International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795

KEY AGGREGATE CRYPTOSYSTEM FOR SCALABLE DATA SHARING IN CLOUD

Priyadharshini.V[1]*, Post Graduate Student*

M.Sc Computer Science

Dr.K.A Jayabalaji[2]*, Assistant .Professor*

Department of Software System,

Sri Krishna Arts and Science College,

ABSTRACT: *In an age of increasing information generation and storage needs, sharing valuable information and security in the cloud has become a major challenge. Key collection cryptography (KAC)has emerged as a promising solution to this problem by allowing users to securely access multiple encrypted files, reducing the encryption overhead. This brief introduces a new key clustered encryption system (KAC-SDS) for cloud data sharing that uses advanced encryption techniques to ensure security and efficient data sharing in the cloud environment.*

Keyword: Cloud, Cloud Storage, Security, Key aggregate decryption, Key aggregate encryption.

1.INTRODUCTION

As the volume of data created and stored in the cloud continues to grow, sharing valuable data and security has become a major concern for people and organizations.Traditional data sharing methods relyon priority management systems and carry high overhead, which hinders data connectivity between multiple users.To solve these problems,key collection cryptography (KAC) has emerged as a promising cryptographic solution to facilitate data sharing and security in cloud-based environments. The key collection encryption system is designed tosimplify the data sharing process by allowing data owners to grant access more effectively. Instead of assigning a key to each user or file,KAC allows data owners to send a single key to gain access to multiple encrypted files.This process not only reduces the burden of switch management,but also improves the scalability of data sharing in the cloud environment.The purpose of this research is to propos e and explore the design and implementation of a new key collection crypto system for scalable data sharing in the cloud.

1.2. LITERATURE REVIEW

[1]"A key collection encryption system for scalable data sharing in cloud storage" (2013), A. Sahai, B. Waters. This map illustrates the concept of KAC and shows its structure as a bilinear map. It lays the foundation for efficient and secure data sharing in cloud storage.

[2]"Attribute-Based Encryption for Detailed Access Control of Encrypted Data" (2010) V. Goyal, O. Pandey, A.Sahai and B. Waters.This document introduces the Attribute-Based Encryption (ABE) concept and its application for effective access control. It forms the basis for the integration of ABE into KAC to extend the input strategy.

[3]"External Attribute-Based Encryption" (2014), R. Lu, X. Lin, X. Liang, and X.

S.shen.This document proposes behavior-based encryption that allows data subjects to send their access decisions to third parties based on the idea of KAC delegation.

[4]"Dynamic Secure Cloud Storage by Source" (2018), S. Kamara, K.Lauter and Zhao Y. This document provides a Dynamic Provable Data Ownership (PDP) scheme for ensuring data integrityand security updates in KAC situations.

[5]"Multi-authority-based encryption with an honest but curious central authority" (2018), Z.Cao, J. Yan,J. Cui, M. Wu and H. Lin.This work extends character-based encryption to support multiple authorizations, which is important for improving the scalability and delivery of KAC.

[6]"A strategic key collection cryptosystem with broadcast aggregation keys for scalable data sharing in cloud storage" (2014), A. Lewko and B. Waters.This document introduces a variant of KAC with shared key distribution that can reduce communication overhead in big data sharing scenarios.

[7]"A Reversible Key Collection Crypto System for Secure Data Sharing in Cloud Storage" (2016), L. Xu,Z. Cao, X.Lin and X.S. shen.This study introduces the removal of KAC, which allows data subjects to remove access to certain users.

[8]“Fine Grain Data Access Control in Provably Secure Multi-Clouds” (2013), los ntawm W. Sun, Y.T. Wu, S. H.Xu and K.C. lan.Based on the nature of cloud computing, this document explores the concept of granular data access to manage multiple instances.

