Indexing Based Public-Key Encryption against Keyword Guessing Attack

Jenif D Souza W S[1], Nonitha V [2], Mary Celestina J [3]
Assistant Professor, Computer Science and Engineering, St. Joseph's College of Chennai, Chennai, India[1]
B.E., Computer Science and Engineering, St. Joseph's College of Chennai, Chennai, India[2] B.E., Computer Science and Engineering, St. Joseph's College of Chennai, Chennai, India[3]

***Abstract*— The Cloud drop box enables users to deploy data to the storage servers and save final or target data efficiently. Some of the deployed data are sensitive or vulnerable and should be prevented from any leakage. This proposed system is mainly to prevent data leakage by encrypting and indexing of keyword thereby protecting from keyword guessing attacks. In this system, the keywords stored in the index are in encrypted format so searching for a keyword is quite impossible. Generally, public key encryption method is used to avoid such problems, but it is unprotected to keyword guessing attacks (KGA). In this system a secure PEKS scheme called SEPSE against KGA is used. Here the user can encrypt the data by using two key servers, each key server provides separate or different keys to produce a common key which is used to encrypt the data. If the obtained common key is equal to the key obtained at the time of user login, it validates the user to upload and save the files which are encrypted. SEPSE can stand up to online KGA, where each keyword request made by the user is combined into a transaction public block chain, which allows key servers to learn the number of keyword requests made by the user without requiring to cohere between them for per-user rate limiting. It blocks the user when the user exceeds the specific count. Thus, SEPSE provides a stronger security guarantee compared with other existing schemes.**

***Index Terms*— Cloud storage, public-key encryption with keyword search, keyword guessing attacks, key restoration, block chain.**

I. INTRODUCTION

With cloud storage services, users can outsource their data to the remote storage server and flexibly access them via the Internet. This kind of services also provides users a structured way to send their data to other users. Typical applications include Cloud-based email systems, where multiple users (called senders) are willing to send data containing a small number of keywords to a single user (called receiver). Senders are able to outsource the data as well as keywords to the drop box server, and the receiver can store the target data from the storage server through searching. This type of searching can be done using keywords. This can free senders and the receiver from paying a lot of money for local storage, and this allows the receiver to access the target data on other devices (e.g., smartphones and personal computers) some other time. While users (henceforth, the senders and receiver are together referred to as "users") enjoy great benefits from the cloud storage services, critical security concerns in data deploying have been raised.

The important security issue is data confidentiality. From users understanding, contents of

farmed out data are very sensitive, and should not be leaked because of privacy concerns. Thus, senders always encrypt the data before outsourcing. This can be achieved by using conventional encryption, but it makes efficient searches over cipher texts by keyword impossible. PEKS is a decoding technique that relates to the above problem. In PEKS, both the data and the corresponding keywords are encrypted under the receiver's public key, and the cipher texts are outsourced to the storage server (i.e., cloud server). The receiver can generate a trapdoor on a keyword by using her/his secret key and the storage server can testwhether the cipher text of keyword matches the trapdoor for data retrieval. However, PEKS is subject to an inherent security limitation: vulnerability against offline keyword guessing attacks (KGA).

Clearly, when a trapdoor is given, an opposer encrypts all keyword candidates by using the receiver's public key and identifies the ciphertext which matches the specific targeted trapdoor; this enables the foe to recover the keyword hidden in the trapdoor to disrupt the users' privacy. Such attacks are based on the study that keywords are repeatedly selected from a small space and the end user usually use familiar keywords for searching the files .This system stress that openness of PEKS against off-line KGA is a major barrier towards the broad adoption of PEKS, since searched data (e.g., emails with important keywords) is measured as being highly secret by many individuals and organizations.

II. RELATED WORKS

The searchable encryption technique plays an important role in data outsourcing systems, such as cloud storage, where a user can submit a search query to a storage server (e.g., cloud server), and the server is able to respond with the corresponding data without learning anything more about the data content than the search result. The problem of searching on encrypted data is first introduced in [1] in which the first scheme for searches on encrypted data was proposed. However, the scheme in requires the server to linearly scan word-by-word of each file, which makes the scheme inefficient. In research work [1] a searchable encryption scheme with enhanced security is presented. To improve the efficiency, Goh [4] presented an index-based searchable encryption scheme.

