A NOVEL MECHANISM FOR SECURING CLOUD DATA UNDER KEY EXPOSURE

[1]B.Manasa, [2]S.S.Shweta, [3]T.Prerna, [4]D.Ashish, [5]P.Kathik

[2]Associate Professor, [1,3,4,5]B.Tech Students
Department of Computer Science and Engineering,
St. Martin's Engineering College, Hyderabad, India

Abstract: **Nowadays technology has been improved a lot. So by using technology, an attacker can get the confidentiality data by acquiring keys. Once the encryption key is exposed, the only viable measure to preserve data confidentiality is to limit the attacker's access to the ciphertext. In order to prevent data even when the keys are exposed, we proposed an idea that the data will we segregate into parts and will be stored in the different cloud storage thinking that the attacker cannot attack all of them. Bastion is most suitable for settings where the ciphertext blocks (data) are stored in multi-cloud storage systems. Bastion gives assurance that the data remains confidential even when the key is leaked.**

IndexTerms - **Key exposure, Secrecy**

I. INTRODUCTION

Cloud computing make use of computing resources comparing of both (hardware and software) which are rendered as service over a network (typically the Internet). Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. If a cloud user approaches the services on the infrastructure layer, for instance, they can run their own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If they approach a service on the application layer, these tasks are normally taken care of by the cloud service provider. The main benefit of the cloud in the Security. In this paper, we will provide a security mechanism or a technique called as Bastion. Bastion is a technique where the data is divided into different clouds so that the attacker cannot get the whole data. When the attacker attacks the cloud he/she cannot get the full data, even he gets the data he cannot download the decrypted data as that data is in an encrypted format(unreadable format).The file will be download when the attacker will have the decrypted keys. Owner will upload the data to the cloud with k1 then the data will be uploaded into different clouds with different keys (let k2, k3). When the user wants to download the data he needs to get the keys (k2, k3, and combination keys). So there will be some burden on the cloud because cloud will provide the keys. To reduce the burden we introduce attribute authority (AA). AA is responsible for the generation of secret keys and view request. If the attributes and keys are verified then the user can download the data. For the generation of keys, we use AES algorithm.

II. LITERATURE SURVEY

There were many surveys conducted for securing cloud data under key exposure. One among them was secret sharing was taken. A secret-sharing scheme is a method by which a dealer distributes shares to parties such that only authorized subsets of parties can reconstruct the secret. A survey was done on sharing the keys secretly. The keys are shared secretly among the person. In the Existing system, there are more chances for the attackers to get the confidential data. The attacker can easily attack the cloud key and download the data. In this way, there is no security to the confidential data. So in order to provide the security here, we proposed Bastion, a scheme which ensures the confidentiality of encrypted data even when the adversary has the encryption key. Bastion is most suitable where the ciphertext blocks are stored in multi-cloud storage systems. The adversary would need to acquire the encryption key and to compromise all servers, in order to recover any single block of plaintext. In proposed system we also gave attributes. If the attributes of owner and user are matched then only the text document will be downloaded. In the attacker module also the attacker will know one key and download that part but it will not be useful to the attacker because the text will be in encrypted format.

2.1 Advantages

- Using Bastion mechanism the security is improved a lot because the data is separated into the different cloud in encrypted format by attribute authority.
- The level of performance is also increased.

III. OVERVIEW

3.1 Module Description

- Owner
- User
- Cloud

- Attribute Authority and Attacker

Owner:

In this module, we develop the Customer features functionalities. The Owner first registers his/her details and login. An Owner can outsource sensitive and valuable data to the cloud by giving access policy's then encrypting data and splitting data into multiple parts.

User:

Data User Requests to download the file by sending his access policy to attribute authority and if verification is done then decryption keys is sent to the user. Using these keys user can download data from the cloud server. To download block 1 from cloud server he should verify with the first key and to get block 2 he should verify with the second key. If both keys are verified successfully then only he can download the file.

Cloud:

In this module, we design the Cloud functionalities. The Cloud entity can view all details, file upload details and file download details. In this module, we use the DriveHQ Cloud Service API for the Cloud Integration and develop the project.

Attribute Authority:

The AA issues every user a decryption key associated with his/her set of attributes.

Attacker:

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.

