EasyChair Preprint

№ 12759

Securing Cloud Data Under Key Exposure

P Jagadeesan, K Mohan, V Naveen and
A. Mohammad Farmaanullah

March 27, 2024

Securing Cloud Data Under Key Exposure

P.Jagadeesan
Computer Science and Engineering
R.M.D Engineering College
Chennai, India
pjn.cse@rmd.ac.in

K.Mohan
Computer Science and Engineering
R.M.D Engineering College
Chennai, India
ucs20307@rmd.ac.in

V.Naveen
Computer Science and Engineering
R.M.D Engineering College
Chennai, India
ucs20314@rmd.ac.in

A.Mohammad Farmaanullah
Computer Science and Engineering
R.M.D Engineering College
Chennai, India
ucs20305@rmd.ac.in

Abstract - *Recent revelations of a sophisticated attacker have underscored the vulnerability of data privacy, as they have been able to breach encryption by acquiring cryptographic keys through coercion or exploiting weaknesses in cryptographic software. Once these keys are compromised, the only recourse to safeguard data privacy is to restrict the attacker's access to the ciphertext. This can be achieved by dispersing fragments of the encrypted data across multiple servers in diverse administrative domains, assuming that the attacker cannot compromise all of them. Nevertheless, conventional encryption methods still leave data vulnerable, as an attacker with the encryption key can compromise a single server and gain access to the ciphertext blocks stored within it. In response to this pressing challenge, we introduce Bastion, a pioneering and efficient solution designed to protect data privacy even in the event of key exposure and an attacker's access to all ciphertext fragments. We scrutinize Bastion's security features and assess its performance through a prototype implementation. Additionally, we explore practical insights regarding the integration of Bastion into existing distributed storage systems. Our findings suggest that Bastion is well-suited for integration into current systems, as it incurs less than 5% overhead compared to existing semantically secure encryption modes.*

Keywords: **Cryptographic Keys, encryption, Bastion, Information Privacy.**

1. INTRODUCTION

In today's digital Landscape, the adoption of cloud computing has revolutionized the way organizations store and manage data, offering unparalleled scalability, flexibility, and cost-effectiveness. However, with increased reliance on cloud services comes the imperative need to prioritize cloud data security.

2. EXISTING SYSTEM

Despite the implementation of various security measures within the targeted service, perpetrators were able to bypass them. For instance, although these services employed encryption mechanisms to ensure data confidentiality, the necessary encryption keys were obtained through backdoors, bribery, or coercion. If the encryption key becomes compromised, the only effective method to maintain confidentiality is to limit the adversary's access to the ciphertext. This could involve dispersing the ciphertext across multiple administrative domains, with the hope that the adversary cannot compromise all of them. However, even if the data is encrypted and distributed across different administrative domains, an adversary with the appropriate encryption keys can infiltrate a server in one domain and decrypt the ciphertext blocks stored within it.

Disadvantage:

- Various models are employed to analyze and understand the "leaks" that occur in real-world implementations of cryptographic primitives.
- Typically, a file is first encrypted using the Advanced Encryption Standard (AES), and then the encrypted file is dispersed across multiple locations. The encryption key is shared using a designated scheme.
- Deniable encryption, however, is crafted to deceive adversaries who lack knowledge of the "original" encryption key.

3. PROPOSED SYSTEM

We investigate methods to protect data confidentiality against an adversary who possesses both the encryption key and has access to a significant portion of the ciphertext blocks. This adversary could obtain the key through exploiting vulnerabilities or backdoors in key-generation software, or by compromising the devices

storing the keys. This threat compromises the security of many cryptographic solutions, including those using secret-sharing to protect encryption keys, as these keys may be leaked upon generation. To address this challenge, we introduce Bastion, a novel and effective scheme that ensures plaintext data remains unrecoverable even if the adversary has access to all but two ciphertext blocks, even when the encryption key is exposed. Bastion achieves this by integrating standard encryption functions with an efficient linear transform, akin to the concept of an all-or-nothing transform (AONT). While an AONT is not encryption itself, it can be employed as a preprocessing step before encrypting data with a block cipher, a technique known as AON encryption, primarily designed to impede brute-force attacks on the encryption key.

Advantage:

- Bastion is proposed as an efficient solution that safeguards data confidentiality against adversaries with knowledge of the encryption key and access to a significant portion of ciphertext blocks.
- We conduct a comprehensive security analysis of Bastion, demonstrating its capability to prevent leakage of any plaintext block when the adversary has access to the encryption key and all but two ciphertext blocks.
- Practical insights are provided regarding the integration of Bastion into existing storage systems, such as grid storage systems.