The security model of searchable encryption was first defined by Curtmola et al [3] which requires a secure searchable encryption scheme to leak no more than the search pattern (i.e., whether a search query is repeated) and the access pattern (i.e., pointers to cipher texts that satisfy search query). Subsequently, many searchable encryption schemes were proposed with different features. These schemes are based on symmetric cryptosystems and are typically applied in the scenario that a user outsources her/his data to the storage server, and later she/he searches the data by keywords.

Recent works [5] have shown great possibility for making the cloud storage services effective with respect to security and privacy from blockchains. Due to the public confirmability of public blockchains, all transactions created by a user on a blockchain can be tracked and cannot be modified and copied/forged. Hence, a public log can be made by using public blockchains to keep track of what happens to the deployed data [2]. The log is intrinsically resistant to modification or forgery, which can be utilized to thwart online KGA. To protect PEKS against off-line KGA without the single point-of-failure problem, system suggest SEPSE, where multiple key servers are selected to jointly help users in producing keywords without learning any information about the keywords. SEPSE supports key restoration or key renewal on the key servers to abide the key compromise on key servers. Additionally, SEPSE resists online KGA by a blockchain-assisted rate-

limiting mechanism, where the number of servers derived keyword requests for each user is documented to the blockchain, and the key servers stop responding after a limit is reached. As a result, SEPSE achieves a stronger security guarantee than other existing schemes [2].

Public key encryption with keyword search (PEKS) is a cryptographic fundamental that talks about the privacy issues in user data. Here both the documents and keywords are encrypted with receivers' public key and decrypted using users' private key. If the searched keyword matches with encrypted keywords stored in the index the user can retrieve the data and further processing will takes place. Since searched data (e.g. sensitive words in email) is kept highly confidential by many individuals and organizations. Also, the keywords are of low entropy so they are more vulnerable to keyword guessing attack [6].

III. PROPOSED SYSTEM

In this proposed system we present a secure and efficient Public key encryption with keyword search (PEKS) scheme called SEPSE to stand against both online and offline keyword guessing attacks (KGA) for secure cloud drop box. In SEPSE it stand up to off-line KGA without the single-point-of-failure problem, where multiple key servers are engaged to assist users in encrypting keywords without learning any information about the keywords. SEPSE supports key renewal on key servers so as to fight against the key compromise and provides a stronger privacy for the data. This system present a block chain assisted rate-limiting mechanism and integrate it into SEPSE to stand up to online KGA, where each request of servers-derived keyword made by the user is combined with a transaction on a public block chain.

This system is implemented by using two web servers such as Tomcat 6.20 and Tomcat 7.0.11 in a windows system. This proposed system comprises of four modules such as User registration and login with key generation, File upload and keyword extraction, Keyword search and file retrieval and Block chain assisted rate-limiting mechanism. The proposed system architecture is shown in FIGURE 1.

