iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VIII **Month of publication:** August 2022

DOI: https://doi.org/10.22214/ijraset.2022.46217

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VIII August 2022- Available at www.ijraset.com

Securing Cloud Data Under Key Exposure

Mr. K. Jummelal[1], K. Kavya[2], T. Nikitha[3], P. Rupa Thapaswika[4]
[1]*Assistant Professor,* [2, 3, 4]*Undergraduate Student, Department of Computer Science Engineering, Sridevi Women's Engineering College, Hyderabad, Telangana*

Abstract: ***Recent news reveal a powerful attacker which breaks data confidentiality by acquiring cryptographic keys, by means of coercion or backdoors in cryptographic software. Once the encryption key is exposed, the only viable measure to preserve data confidentiality is to limit the attacker's access to the ciphertext. This may be achieved, for example, by spreading ciphertext blocks across servers in multiple administrative domains—thus assuming that the adversary cannot compromise all of them. Nevertheless, if data is encrypted with existing schemes, an adversary equipped with the encryption key, can still compromise a single server and decrypt the ciphertext blocks stored therein. In this paper, we study data confidentiality against an adversary which knows the encryption key and has access to a large fraction of the ciphertext blocks[3]. To this end, we propose Bastion, a novel and efficient scheme that guarantees data confidentiality even if the encryption key is leaked and the adversary has access to almost all ciphertext blocks. We analyze the security of Bastion, and we evaluate its performance by means of a prototype implementation. We also discuss practical insights with respect to the integration of Bastion in commercial dispersed storage systems.***

I. INTRODUCTION

A. Purpose

The world recently witnessed a massive surveillance program aimed at breaking user's privacy. Perpetrators were not hindered by the various security measures deployed within the targeted services. For instance, although these services relied on encryption mechanisms to guarantee data confidentiality, the necessary keying material was acquired by means of backdoors, bribe, or coercion. If the encryption key is exposed, the only viable means to guarantee confidentiality is to limit the adversary's access to the ciphertext, e.g., by spreading it across multiple administrative domains, in the hope that the adversary cannot compromise all of them. However, even if the data is encrypted and dispersed across different administrative domains, an adversary equipped with the appropriate keying material can compromise a server in one domain and decrypt ciphertext blocks stored therein. In this paper, we study data confidentiality against an adversary which knows the encryption key and has access to a large fraction of the ciphertext blocks. The adversary can acquire the key either by exploiting flaws or backdoors in the key-generation software, or by compromising the devices that store the keys (e.g., at the user-side or in the cloud). As far as we are aware, this adversary invalidates the security of most cryptographic solutions, including those that protect encryption keys by means of secret-sharing (since these keys can be leaked as soon as they are generated). To counter such an adversary, we propose Bastion, a novel and efficient scheme which ensures that plaintext data cannot be recovered as long as the adversary has access to at most all but two ciphertext blocks, even when the encryption key is exposed. Bastion achieves this by combining the use of standard encryption functions with an efficient linear transform. In this sense, Bastion shares similarities with the notion of all-or-nothing transform. An AONT is not an encryption by itself, but can be used as a pre-processing step before encrypting the data with a block cipher. This encryption paradigm called AON encryption was mainly intended to slow down brute-force attacks on the encryption key. However, AON encryption can also preserve data confidentiality in case the encryption key is exposed, as long as the adversary has access to at most all but one ciphertext blocks.

Existing AON encryption schemes, however, require at least two rounds of block cipher encryptions on the data: one preprocessing round to create the AONT, followed by another round for the actual encryption. Notice that these rounds are sequential, and cannot be parallelized. This results in considerable often unacceptable overhead to encrypt and decrypt large files.

On the other hand, Bastion requires only one round of encryption which makes it well-suited to be integrated in existing dispersed storage systems.

