**PUBLIC KEY ENCRYPTION WIYTH MULTICIPHER USING CLOUD COMPUTING**

**M Biju Balakrishnan.**

Dept of Computer Science and Engineering,

Assistant Professor, Dept of Computer Science and Engineering,

Hindusthan College of Engineering and Technology, Coimbatore, Tamilnadu, INDIA

Abstract -Cloud-based data storage service has drawn increasing interests from both academic and industry in the recent years due to its efficient and low cost management. Since it provides services in an open network, it is urgent for service providers to make use of secure data storage and sharing mechanism to ensure data confidentiality and service user privacy. To protect sensitive data from being compromised, the most widely used method is encryption. However, simply encrypting data (e.g., via AES) cannot fully address the practical need of data management. Besides, an effective access control over download request also needs to be considered so that Economic Denial of Sustainability (EDoS) attacks cannot be launched to hinder users from enjoying service. In this paper, we consider the dual access control, in the context of cloud-based storage, in the sense that we design a control mechanism over both data access and download request without loss of security and efficiency. Two dual access control systems are designed in this paper, where each of them is for a distinct designed setting. The security and experimental analysis for the systems are also presented.

**INTRODUCTION**

In the recent decades, cloud-based storage service has attracted considerable attention from both academia and industries. It may be widely used in many Internet-based commercial applications (e.g., Apple iCould) due to its long-list benefits including access flexibility and free of local data management. Increasing number of individuals and companies nowadays prefer to outsource their data to remote cloud in such a way that they may reduce the cost of upgrading their local data management facilities/devices. However, the worry of security breach over outsourced data may be one of the main obstacles hindering Internet users from widely using cloud-based storage service.

In many practical applications, outsourced data may need to be further shared with others. For example, a Dropbox user Alice may share photos with her friends. Without using data encryption, prior to sharing the photos, Alice needs to generate a sharing link and further share the link with friends. Although guaranteeing some level of access control over unauthorized users (e.g., those are not Alice’s friends), the sharing link may be visible within the Dropbox administration level (e.g., administrator could reach the link). Since the cloud (which is deployed in an open network) is not be fully trusted, it is generally recommended to encrypt the data prior to being uploaded to the cloud to ensure data security and privacy. One of the corresponding solutions is to directly employ an encryption technique (e.g., AES) on the outsourced data before uploading to cloud, so that only specified cloud user (with valid decryption key) can gain access to the data via valid decryption

**II. RELATED STUDIES**

.

**III. METHODOLOGY**

1. Select the encryption algorithms/ciphers: Choose the ciphers that you want to use for encrypting the data. Some commonly used ciphers are RSA, AES, and Blowfish.
2. Generate public and private keys: Generate a public key and a private key for each cipher that you want to use. Ensure that the keys are generated securely and that the private keys are kept confidential.
3. Store the public keys on the cloud server: Store the public keys on the cloud server so that anyone can access them.
4. Encrypt the data: Encrypt the data using one or more of the ciphers and the corresponding public key(s). Ensure that the encryption process is secure and that the encrypted data is protected during transmission and at rest.
5. Store the encrypted data on the cloud server: Store the encrypted data on the cloud server, ensuring that it is stored securely and that access controls are in place to prevent unauthorized access.
6. Retrieve the private keys: Retrieve the corresponding private key(s) from the cloud server to decrypt the data. Ensure that the private keys are retrieved securely and that access controls are in place to prevent unauthorized access.
7. Decrypt the data: Use the private key(s) to decrypt the data. Ensure that the decryption process is secure and that the decrypted data is protected during transmission and at rest.
8. Repeat for each cipher used: Repeat the process for each cipher used to encrypt the data.

The methodology for implementing public key encryption with multiple ciphers using cloud computing should also include considerations for security, scalability, and cost. For example, security considerations might include using secure communication protocols, implementing access controls, and following best practices for key management. Scalability considerations might include selecting a cloud provider that can scale with the needs of the user, while cost considerations might include optimizing cloud server usage and minimizing the need for additional hardware or software.

**IV. EXPERIMENTAL RESULT:**

Computer output is the most important and direct information source to the user. Output design is a process that involves designing necessary outputs in the form of reports that should be given to the users according to the requirements. Efficient, intelligible output design should improve the system's relationship with the user and help in decision making. Since the reports are directing referred by the management for taking decisions and to draw conclusions they must be designed with almost care and the details in the reports must be simple, descriptive and clear to the user. So while designing output the following things are to be considered.

Determine what information to present Arrange the presentation of information in an acceptable format Decide how to distribute the output to intended receipts Depending on the nature and future use of output required, they can be displayed on the monitor for immediate need and for obtaining the hardcopy. The options for the output reports are given in the appendix.