2.EXISTING SYSTEM

The current KAC system offers the basic concept ofcollecting encryption keys to enable data sharing in a cloud storage environment. It uses binary linear regression in the basic mathematical model to use aggregate values. In this system, the data owner generates a master public key (MPK) and a master private key (MPSK) during installation. The data owner then encrypts the personal data files using symmetric encryption and stores them in cloud storage. Key collection in KAC involves combiningthe data's unique encryption key and access operation (ie, key file) using MPK.For example, customers).The result is a combined key that provides access to multiple encrypted files without requiring a separate decryption key for eachfile. Authorized users receive bulk keys from dataowners or trusted sources. Users using batch keys can decide which encrypted data they can access based on the access control rules defined during thebatch process.

DRAWBACKS OF EXISTING SYSTEM:

One master file.

Restricted access to administration.

Difficulty in extracting key.

Communication and storage overhead.

3.PROPOSED SYSTEM

In a cloud environment, it is important to share different information for effective collaboration among multiple users. Traditional data sharing methods often involve assigning a key to each encrypted file, making the process cumbersome and inefficient. As the volume of data and the number of users increase, the management of values becomes amajor challenge. To solve this problem, key collection cryptography (KAC) is proposed as a solution for data sharing in the cloud.

ADVANTAGES OF PROPOSED SYSTEM:

We can also use general aggregation to share encrypted data.

The master key is fixed.

There is no special relationship between classes.

4.PROBLEM DEFINITION

The problem definition of Key Collection Cryptography (KAC) in the context of scalable datasharing in the cloud revolves around the challenges of efficient and secure data sharing in the cloud storage environment. As data volumes continue to increase, traditional encryption methods become ineffective due to the need to manage individual encryption keys for each user. This increases computing overhead and storage costs, hindering integration and access to data between multiple users. The problem KAC needs to solve is how to create an encryption solution that allows data owners to share data with multiple recipients in cloud-based system, while reducing the hassle of curating critical management and personal information. The main goal is to combine multiple ciphertexts into a single ciphertext agreement and create a unified code that allows users to decipher the information.In addition, KAC should support effective access control to allow access based on user properties and to allow modification of important actions and deletions for users' access rights.

5.KEY AGGREGATE CRYPTOSYSTEM

Key Aggregate Cryptosystem (KAC) represents an advance in the field of easily distributed data in cloud environments. With the growth of data and theincreasing number of users, traditional data encryption methods are difficult to use and easy to access. KAC overcomes this challenge by introducing a new concept of "bulk key" by changing the way data is encrypted, shared, and accessed in the cloud uses an encryption method that contains personal information and public keys at its core. When data owners upload files to the cloud, all files are encrypted with unique data. Instead of storing data separately for each file or user, KAC allows data owners to aggregate this data into a single set of keys that represent a shared combination of many users and data. This integration reduces the number of encryption keys required, simplifies key management processes, andenables data integration in the cloud.A key feature of the KAC is quality control support. Data owners can grant access to certain groups of users via shared content links. Each user is then equipped with the keys necessary to access the encrypted data they are authorized to view or modify.This approach ensures that data remains private from unauthorized users, enabling secure collaboration and data sharingacross multiple weather scenarios.In addition, KAC provides an effective access revocation mechanism. When the user's login is removed, the data owner can replace the corresponding key, overriding the previous key. This dynamic approach to management eliminates the need to re-encrypt entire datasets to provide real-time data management access.The versatility of the KAC makes it an invaluable tool in many data-intensive field.Also, in a distributed system KAC improves the performance of data sharing between multiple users, making it the best solution for collaboration, shared office space, and enterprise, KAC makes it easy to distribute content and make incompatible data connections between separate domains.Although KAC has many advantages, its safety implications must be carefully considered. Researchers are constantly working to improve cryptography security to strengthen KAC against attacks and vulnerabilities. The design of the KAC algorithm should take precedence over known security measures to ensure that sensitive data is protected in the cloud.As the demand for data sharing solutions continues to grow, KAC remains the focus of research and development. Experts in the field are working on improving the KAC algorithm, improving its performance and making it compatible with existing weather systems. In addition, modeling efforts and collaboration between researchers, industry and policy makers have played an important role in enabling KAC to provide reliable and widely used data sharing in the cloud.KAC offers key collection and efficient management, simplifying key management, increasing security and facilitating collaboration in the cloud data environment.KAC is poised to become the future of information sharing, supporting the growth of cloud computing and providing organizations with a secure and efficient information access environment.