We evaluate the performance of Bastion in comparison with a number of existing encryption schemes. Our results show that Bastion only incurs a negligible performance deterioration (less than 5%) when compared to symmetric encryption schemes, and considerably improves the performance of existing AON encryption schemes. We also discuss practical insights with respect to the possible integration of Bastion in commercial dispersed storage systems.

©IJRASET: All Rights are Reserved

B. Scope

To counter such existing adversaries we propose Bastion, a novel and efficient scheme which ensures that plaintext data cannot be recovered as long as the adversary has access to at most all but two ciphertext blocks, even when the encryption key is exposed. We analyze the security of Bastion, and we show that it prevents leakage of any plaintext block as long as the adversary has access to the encryption key and to all but two ciphertext blocks. We evaluate the performance of Bastion analytically and empirically in comparison to a number of existing encryption techniques.

C. Model Diagram/Overview

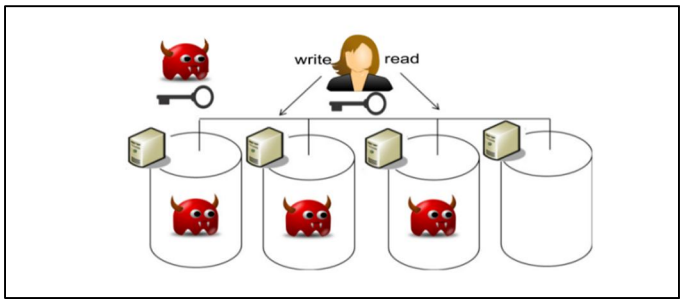


Fig. Model Diagram

Our attacker model. We assume an adversary which can acquire all the cryptographic secret material, and can compromise a large fraction (up to all but one) of the storage servers.

II. SYSTEM ANALYSIS

A. Existing System

The world recently witnessed a massive surveillance program aimed at breaking users' privacy. Perpetrators were not hindered by the various security measures deployed within the targeted services. For instance, although these services relied on encryption mechanisms to guarantee data confidentiality, the necessary keying material was acquired by means of backdoors, bribe, or coercion. If the encryption key is exposed, the only viable means to guarantee confidentiality is to limit the adversary's access to the ciphertext, e.g., by spreading it across multiple administrative domains, in the hope that the adversary cannot compromise all of them. However, even if the data is encrypted and dispersed across different administrative domains, an adversary equipped with the appropriate keying material can compromise a server in one domain and decrypt ciphertext blocks stored therein. In this paper, we study data confidentiality against an adversary which knows the encryption key and has access to a large fraction of the ciphertext blocks[7].

B. Disadvantages Of Existing System

The adversary can acquire the key either by exploiting flaws or backdoors in the key-generation software, or by compromising the devices that store the keys (e.g., at the user-side or in the cloud).

C. Problem Statement

Server model or client model are not worthy for cloud storage devices. Thus, the challenging issue in the cloud data storage is the data access control. To overcome this problem many schemes have been proposed worldwide. Among the proposed schemes the most efficient and secure way to secure the cloud data in cloud data storage systems is by using Ciphertext Policy Attribute Based Encryption. One of the major features of this scheme is that it allows the owners of the data to have complete controls over the file like providing permissions, accessing policies etc. Cryptography is using in this scheme to have access control over the cloud data. In this, the data is encrypted by using a special technique. The data is encrypted over the attributes with an access structure and a secret passcode is stamped on owner attributes. The user can only decrypt the file if the secret passcode linked with the attributes matches the passcode entered by the user. This scheme is evolved into two categories. They are single attribute authority and multiple attribute authority.

In the single attribute authority there will be only one authority and in the multiple attribute authorities there will be more than two attribute authorities.

D. Proposed System

To counter such existing adversaries we propose Bastion, a novel and efficient scheme which ensures that plaintext data cannot be recovered as long as the adversary has access to at most all but two ciphertext blocks, even when the encryption key is exposed. We analyze the security of Bastion, and we show that it prevents leakage of any plaintext block as long as the adversary has access to the encryption key and to all but two ciphertext blocks. We evaluate the performance of Bastion analytically and empirically in comparison to a number of existing encryption techniques.