4. OBJECTIVE

Our research bears resemblance to the concept of shared key deniable encryption. Deniable encryption refers to a scheme where the legitimate owner, when coerced to disclose the encryption key, reveals "fake keys," thereby causing the ciphertext to resemble the encryption of a different plaintext, thus preserving the confidentiality of the original plaintext. Deniable encryption aims to deceive adversaries who lack knowledge of the "original" encryption key and can only obtain "fake" keys. Our security definition considers an adversary with access to the genuine key material. Secret sharing schemes enable a dealer to distribute a secret among multiple shareholders, allowing only authorized subsets to reconstruct the secret. In a threshold secret sharing scheme, the dealer sets a threshold value, and subsets of shareholders equal to or exceeding this threshold can reconstruct the secret. While secret sharing ensures security against unauthorized subsets of shareholders, it incurs high computation and storage costs, rendering it impractical for sharing large files.

5. LITERATURE SURVEY

1. This study introduces a novel secure computation approach based on a client-server model, where a group of servers executes computations using inputs from multiple clients. The approach leverages the (k, n) threshold secret sharing technique, dividing an input s is divided into n shares that can be reconstructed from a subset of k. However, conventional secure computation using (k, n) threshold secret sharing generally requires the condition $n \geq 2k - 1$ and communication among multiple servers for each multiplication. To our knowledge, prior research has not fully addressed this challenge. Our study demonstrates a solution where communication-intensive processes are concentrated in the preprocessing phase, enabling secure computation without communication during the main computation phase, even during multiplication operations. Furthermore, we illustrate that the number of communications is independent of the number of multiplications, unlike conventional methods. As communication often consumes more processing time than secure computation itself, our approach facilitates faster overall processing compared to conventional methods. Additionally, we conduct a comprehensive security analysis and experimental simulations, demonstrating that our proposed method achieves information-theoretic security against semi-honest adversaries under specific conditions with $n < 2k - 1$.
   Year: 2023[2]

2. Cloud storage has emerged as a dominant industry for remote data management services, but it also brings security concerns, making encryption the best available approach to prevent data disclosure. Public key encryption with keyword search (PKSE) stands out as a promising technique, allowing clients to search over encrypted data files. In this method, a client generates a search token to query data files, which the cloud server uses to execute the query over encrypted data files. However, a significant security vulnerability arises when PKSE is implemented in the cloud. Specifically, the cloud server can exploit search tokens received to learn information about newly added encrypted data files containing previously queried keywords, compromising privacy. To mitigate this threat, we propose a forward secure public key searchable encryption scheme. In our

scheme, the cloud server cannot glean any information about newly added encrypted data files containing previously queried keywords. To elucidate the design rationale, we introduce a framework for constructing forward secure public key searchable encryption schemes based on attribute-based searchable encryption. Finally, experimental results demonstrate the efficiency of our proposed scheme.
Year 2022 [3]

3. Searchable Encryption (SE) is a crucial technique for ensuring both data security and usability in cloud environments. By utilizing Ciphertext-Policy Attribute-Based Encryption (CP-ABE), the Ciphertext-Policy Attribute-Based Keyword Search (CP-ABKS) scheme enables keyword-based retrieval and fine-grained access control simultaneously. However, existing CP-ABKS schemes suffer from drawbacks such as costly user certificate verification and secret key distribution, as well as a performance bottleneck caused by a single attribute authority in distributed cloud systems. To address these limitations and reduce the computational and storage burden on resource-limited cloud devices, we propose a secure Multi-authority CP-ABKS (MABKS) system in this paper. Furthermore, we extend the MABKS system to support malicious attribute authority tracing and attribute updates. Our rigorous security analysis demonstrates the selective security of the MABKS system in both selective-matrix and selective-attribute models. Additionally, our experimental results using real-world datasets highlight the efficiency and practical utility of the MABKS system in various applications.
Year: 2021 [4]

