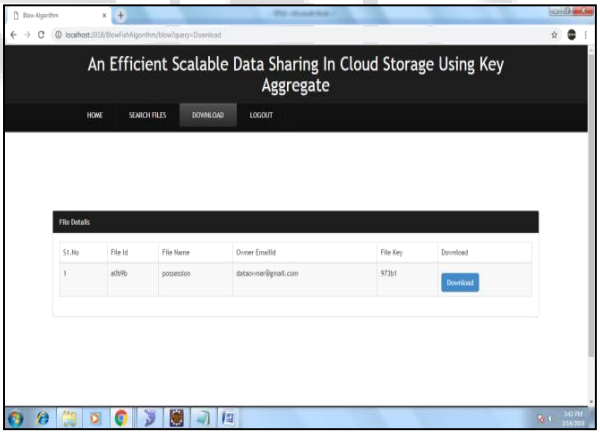


K. User-Home

L. Search-Files



M. Download



VI. CONCLUSION

Users‟ information (content/information) protection could be a focal inquiry of distributed storage. In this, we underscore Compression of the mystery enters in open key cryptosystems that help different figure content classes in distributed storage. It could be any of the capacity set of classes, no issues; the agent will everlastingly get blend key of consistent size. In distributed storage, the amount of figure messages in some cases develops expediently with none limitations. Hence we've to arrange enough figure content classes for the more extended term augmentation. Else, we ought to extend the open key.

REFERENCES

[1] Cheng-Kang Chu, Sherman S. M, "Key Aggregate Cryptosystem for Scalable Data Sharing in cloud storage", IEEE Transactions on Parallel and Distributed Systems, vol. 25, issue2, 2014.
[2] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," in Proceedings of ACM Workshop on Cloud Computing Security (CCSW ‟09). ACM, 2009, pp. 103–114.
[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data," in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS ‟06). ACM, 2006, pp. 89–98 .
[4] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," in Proceedings of Information Security and Cryptology (Inscrypt ‟07), ser. LNCS, vol. 4990. Springer, 2007, pp. 384–398.
[5] M. Chase and S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proceeding ACM Conference on Computer and Communication Security, pp. 121- 130. 2009.
[6] S. G. Akl and P. D. Taylor, "Cryptographic Solution to a Problemof Access Control in a Hierarchy," ACM Transactions on Computer Systems (TOCS), vol. 1, no. 3, pp. 239–248, 1983.
[7] G. C. Chick and S. E. Tavares, "Flexible Access Control with Master Keys," in Proceedings of Advances in Cryptology – CRYPTO ‟89, ser. LNCS, vol. 435. Springer, 1989, pp. 316–322.
[8] W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 14, no. 1, pp. 182– 188, 2002.
[9] R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, vol. 27, no. 2, pp. 95– 98, 1988.
[10] Y. Sun and K. J. R. Liu, "Scalable Hierarchical Access Control in Secure Group Communications," in Proceedings of the 23th IEEE International Conference on Computer Communications (INFOCOM ‟04). IEEE, 2004.
[11] D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing," in Proceedings of Advances in Cryptology – CRYPTO ‟01, ser. LNCS, vol. 2139. Springer, 2001, pp. 213–229.
[12] Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in Proceedings of Advances in Cryptology - EUROCRYPT ‟05, ser. LNCS, vol. 3494. Springer, 2005, pp. 457–473.
[13] S. S. M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions," in ACM Conference on Computer and Communications Security, 2010, pp. 152–161