**V. CONCLUSION:**

We addressed an interesting and long-lasting problem in cloud-based data sharing, and presented two dual access control systems. The proposed systems are resistant to DDoS/EDoS attacks. We state that the technique used to achieve the feature of control on download request is “transplantable” to other CP-ABE constructions. Our experimental results show that the proposed systems do not impose any significant computational and communication overhead (compared to its underlying CP-ABE building block).

In our enhanced system, we employ the fact that the secret information loaded into the enclave cannot be extracted. However, recent work shows that enclave may leak some amounts of its secret(s) to a malicious host through the memory access patterns or other related side-channel attacks. The model of transparent enclave execution is hence introduced in. Constructing a dual access control system for cloud data sharing from transparent enclave is an interesting problem. In our future work, we will consider the corresponding solution to the problem.

**REFERENCES**

. Joseph A Akinyele, Christina Garman, Ian Miers, Matthew W Pagano, Michael Rushanan, Matthew Green, and Aviel D Rubin. Charm: a framework for rapidly prototyping cryptosystems. Journal of Cryptographic Engineering, 3(2):111–128, 2013.

[2] Ittai Anati, Shay Gueron, Simon Johnson, and Vincent Scarlata. Innovative technology for cpu based attestation and sealing. In Workshop on hardware and architectural support for security and privacy (HASP), volume 13, page 7. ACM New York, NY, USA, 2013.

[3] Alexandros Bakas and Antonis Michalas. Modern family: A revocable hybrid encryption scheme based on attribute-based encryption, symmetric searchable encryption and SGX. In SecureComm 2019, pages 472–486, 2019.

[4] Amos Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[5] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In S&P 2007, pages 321–334. IEEE, 2007.

[6] Victor Costan and Srinivas Devadas. Intel sgx explained. IACR Cryptology ePrint Archive, 2016(086):1–118, 2016.

[7] Ben Fisch, Dhinakaran Vinayagamurthy, Dan Boneh, and Sergey Gorbunov. IRON: functional encryption using intel SGX. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, pages 765–782, 2017.

[8] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Advances in Cryptology-CRYPTO 1999, pages 537–554. Springer, 1999.

[9] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In ACM CCS 2006, pages 89–98. ACM, 2006.

[10] Jinguang Han, Willy Susilo, Yi Mu, Jianying Zhou, and Man Ho Allen Au. Improving privacy and security in decentralized ciphertext-policy attribute-based encryption. IEEE transactions on information forensics and security, 10(3):665–678

[11] Christofer Hoff. Cloud computing security: From ddos (distributed denial of service) to edos (economic denial of sustainability). http://www. rationalsurvivability.com/blog/?p=66.

[12] Joseph Idziorek, Mark Tannian, and Doug Jacobson. Attribution of fraudulent resource consumption in the cloud. In IEEE CLOUD 2012, pages 99–106. IEEE, 2012.

[13] Simon Johnson, Vinnie Scarlata, Carlos Rozas, Ernie Brickell, and Frank Mckeen. Intel R software guard extensions: Epid provisioning and attestation services. White Paper, 1:1–10, 2016.

[14] Sangho Lee, Ming-Wei Shih, Prasun Gera, Taesoo Kim, Hyesoon Kim, and Marcus Peinado. Inferring fine-grained control flow inside sgx enclaves with branch shadowing. In 26th USENIX Security Symposium, USENIX Security, pages 16–18, 2017.

[15] Jiguo Li, Xiaonan Lin, Yichen Zhang, and Jinguang Han. Ksfoabe: outsourced attribute based encryption with keyword search function for cloud storage. IEEE Transactions on Services Computing, 10(5):715–725, 2017.

[16] Jiguo Li, Yao Wang, Yichen Zhang, and Jinguang Han. Full verifiability for outsourced decryption in attribute based encryption. IEEE Transactions on Services Computing, DOI:

10.1109/TSC.2017.2710190, 2017.

[17] Wei Li, Kaiping Xue, Yingjie Xue, and Jianan Hong. Tmacs: A robust and verifiable threshold multi-authority access control system in public cloud storage. IEEE Transactions on parallel and distributed systems, 27(5):1484–1496, 2016.

[18] Ben Lynn et al. The pairing-based cryptography library. Internet: crypto. stanford. edu/pbc/[Mar. 27, 2013], 2006.

[19] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V. Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R. Savagaonkar. Innovative instructions and software model for isolated execution. In HASP@ISCA 2013, page 10, 2013.

[20] Antonis Michalas. The lord of the shares: combining attributebased encryption and searchable encryption for flexible data sharing. In SAC 2019, pages 146–155, 2019.