5.1.ALGORITHMIC STEPS IN KAC

STEP 1:

SETUP:

DESCRIPTION: The data owner takes the first action to set up the KAC system. During setup, the data manager generates a master public key (MPK)and a master private key (MPSK) using appropriate cryptographic primitives such as binary mapping Or elliptic curve cryptography. MPK is public while MPSK is reserved and used for key derivation,

STEP 2:

KEY GENERATION:

DESCRIPTION: In Key Geneartion,generate the public key and private key for the data owner.

STEP 3:

DATA ENCRYPTION:

DESCRIPTION: In the data encryption algorithm, the data owner uses appropriate encryption techniques to encrypt the data.

STEP 4:

AGGREGATE KEY GENERATION:

DESCRIPTION: The Aggregate key (AK) is generated as the product of the bilinear mapping of each user's authorized public key, and is the hash ofthe block identifier that hash the increment to a random user's strength hidden value.

STEP 5:

AGGREGATE KEY ENCRYPTION:

DESCRIPTION: The Aggregate Key Encryption step allows the data subject to securely encrypt the BulkKey (AK) in each user's authorized public key. This gives users access to blocks of data, allowing for good results and specifying their specific content.

STEP 6:

DATA DECRYPTION:

DESCRIPTION: Data decryption involves using a combined key to achieve the best results of multipleencrypted files.

5.2.CONSTRUCTION OF KAC

STEP1:

- Choose an appropriate elliptic curve group and a bilinear pairing function e.

- Generate a master public parameter P and a master secret parameter a.

STEP2:

- Key generation for user I,
- **INPUT:**User identifier i.
- Compute the private key and public key for user i: private_key_i = a * i (scalar multiplication)
- public_key_i = P * i (scalar multiplication)
- **OUTPUT:**private_key_i and public_key_i for user i.

STEP3:

- Data Encryption for Data Block b,
- **INPUT:**Data block b and a random symmetric encryption key K_b.
- Encrypt the data block b using the symmetric key K_b to produce the ciphertext C_b.
- **OUTPUT:**C_b, the encrypted data block.

STEP4:

- In aggregate key generation,
- **INPUT:**Set of authorized user identifiers A and data block identifier ID_b.
- For each user i in A, compute the following:
- Compute H(ID_b), where H is the cryptographic hash function, and ID_b is the identifier of the data block.
- Compute α_i, a random secret value associated with user i.
- Compute e(public_key_i, H(ID_b))^α_i, where public_key_i is the public key of user i.
- Aggregate the computed values from all authorized users:
- AK = ∏(e(public_key_i, H(ID_b)))^α_i, where the product is taken over all authorized users in A.
- **OUTPUT:**AK, the aggregate key

STEP5:

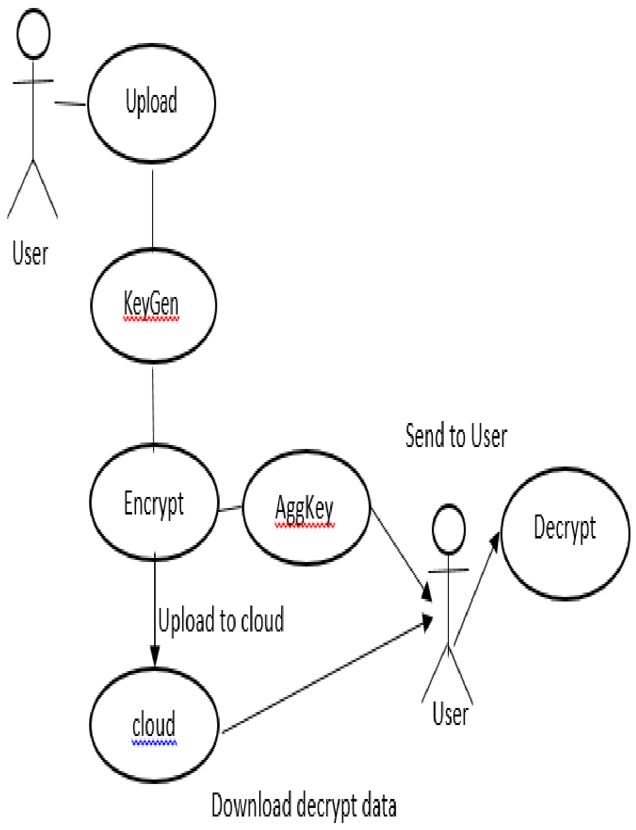
- Aggregate Key Encryption for User I,
- **INPUT:**AK (the aggregate key) and public_key_i (the public key of user i).
- Encrypt the aggregate key AK under the public key public_key_i to generate the encrypted aggregate key EK_i.
- EK_i = Enc(public_key_i, AK), where Enc() represents the asymmetric encryption process.
- **OUTPUT:**EK_i, the encrypted aggregate key for user i.