6. ARCHITECTURE DIAGRAM

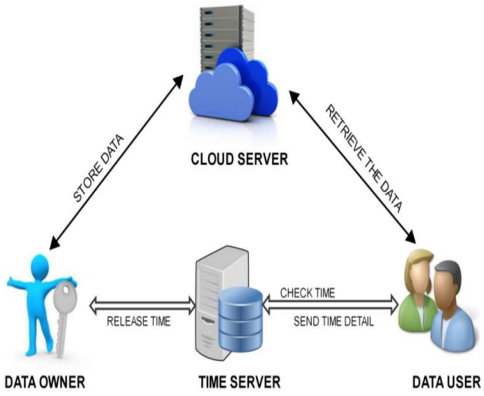

Fig -1: Architecture Diagram

7. ALGORITHMS USED

7.1 AONT Algorithm:

AONT, or All-or-Nothing Transform, is a cryptographic algorithm used to provide data confidentiality and integrity protection. It operates by dividing a plaintext message into blocks and applying a secret key to each block individually. The key characteristic of AONT is that decryption of any portion of the ciphertext requires knowledge of the entire key, making it an all-or-nothing proposition for attackers.

7.2 AES Algorithm

AES, which stands for Advanced Encryption Standard, is a symmetric encryption algorithm. This cryptographic technique was created by Belgian cryptographers Joan Daemen and Vincent Rijmen. The design of AES aimed for efficiency in both hardware and software implementations. AES supports a block length of 128 bits and offers key lengths of 128, 192, and 256 bits, making it versatile for various security requirements.

8. MODULE DESCRIPTION

1. User Module
2. Owner Module
3. Admin Module
4. Time Server Module
5. Cloud Server Module
6. destruction module

8.1 User Module:

Initially Data users has to register themselves on the website. When a data user wishes to access a file, they send a file request to the data owner. The data owner then uploads the requested file for that specific data user. Security keys are generated for each user based on their email and mobile number, ensuring uniqueness for each user. the data user only has access to the within the designated time interval allocated by the owner.

8.2 Owner module

The data owner has the capability to share files containing sensitive information with friends. These shared files are then outsourced to cloud servers for storage. The data owner generates a three-attribute key for each data user, which serves as a security measure. Additionally, the data owner allocates specific time intervals during which the data user can access the files. During these designated time

intervals, the data owner decrypts the files and uploads them to the cloud server.

8.3 Admin module

The administrator oversees the entire workflow, monitoring activities from both the file owner and file users. Additionally, all private keys and attribute keys are generated by both the file owner and the administrator to ensure the security of the files. The administrator is responsible for sending the key generation for file uploading to the data owner, and for generating the key required for file download by the data user. This key is crucial to prevent unauthorized users from accessing the file stored on the cloud server without the appropriate authorization.

8.4 Time Server module

The Time Server Module is a critical component of this project. In this module, the file owner assigns a time interval during which the file will be available for download. When the file owner uploads the file to the cloud server, they specify the duration of the file download interval. This ensures that other users cannot access the file from the cloud server outside of this specified time frame. If the current system time exceeds the allocated ending time, the file will be automatically deleted from the cloud server.

8.5 Cloud Server module

The Cloud Server Module is a crucial component of our project. It serves as the repository for all user and file owner registrations, as well as the storage location for file details uploaded by the file owner. Within this module, databases for data owners, users, and administrators are maintained. Additionally, file uploads and retrievals are exclusively conducted through the cloud server. This module is integral to our project, ensuring that unauthorized users are unable to access files without proper authentication on the cloud server.

8.6 destruction Scheme module

In our system, we utilize the generation of attributes derived from user information. For example, attributes are generated using data such as email addresses and mobile numbers. These attributes serve as the basis for generating security keys. This key generation process is initiated by the file owner at the time of file upload to the cloud server. By leveraging these attributes, we enhance the security measures for files stored in the cloud, thereby bolstering overall data protection.

9. Requirements

Table -1: Software Requirements

Operating System	Windows 7/8/10
Technology	Java
Web Technologies	Html, JavaScript, CSS
IDE	NetBeans 7.3.1
Web Server	Tomcat
Database	My SQL

Table -2: Hardware Requirements

Processor	Intel
Speed	1.1 GHz
RAM	GB
Hard Disk	260 GB

10. Activity Diagram

10.1 Data Flow Diagram

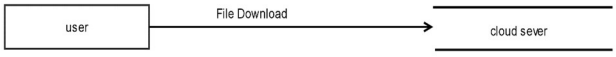

Fig -2: Level-0

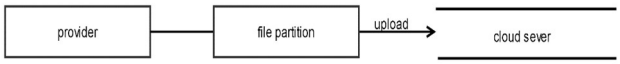

Fig -2: Level-1

11. FUTURE ENHANCEMENTS

While existing research provides valuable insights, several challenges remain, including balancing security and usability, ensuring compatibility with existing cloud

infrastructures, and addressing regulatory compliance requirements. Future research directions may include exploring novel encryption schemes, enhancing key management protocols, and developing automated detection and response mechanisms for key exposure incidents.