E. Advantages Of Proposed System

Bastion considerably improves (by more than 50%) the performance of existing AON encryption schemes, and only incurs a negligible overhead when compared to existing semantically secure encryption modes.

III. SYSTEM REQUIREMENTS SPECIFICATIONS

A. Non Functional Requirements

1) *Economic Feasibility:* A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems.
   Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.
2) *Operational Feasibility*: Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.
3) *Technical Feasibility:* Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users.
   The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

B. Functional Requirements

1) *User Interface:* The user interface of this system is a userfriendly Java Graphical User Interface.
2) *Hardware Interfaces:* The interaction between the user and the console is achieved through Java capabilities.
3) *Software Interfaces:* The required software is JAVA1.6.
4) *Operating Environment:* Windows XP, Linux.

C. Hardware Requirements

1)	Processor	Pentium
2)	Speed	1.1 Ghz
3)	RAM	256 MB(minimum)
4)	Hard Disk	20 GB

D. Software Requirements

1)	Operating System	:	Windows Xp
2)	Programming Language	:	Java
3)	Database	:	MySQL

©IJRASET: All Rights are Reserved

IV. SYSTEM DESIGN

A. *System Implementation*

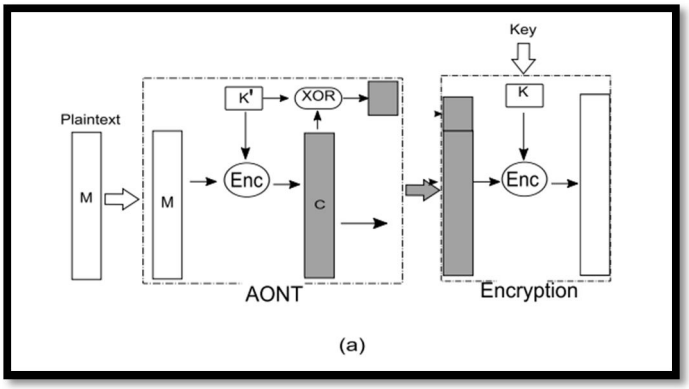


Fig. system implementation

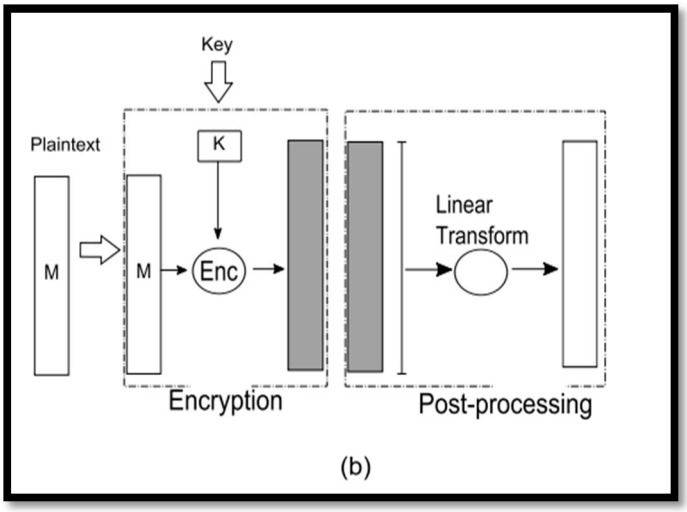


Fig. system implementation

B. *System Components*

1) *Post-Processing:* Bastion first encrypts the data with one round of block cipher encryption, and then applies an efficient linear post-processing to the ciphertext. By doing so, Bastion relaxes the notion of all-or-nothing encryption at the benefit of increased performance.
2) *Encryption:* More specifically, the first round of Bastion consists of CTR mode encryption with a randomly chosen key. The output ciphertext y′ is then fed to a linear transform.
3) *Data Owne*r: In Data Owner module, Initially Data Owner must have to register their detail and admin will approve the registration by sending signature key and private key through email. After successful login he/she have to verify their login by entering signature and private key. Then data Owner can upload files into cloud server with Polynomial key generation. He/she can view the files that are uploaded in cloud by entering the secret file key.
4) *Data User:* In Data User module, Initially Data Users must have to register their detail and admin will approve the registration by sending signature key and private key through email. After successful login he/she have to verify their login by entering signature and private key. Data Users can search all the files upload by data owners. He/she can send search request to admin then admin will send the search key. After entering the search key he/she can view the file
5) *Admin:* In Admin module, Admin can view all the Data owners and data user's details. Admin will approve the users and send the signature key and private key to the data owners and data users. Also admin will send the search request key to the users. Admin can able see the files in cloud uploaded by the data owners.

V. CONCLUSION

In this paper, we addressed the problem of securing data outsourced to the cloud against an adversary which has access to the encryption key. For that purpose, we introduced a novel security definition that captures data confidentiality against the new adversary. We then proposed Bastion, a scheme which ensures the confidentiality of encrypted data even when the adversary has the encryption key, and all but two ciphertext blocks. Bastion is most suitable for settings where the ciphertext blocks are stored in multi-cloud storage systems.

In these settings, the adversary would need to acquire the encryption key, and to compromise all servers, in order to recover any single block of plaintext. We analyzed the security of Bastion and evaluated its performance in realistic settings. Bastion considerably improves (by more than 50%) the performance of existing primitives which offer comparable security under key exposure, and only incurs a negligible overhead (less than 5%) when compared to existing semantically secure encryption modes (e.g., the CTR encryption mode). Finally, we showed how Bastion can be practically integrated within existing dispersed storage systems.

VI. FUTURE ENHANCEMENT

The future enhancement for this CP-ABE process is that it can be made to find the malicious files and the owners who upload them.To track a person who uploads the files.To track the details of the members who download a particular file and to see how many times a person viewed a particular file.