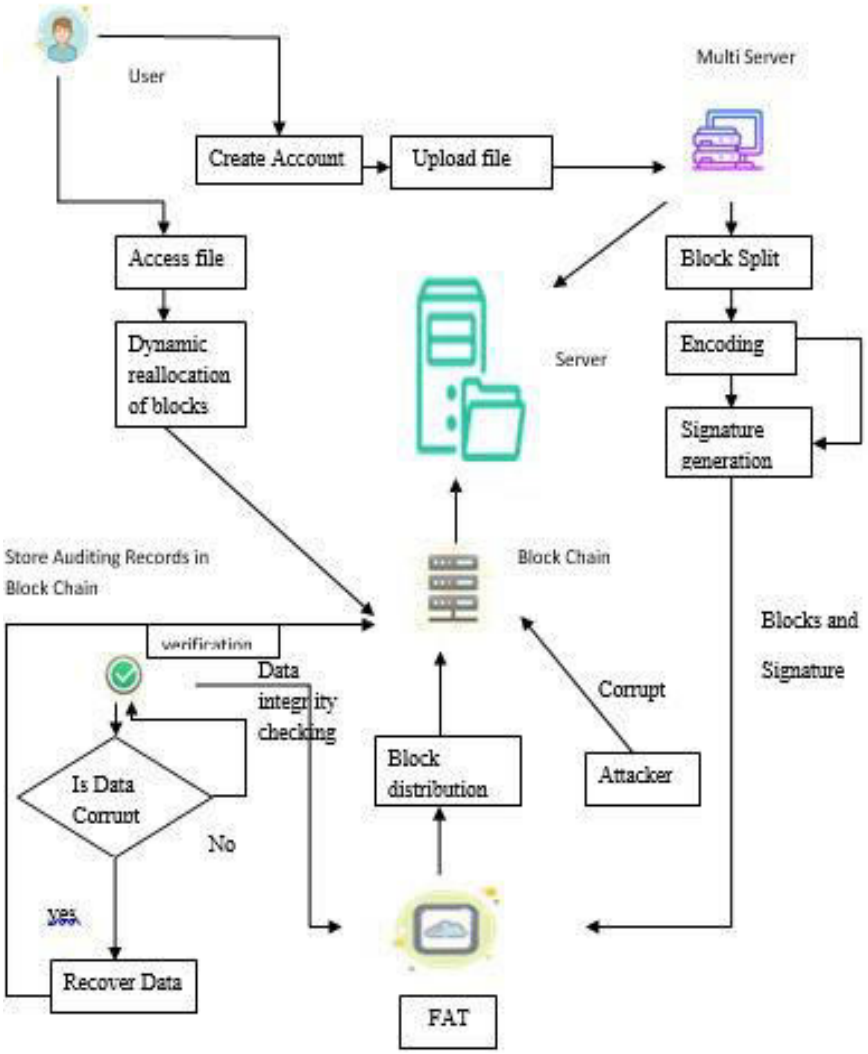

Figure 1. Proposed system Architecture

The working of the proposed system is shown in the below algorithm.

Algorithm:

Step 1: User has to create an account and a key produced at the time of user login.

Step 2: The two key servers communicates with each other to produce hash keys on its servers and later combines it into a common key.

Step 3: If the key generated during user login and the common key matches the only the files can be uploaded in cloud.

Step 4: Using the receivers' public key the user encrypt the file and upload it.

Step 5: Indexing is done for the uploaded file.

Step 6: If the data is corrupted retrieval is done only if the encrypted keyword matches the keywords in the index.

Step 7: All the transactions were updated in the blockchain. IMPLEMENTATION

The entire proposed system is divided into four different modules such as User registration and login with key generation, File upload and keyword extraction, Keyword search and file retrieval and Block chain assisted rate-limiting mechanism.

[1] User registration and login with key generation

In this module user creates an account and register it. The two keys such as common key and user key generation is done. One key is produced at the time of user login. Whereas another key is generated by the key servers. Here, the two key servers S1 and S2 starts communicate with each other to produce hash keys on each key servers and combines to form a common key. If the two keys such as common key and user login key match each other it authenticates the user to upload and download the files by data owner and receiver.

[2] File upload and Keyword Extraction

The file upload and keyword extraction is done in this module. Before uploading the file, it should get encrypted. The data owner encrypts the file by using the receivers' public key and then uploads the files to cloud as in Figure 2. This process will happen only if two keys are matched. After the file is successfully uploaded, the keyword extraction and indexing is done. Thereby, performs indexing by extracting all the keywords in an uploaded file and stored in an index.