3.2 Architecture

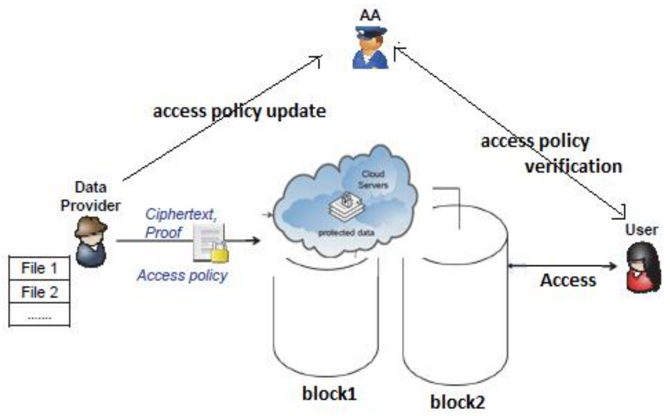


Fig. 1 Architecture of Proposed System

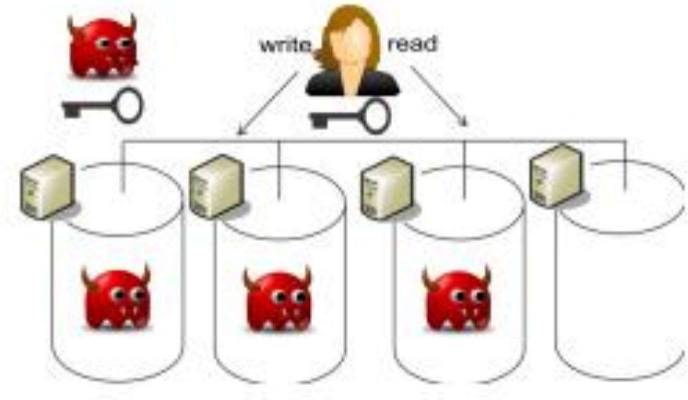


Fig. 2 Architecture of Attacker

From the above architecture we can say that the attacker cannot attack the whole data but if he attacks some part of the data it is not useful to him because the text is in the encrypted format.

IV. METHODOLOGY

Coming to the method we used in this project is AES algorithm method. The Advanced Encryption Standard, or AES, is a symmetric block cipher. Before storing the data into cloud encryption process is done that convert the readable format into an unreadable format so that only the true person can download the data with the encryption and decryption key. The first step of the ciphertext after which the cipher transformations are repeated over a number of encryption rounds. The number of rounds is determined by the key length.

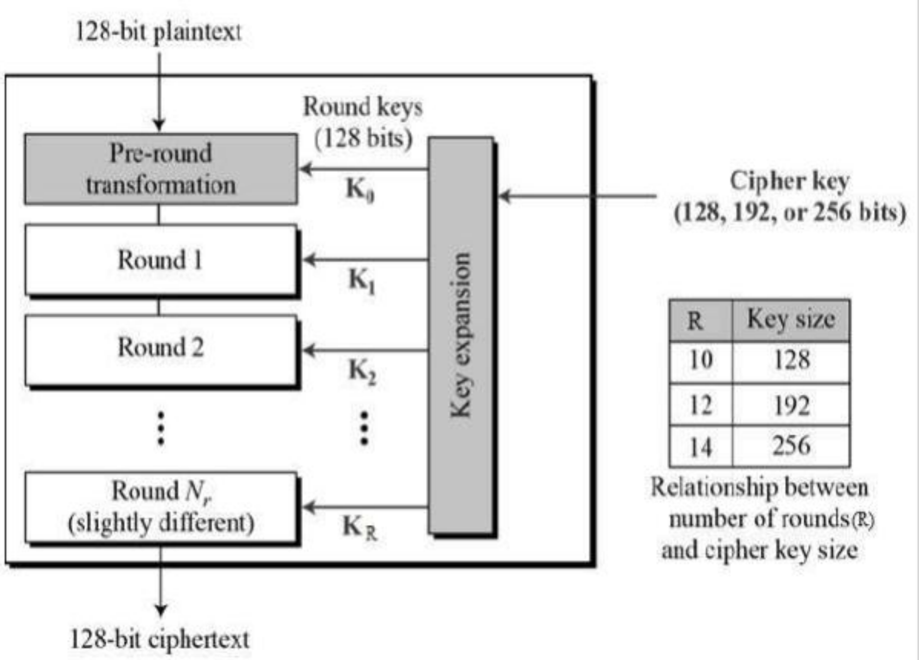


Fig. 3 AES algorithm

ENCRYPTION PROCESS

The encryption process is done in round process. Each round consists of 4 steps. The below are the 4 steps for each round.