STEP6:

- In data encryption,
- Data owner shares the encrypted data blocks (C_b) and their corresponding encrypted aggregate keys (EK_i) with the cloud storage service.
- An authorized user retrieves the necessary encrypted data blocks and encrypted aggregate key EK_i.
- The user decrypts their specific encrypted aggregate key EK_i using their private key private_key_i to obtain AK.

- With AK in hand, the user can efficiently decrypt the encrypted symmetric keys K_b associated with the data blocks they are authorized to access.
- The user can then use K_b to decrypt the corresponding data blocks and access the original data.

6.FLOW DIAGRAM



7.CONCLUSION

Key aggregate cryptosystem have a number of advantages that make them a good choice for cloud-based data sharing situations.Efficiency is another important aspect of KAC. By implementing binary matching, elliptic curve cryptography and cryptographic hash functions, KAC optimizes the key collection process, reducing computational overhead and communication. This is ideal for large data sharing applications where data management is critical.Data privacy is important in any data sharingprocess and KAC is good at it. Through multiple layers of encryption, KAC ensures that data is kept private and accessible only to authorized users.Symmetric encryption of data blocks, combined with encryption of all keys in each user's public key, provides effective protection against data breaches and unauthorized access attempts yes. Scalability is another key factor that makes KAC a great solution. A cloud-based system that increases data count and user interactions, KAC is designed tomeasure quality with users and data blocks. This scalability is necessary to manage shared data efficiently and flexibly according to different needs.In addition, KAC provides ease of updating and deleting user information.Data owners can change data or remove access to specific users without re-encrypting all data. This agility ensures that data sharing operations remain seamless and responsive to changing data management needs.Overall, key collection encryption systems area strong and future-proof encryption solution for scalable data sharing in the cloud. The combination of management quality, efficiency, data privacy, scalability and flexibility makes it the best choice for using the cloud to improve integrated data sharing while maintaining data security and privacy.With the continuous development of the cloud industry, KAC has always been at the forefront of secure and efficient data sharing, providing a reliable foundation for the future cloud based on data management.

8.FUTURE WORK

Future work of key aggregate cryptography (KAC) for scalable data sharing in the cloud has great potential to improve cutting-edge cloud security andefficiency. A key area of focus is optimizing KAC's computing efficiency. Researchers can explore new algorithms and encryption techniques to reduce the computational load involved in key processing, encryption, and decryption. By simplifying these tasks, KAC can achieve greater scalability and be more adaptable to more data and users in the cloud environment.Furthermore, improving the stability ofKAC remains an important area for future research.Although KAC is designed to resist a variety of cryptographic attacks, more effort is needed to verify and prove its security features. By identifying potential vulnerabilities and strengthening defenses, KAC can provide security and increase trust between users and data owners. Additionally, future work may focus on addressing data changes in KAC. As cloud-based data evolves over time, it is important to properly adapt to data updates and user deletion processes. Finding ways toadd or remove blocks of data without requiring full re-encryption and ensuring that no users are deleted can increase KAC's flexibility and reduce management complexity for data owners and cloud service providers.Collaboration and modeling efforts are also critical to the future development of KAC. Collaborative research to develop a similar model for KAC can contribute to wider and more efficient use of different clouds and services. This will encourage integration and collaboration to secure the shared data ecosystem and encourage theexpansion and use of KAC in different environments.As a result, future work on key aggregation cryptosystems for scalable data sharingin the cloud will lead to successful growth in cloud security, efficiency and availability. Increasing computing efficiency, strengthening security measures, addressing the power of information, and encouraging collaboration through design are priorities that researchers and physicians need to explore.These efforts could further strengthen KAC's position as a powerful and versatile encryption tool that enables secure and seamless data sharing in a changing world.