REFERENCES

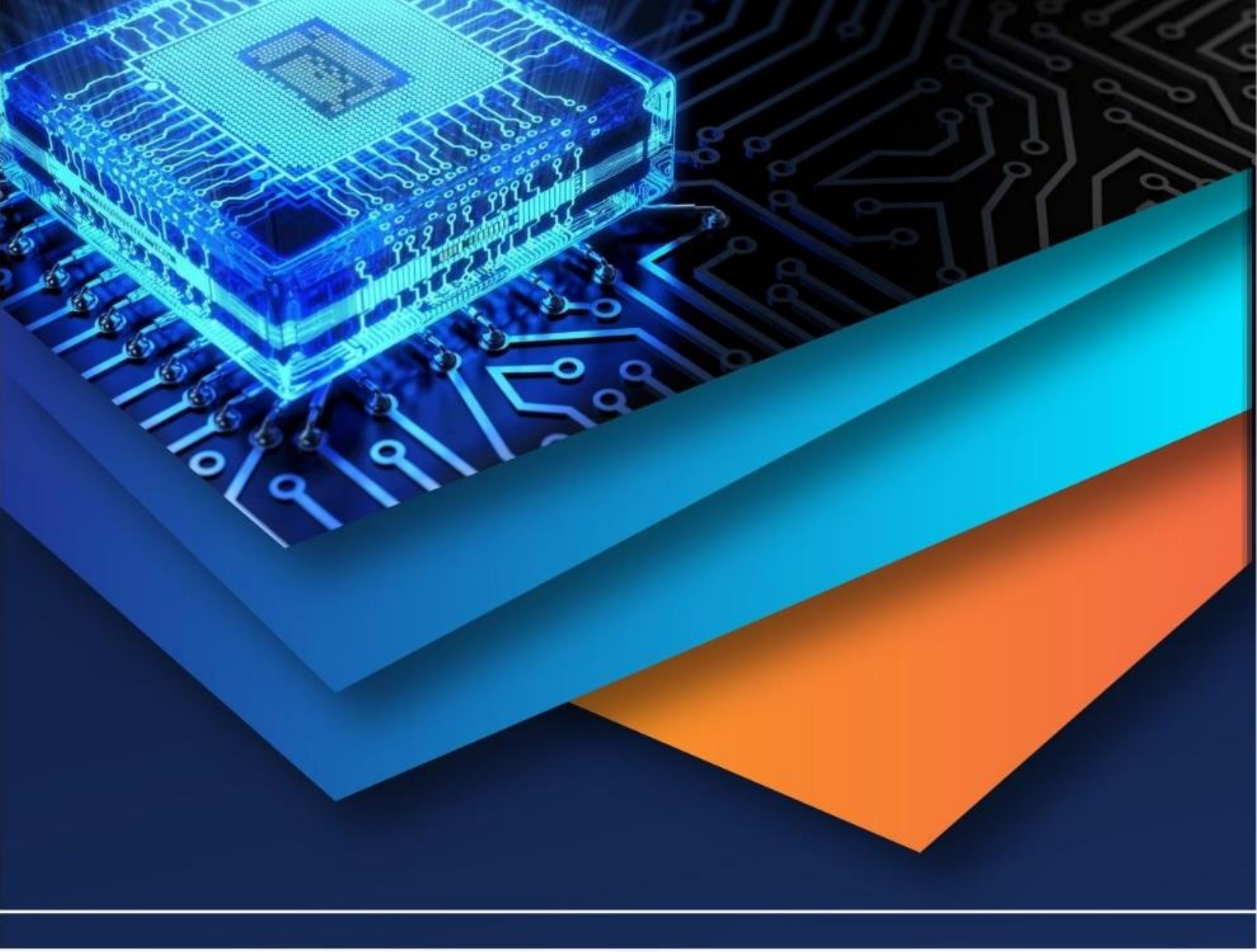
[1] M. Abd-El-Malek, G. R. Ganger, G. R. Goodson, M. K. Reiter, and J. J. Wylie, “Fault-Scalable Byzantine Fault-Tolerant Services,” in ACM Symposium on Operating Systems Principles (SOSP), 2005, pp. 59–74.
[2] M. K. Aguilera, R. Janakiraman, and L. Xu, “Using Erasure Codes Efficiently for Storage in a Distributed System,” in International Conference on Dependable Systems and Networks (DSN), 2005, pp. 336–345.
[3] W. Aiello, M. Bellare, G. D. Crescenzo, and R. Venkatesan, “Security amplification by composition: The case of doublyiterated, ideal ciphers,” in Advances in Cryptology (CRYPTO), 1998, pp. 390–407.
[4] C. Basescu, C. Cachin, I. Eyal, R. Haas, and M. Vukolic, “Robust Data Sharing with Key-value Stores,” in ACM SIGACTSIGOPS Symposium on Principles of Distributed Computing (PODC), 2011, pp. 221–222.
[5] A. Beimel, “Secret-sharing schemes: A survey,” in International Workshop on Coding and Cryptology (IWCC), 2011, pp. 11–46.
[6] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, “DepSky: Dependable and Secure Storage in a Cloud-ofclouds,” in Sixth Conference on Computer Systems (EuroSys), 2011, pp. 31–46.
[7] G. R. Blakley and C. Meadows, “Security of ramp schemes,” in Advances in Cryptology (CRYPTO), 1984, pp. 242–268.
[8] V. Boyko, “On the Security Properties of OAEP as an Allor-nothing Transform,” in Advances in Cryptology (CRYPTO), 1999, pp. 503–518.
[9] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, “Deniable Encryption,” in Proceedings of CRYPTO, 1997.
[10] Cavalry, “Encryption Engine Dongle,” http://www. cavalrystorage.com/en2010.aspx/.

TEXTBOOKS:

- Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.
- Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.

WEBSITES:

1. https://www.javapoint.com
2. https://www.geeksforgeeks.org/java/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)