12. CONCLUSION

Securing cloud data under key exposure is a complex and multifaceted challenge that requires a holistic approach encompassing key management, encryption techniques, and access controls. By understanding the implications of key exposure and leveraging advanced security mechanisms, organizations can better protect their data and mitigate the risks associated with cloud computing environments. Continued research and innovation in this area are essential to stay ahead of evolving threats and ensure the confidentiality and integrity of cloud-based data.

REFERENCES

[1] G.Sucharitha , Vedula Sitharamulu , Sachi Nandan Mohanty ,Anjanna Matta , Deepa Jose. Enhancing Secure Communication in the Cloud Through Blockchain Assisted-CP-DABE, Volume 11, 2023.

[2] Keiichi Iwamura1, Ahmad Akmal Aminuddin Mohd Kamal, Communication-Efficient Secure Computation of Encrypted Inputs Using (k, n) Threshold Secret Sharing. Volume 11, 2023.

[3] Ming Zeng, Haifeng Qian, Jie Chen, and Kai Zhang, Forward Secure Public Key Encryption with Keyword Search for Outsourced Cloud Storage, Volume: 10, Issue: 1, 01 Jan.-March 2022.

[4] Yinbin Miao, Robert H. Deng, Fellow, Ximeng Liu, Kim-Kwang Raymond Choo, Senior Member, Hongjun Wu, and Hongwei Li, Multi-authority Attribute-Based Keyword Search over Encrypted Cloud Data, Volume: 18, Issue: 4, 01 July-Aug. 2021.

[5] M A Manazir Ahsan, Student Member, Ihsan Ali, Student Member, Muhammad Imran,Mohd Yamani Idna Bin Idris, Suleman Khan, Anwar Khan, A Fog-centric Secure Cloud Storage Scheme, Volume: 7, Issue:2,01April-June2022.

[6] J.Gao, H. Yu, X. Zhu, and X. Li, ''Blockchain-based digital rights management scheme via multiauthority ciphertext-policy attribute-based encryption and proxy re-encryption,'' IEEE Syst. J., vol. 15, no. 4, pp. 5233–5244, Dec. 2021.

[7] C . Li, M. Dong, J. Li, G. Xu, X.-B. Chen, W. Liu, and K. Ota, ''Efficient medical big data management with keyword-searchable encryption in health chain,'' IEEE Syst. J., vol. 16, no. 4, pp. 5521–5532, Dec. 2022.

[8] Y . He, H. Wang, Y. Li, K. Huang, V. C. M. Leung, F. R. Yu, and Z. Ming, ''An efficient ciphertext-policy attribute-based encryption scheme supporting collaborative decryption with blockchain,'' IEEE Internet Things J., vol. 9, no. 4, pp. 2722–2733, Feb. 2022.

[9] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," *2007 IEEE Symposium on Security and Privacy (SP '07)*, Berkeley, CA, USA, 2007, pp. 321-334, doi:10.1109/SP.2007.11.

[10] D. Kim and K. S. Kim, "Privacy-Preserving Public Auditing for Shared Cloud Data With Secure Group Management," in *IEEE Access*, vol. 10, pp. 44212-44223, 2022, doi:10.1109/ACCESS.2022.3169793

[11] X. Liu, Y. Zhang, B. Wang and J. Yan, "Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1182-1191, June 2013, doi:10.1109/TPDS.2012.331.

[12] G. O. Karame, C. Soriente, K. Lichota and S. Capkun, "Securing Cloud Data Under Key Exposure," in *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 838-849,1 July-Sept. 2019, doi:10.1109/TCC.2017.2670559.

[13] The Security of All-or-Nothing Encryption: Protecting against Exhaustive Key Search, August 2000, DOI: 10.1007/3-540-44598-6 23

[14] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "DepSky: Dependable and Secure Storage in a Cloud of clouds," in Sixth Conference on Computer Systems [EuroSys], 2011

[15] H. Yao, "Data Storage Security System based on Cloud Computing," *2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI)*, Changchun, China, 2022, pp. 1220-1223, doi:10.1109/ICETCI55101.2022.9832390