9.REFERENCES

[1]A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in Advances in Cryptology – EUROCRYPT 2005.
[2]A. Lewko and B. Waters, "Decentralizing Attribute-Based Encryption," in Proceedings of the 29th Annual International Conference on the Theory and the Applications of Cryptographic Techniques – EUROCRYPT 2010.
[3]A. Sahai, B. Waters, and B. F. Wu, "Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption," in Proceedings of the 2014 ACM Conference on Computer and the Communications Security – CCS 2014.
[4]B. Libert, M. Joye, and P. Paillier, "An Efficient Threshold Public Key Cryptosystem Secure Against Adaptive Chosen Ciphertext Attack," in Proceedings of the 26th International Conference on Theory and Applications in Cryptographic Techniques – EUROCRYPT 2007.
[5]G. C. Chick and S. E. Tavares, "Flexible Access Control with Master Keys," in Proceedings the Advances in Cryptology – CRYPTO '89, ser. LNCS, vol. 435. Springer, 1989, pp. 316–322.
[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]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.
[8]L.B. Oliveira, D. Aranha, E. Morais, F. Daguano, J. Lopez, and R. Dahab, "TinyTate: Computing the Tate Pairing in ResourceConstrained Sensor Nodes," Proc. IEEE Sixth Int'l Symp. Network Computing and the Applications (NCA '07), pp. 318-323, 2007.

[9]V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-Based Encryption for the Fine-Grained Access Control of Encrypted data,” in Proceedings of the 13th ACM Conference on Computer and Com-munications Security (CCS ’06). ACM, 2006, pp. 89-98.
[10]M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, “Dynamic and Efficient Key Management for the Access Hierarchies,” ACM Transactions on Information and the System Security (TISSEC), vol. 12, no. 3, 2009
[11]T. Okamoto and K. Takashima, Achieving Short Cipher texts or the Short Secret-Keys for Adaptively Secure General Inner-Product Encryption, in Cryptology and Network Security (CANS 11), 2011, pp. 138159.
[12]G. Ateniese, K. Fu, M. Green, and S. Hohenberger, Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage, ACM Transactions on Information and System Security (TISSEC), vol. 9, no. 1, pp. 130, 2006.
[13]T. H. Yuen, S. S. M. Chow, Y. Zhang, and S. M. Yiu, Identity- Based Encryption Resilient to Continual Auxiliary Leakage, in Proceedings of Advances in Cryptology – EUROCRYPT 12, ser.LNCS, vol. 7237, 2012, pp. 117134.
[14]V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-Based on Encryption for Fine-Grained Access Control of Encrypted data,” in Proceedings the 13th ACM Conference on Computer and Communications Security (CCS ’06). ACM, 2006, pp. 89–98
[15]F. Guo, Y. Mu, and Z. Chen, “Identity-Based Encryption: How to Decrypt Multiple Cipher texts Using a Single Decryption Key,” in Proceedings of Pairing-Based Cryptography (Pairing ’07), ser. LNCS, vol. 4575. Springer, 2007, pp. 392–406.
[16]R. S. Sandhu, “Cryptographic Implementation of the Tree Hierarchy for Access Control,” Information Processing Letters, vol. 27, no. 2, pp. 95–98, 1988.
[17]R. Canetti and S. Hohenberger, “Chosen-Ciphertext Secure Proxy Re-Encryption,” in Proceedings of the 14th ACM Conference on the Computer and Communications Security (CCS ’07). ACM, 2007, pp. 185–194
[18]D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from the Bilinear Maps,” in Proceedings of Advances in Cryptology - EUROCRYPT ’03, ser. LNCS, vol. 2656. Springer, 2003, pp. 416–432.