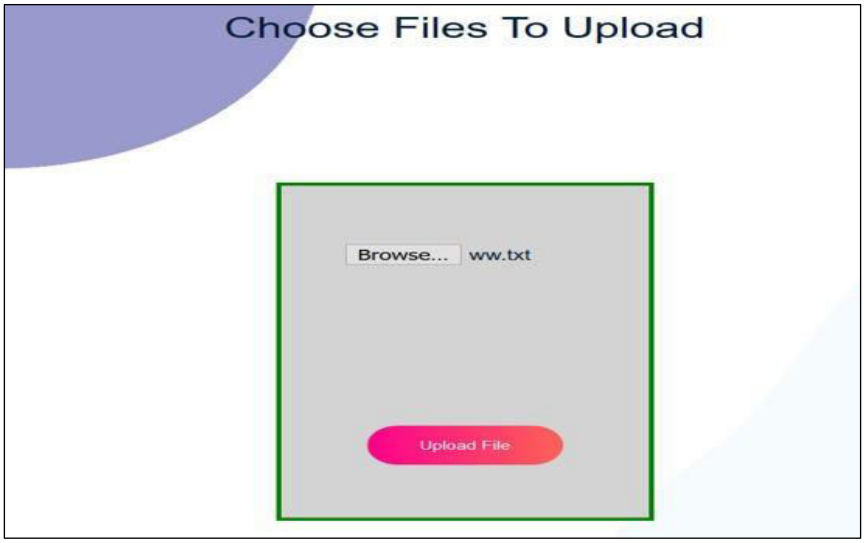

Figure 2. File upload

[3] Keyword Search and File Retrieval

In this module, the keyword searching and retrieval of file takes place. The receiver retrieves the file by searching the keywords which is taken as encrypted format as shown in Figure 3. If the encrypted keyword matches the keywords in the index, the file is then decrypted and can be viewed by a receiver.

Figure 3. File retrieval

[4] Block chain assisted rate-limiting mechanism

In this module, updating of all the transaction is done. The transactions such as user id, threshold values, status and time get updated into the block chain. Thus each transaction updated simultaneously to maintain high security. The Figure 4 shows the user detail which is uploaded in blockchain. Such type of mechanism is known as Blockchain assisted rate-limited mechanism.

Figure 4. Rate limiting

The main advantage of this system is that, it can monitors and alerts the data representatives if any chance of insider threats or intruders. This system keeps a track of attackers' details. Using this system the attackers or intruder's details and status can be viewed as explained in Figure 5.

Figure 5. Attacker's details and status

Hardware Interface

For implementing this project we require the following hardware requirements.

- Hard disk : 500 GB and above.
- Processor : i3 and above.
- RAM : 4GB and above.

Software Interface

A set of instructions or programs required to make a hardware platform suitable for the desired task is known as software. The software also can be defined because of the utility programs that are required to drive hardware of the pc.

- Operating System: Windows 7 and above (64-bit).
- Java version : JDK 1.7
- Web Server :Tomcat 6.20 and Tomcat 7.0.11

IV.CONCLUSION

This system overcomes the keyword guessing Attack by generating separate key servers. This system also monitors and alerts the data representatives if any chance of insider threats or intruders. This system also shows the attacker or intruders details and status. Each key server can periodically renew its secret share to resist the key compromise. The security is improved in this way. This system also resists KGA without the single point of failure. This proposed system obtains 97% accuracy when compared to other existing system.

Future enhancement

For the future work, SEPSE can efficiently resist online KGA without requiring a synchronization between them for per-user rate limiting. It blocks the user when the user exceeds the specific count with a threshold value. Thus, SEPSE provides a stronger security guarantee compared with existing schemes. Study for the potentials for enhancing the security, efficiency, and functionality of data outsourcing systems will be achieved.

REFERENCES

[1] Y. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. Of ACNS, vol. 5, 2005, pp. 442–455.

[2] Y. Zhang, X. Lin, and C. Xu, "Blockchain-based secure data provenance for cloud storage," in Proc. ICICS, 2018, pp. 3–19.

[3] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions in Proc. of ACM CCS, 2006, pp. 79–88.

[4] Goh, "Secure indexes," Cryptology ePrint Archive, report 2003/216, 2003.

[5] Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy- preserving smart contacts", 2015.

[6] X. Lin , X. Shen , C. Xu ,Y. Zhang, "Block chain-based public integrity verification for cloud storage against procrastinating auditors," IEEE Transactions on Cloud Computing, TCC.2019.2908400,2019.