International Journal of Scientific Research & Engineering Trends
Volume 7, Issue 5, Sept- Oct-2021, ISSN (Online): 2395-566X

Attribute-Based Temporary Keyword Search Scheme in Cloud Storage Server

M. Tech. Scholar Sindhu Mathuku, Asst. Prof. V Dakshayani, Asst. Prof. V Subhasini
Department of Computer Science & Engineering,
MJR College of Engineering & Technology, Piler, AP, India.

***Abstract*- Attribute-based keyword search (ABKS), as an important type of searchable encryption, has been widely utilized for secure cloud storage. In a key-policy attribute-based temporary keyword search (KP-ABTKS) scheme, a private key is associated with an access policy that controls the search ability of the user, while a search token is associated with a time interval that controls the search time of the cloud server. However, after a careful study, we uncover that the only existing KP-ABTKS construction [1] is not secure. Through two carefully designed attacks, we first show that the cloud server can search the cipher-text in any time. As a result, their scheme cannot support temporary keyword search. To address this problem, we present an enhanced KP-ABTKS scheme and prove that it is selectively secure against chosen-keyword attack in the random oracle model. The proposed scheme achieves both fine-grained search control and temporary keyword search simultaneously. In addition, the performance evaluation indicates that our scheme is practical.**

***Keywords*- Cloud computing, fine-grained search control, searchable encryption, and temporary keyword search.**

I. INTRODUCTION

Cloud computing is an emerging Internet technique that provides massive computing and storage service for individuals and companies. As a significant application of cloud computing, cloud storage can efficiently reduce local storage costs and realize data sharing. Due to its cheapness and convenience, more and more data owners store their sensitive data in the cloud.

However, this causes huge concerns for the reveal of the sensitive data, because the data owners lose control over the local data when they outsource these data to the cloud. For example, personal health records (PHR), email data and financial documents stored in iCloud may be compromised by the attacks from the hacker and the legal pressure faced by the cloud service. One method for protecting sensitive data is to encrypt data and upload cipher-text to the cloud. However, traditional encryption schemes make data retrieval impossible by the user.

To address this issue, Boneh et al. [2] introduced the concept of public key encryption with keyword search (PEKS), which can achieve data confidential and searchable simultaneously. In a PEKS scheme, the data owners encrypt the keyword of each file and store the cipher-texts in the cloud; the cloud server who has a search token associated with a keyword can retrieve data by testing whether the search token and the cipher-text correspond to the same keyword.Although PEKS can achieve data confidential and searchable simultaneously, it cannot support fine-grained access control for the user's retrieval permission. To address this problem, Zheng et al. [3] and Sun et al. [4] proposed the primitive of attribute-based keyword search (ABKS) based on PEKS and attributed-based encryption (ABE) [5].

Although PEKS can achieve data confidential and searchable simultaneously, it cannot support fine-grained access control for the user's retrieval permission. To address this problem, Zheng et al. [3] and Sun et al. [4] proposed the primitive of attribute-based keyword search (ABKS) based on PEKS and attributed-based encryption (ABE) [5].

However, once the cloud server has the search token in ABKS schemes, it can search the past and future cipher-texts, which may cause privacy leakage. To improve the security of ABKS, Ameri et al. [1] presented the primitive of key-policy attribute-based temporary keyword search (KPABTKS) in which the search token can only be used in a time interval rather than any time.

In a KP-ABTKS scheme, a search token is labeled with an access policy and a time interval, while a cipher-text is labeled with attributes and an encrypting time; the search token will only allow the cloud server to retrieve the cipher-text when the attributes satisfy the access policy and the time interval contains the encrypting time. Ameri et al. [1] proposed the first construction of KP-ABTKS and claimed their construction is selectively secure against chosen-keyword attack. However, after carefully studying, we find the ADMS construction is not secure, and existing ABKS schemes cannot achieve fine-grained search control and temporary keyword search simultaneously.

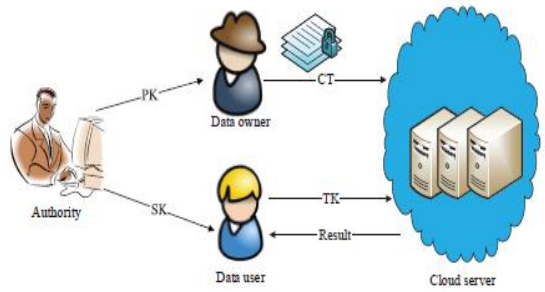

Fig 1. Model of KP-ABTKS.

II. EXISTING SYSTEM

After a careful study, we uncover that the only existing KP-ABTKS construction is not secure. Through two carefully designed attacks, we first show that the cloud server can search the cipher text in any time. As a result, their scheme cannot support temporary keyword search.

The data owners encrypt the keyword of each file and store the cipher texts in the cloud; the cloud server who has a search token associated with a keyword can retrieve data by testing whether the search token and the cipher text correspond to the same keyword. Although PEKS can achieve data confidential and searchable simultaneously, it cannot support fine-grained access control for the user's retrieval permission.

In ABKS, a search token is associated with an access policy (resp., attributes) and a ciphertext is associated with attributes (resp., an access policy); a user can retrieve the cipher text only when the attributes satisfy the access policy. Once the cloud server has the search token in ABKS schemes, it can search the past and future cipher texts, which may cause privacy leakage. ABKS schemes cannot achieve fine-grained search control and temporary keyword search simultaneously.

1. Disadvantages:

- We find the KP-ABTKS construction is not secure, and existing ABKS schemes cannot achieve fine-grained search control and temporary keyword search simultaneously.
- Through two carefully designed attacks, we first show that the cloud server can search the cipher-text in any time. As a result, their scheme cannot support temporary keyword search.