1) Byte Substitution
2) Shift Rows
3) Mix Columns
4) Add round key

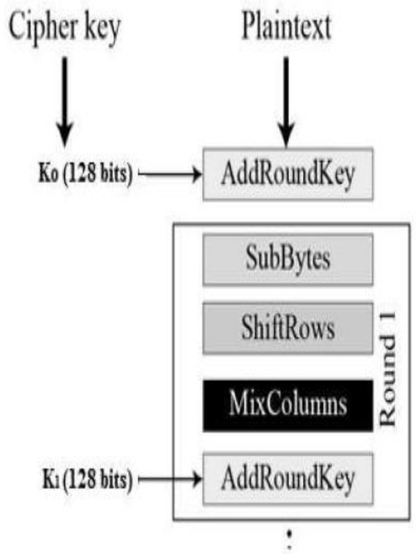


Fig. 4 Rounds in AES algorithm

Substitution (Sub Bytes):

The input is given. The result is in a matrix form

Shift rows:

The shift is carried out as follows −

- The first row will be the same.
- The second row is moved by one (byte) position to the left side.

- The third row is moved by two positions to the left side.
- The fourth row is moved by three positions to the left side.
- The output is a new matrix.

Mix Columns:

Mix column is done by performing a mathematical function. The will takes input as four bytes of one column and outputs as four bytes which replace the original column. The result is 16 new bytes.

Add round key:

A combination of 128bits of round key and 128 bits of a matrix (that is 16 bytes) is considered. If this is the final round then the output is the ciphertext. Otherwise, the 16 bytes are considered and we begin another similar round.

Decryption Process::

The decryption process is exactly similar to the encryption process but in the reverse order. There are four rounds which are performed in reversed order. The rounds are given below:

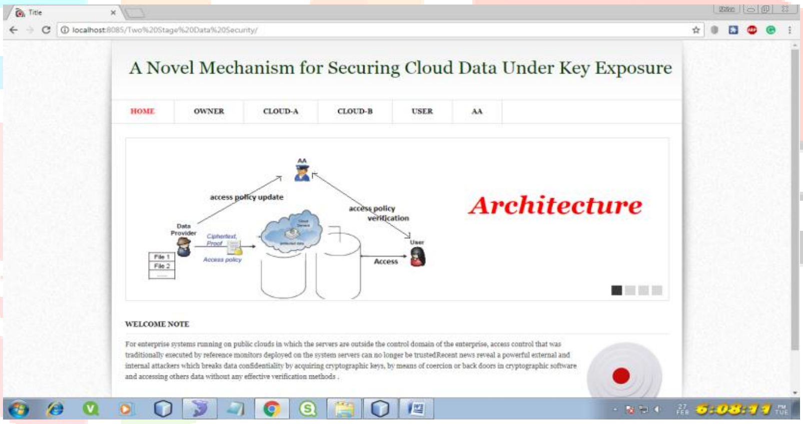
- Add round key
- Mix columns
- Shift rows
- Byte substitution

We can also use an AESBase64 encoding and decoding technique. For key generation also we can use an AESBase64 technique.

V. RESULTS

The positive result of this project is that after uploading the text document to the cloud the attribute authority will generate keys for that text document. After getting the key with that key the text document is separated into different into clouds. We will get the text document only when the attributes of the user and owner are matched. The keys are sent to the registered mail. With that keys, the user can download the full- text document. The negative result of this project is that if the attributes are not matched then the text document is not downloaded and even when the attacker tries to attack the one part of the cloud he will get the text document in the encrypted format (not in readable format).

VI. SCREENSHOTS



A Novel Mechanism for Securing Cloud Data Under Key Exposure

HOME | OWNER | CLOUD-A | CLOUD-B | USER | AA

WELCOME NOTE

For enterprise systems running on public clouds in which the servers are outside the control domain of the enterprise, access control that was traditionally executed by reference monitors deployed on the system servers can no longer be trustedRecent news reveal a powerful external and internal attackers which breaks data confidentiality by acquiring cryptographic keys, by means of coercion or back doors in cryptographic software and accessing others data without any effective verification methods .

Fig. 5 Screen shot of the proposed system

VII. CONCLUSION AND FUTURE SCOPE

We introduced Bastion, a scheme which ensures that the confidentiality of encrypted data is secured even when the attacker has the encryption key and all but two cipher-text blocks. Bastion is most suitable for settings where the ciphertext blocks are stored in multi-cloud storage systems. In the future scope, we will see a more advanced method for securing the confidential data. Many more advanced methods will be introduced.

REFERENCES

[1] 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.

[2] W. Aiello, M. Bellare, G. D. Crescenzo, and R. Venkatesan, "Security amplification by composition: The case of doubly iterated, ideal ciphers," in Advances in Cryptology (CRYPTO), 1998, pp. 390–407.

[3] 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-of clouds," 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.