2. Attributed-Based Encryption:

Attributed-Based Encryption is a type of public-key encryption in which the secret key of a user and the cipher-text are dependent upon attributes. In such a system, the decryption of a cipher-text is possible only if the set of attributes of the user key matches the attributes of the cipher-text. There are mainly two types of attribute-based encryption schemes: Key-policy attribute-based encryption (KP-ABE) and cipher-text-policy attribute-based encryption (CP-ABE). Sahai and Waters [5] first introduced ABE, and Goyal et al. [6] later divided it into key-policy ABE (KP-ABE) and cipher text-policy ABE (CPABE).

In KP-ABE [6], a cipher-text is associated with attributes and a secret key is associated with an access policy; a user can decrypt the cipher-text if and only if the attributes satisfy the access policy. While in CP-ABE a cipher-text is associated with an access policy and a secret key is associated with attributes. In the past 15 years, there have been extensive study and application of ABE. The fully secure schemes were presented to improve the security of ABE.

3. Public Key Encryption with Keyword Search:

Searchable encryption is a practical technique for searching over encrypted data, and can be divided into symmetric searchable encryption (SSE) and PEKS. This paper focuses on PEKS, which can provide more flexible search queries. In PEKS, the user sends his search token and outsources the search operations to the cloud; the cloud server can only test whether the search token and the cipher-text correspond to the same keyword, but without learning anything more about the keyword.

4. Attribute-Based Keyword Search:

To control the user's retrieval permission, the concept of ABKS was independently introduced by Zheng et al. [3] and Sun et al. [4]. ABKS can be divided into key-policy ABE (KP-ABKS) and cipher-text-policy ABE (CP-ABKS).

In a KP-ABKS (resp., CP-ABKS) system, a data owner encrypts a keyword with attributes (resp., an access policy) and outsources the cipher-text to the cloud; a data user sends the cloud server a search token that is associated with an access policy (resp., attributes) and a keyword; if the attributes satisfy the access policy, the cloud server can use the search token to retrieval the cipher-text.

III. PROPOSED SYSTEM

In this paper, we first discussed the existing KP-ABTKS scheme, namely ADMS. By presenting two carefully designed attacks against ADMS scheme, we showed that the cloud server can retrieve the cipher-text created in any time, which causes the ADMS scheme insecure. Then, we presented a secure KP-ABTKS scheme in the prime order groups, which achieves fine-grained search control and temporary keyword search. In our construction, the uses are allowed to search the cipher-text when their attributes satisfy the access tree, while the cloud server is only allowed to search the cipher-text in a limit time interval.
To improve the security of ABKS, Ameri et al. [1] presented the primitive of key-policy attribute-based temporary keyword search (KPABTKS) in which the

search token can only be used in a time interval rather than any time.In this work, we design two attacks against ADMS scheme and construct a secure enhanced KP-ABTKS scheme.

The main contributions are summarized as follows:

- We first point out that ADMS scheme is not secure by constructing two effective attacks. In the first attack, by modifying the old search token, the cloud server can construct a new search token that can be used in any time period.
- In the second attack, we show that anyone can change the encrypting time to anytime he wants. As a result, even though a cipher text is not created in the time interval that the search token is associated with, it can also be searched by this search token. Hence, alike with other ABKS schemes, the ADMS scheme cannot support temporary keyword search.
- We provide a construction of KP-ABTKS and prove that it is selectively secure against chosen-keyword attack in the random oracle model. In our system, a data user's secret key is corresponding to an access tree, and a data owner can control the search permission by encrypting the keyword with attributes.
- A data user can outsource the temporary keyword search operations to the cloud by creating a search token that is corresponding to an access tree and a time interval. The cloud server can search the encrypted data only when the attributes satisfy the access tree and encrypting time is included in the time interval simultaneously. To the best of our knowledge, this is the first public key encryption that achieves attribute based search control and temporary keyword search simultaneously.
- We implement our KP-ABTKS construction and compare its performance with other related works. The comparison and experimental results show that our scheme is efficient and practical.

1. Advantages:

- The proposed scheme achieves both fine-grained search control and temporary keyword search simultaneously.
- Attribute-based keyword search is used to control the user's retrieval permission; the concept of ABKS was independently.

IV. SYSTEM MODEL

As shown in Figure 1, the system model of KP-ABTKS comprises four entities: the authority, data owner, data user and cloud server. First, the authority publishes the public key PK, and sends the secret key SK to the user through a secure channel. After that, the data owner encrypts the keyword, and uploads the cipher-text CT to the cloud server. Next, the data user sends the cloud server a search token TK that allows the cloud server to search cipher-text. Finally, the cloud server returns the search result to the data user. Specifically, the role of each entity is described as follows:

1. Authority:
The authority runs the setup algorithm to generate the master key MK for itself and the public key PK for the system; for each data user, it runs the key generation algorithm to generate the user's secret key SK for an access structure.

2. Data Owner:
The data owner runs the encryption algorithm to encrypt the keyword 'ω' under the time period 't' and attributes set S, and stores the cipher-text CT in the cloud.

3. Data User:
When the data user wants to search the data files corresponding to the keyword 'ω', she/he first runs the token generation algorithm to generate the search token TK for the access structure, the keyword 'ω', and a time interval T; then he makes search query by submitting TK to the cloud server.

4. Cloud Server:
After receiving the search token TK, the cloud server first checks whether Ƭ (S) = 1 and t ε T hold or not. if data is not available or data not belongs to search token, then the cloud server terminates the searching algorithm and returns the error symbol.In our system model, we assume the authority, data owners, and authorized data users are fully trusted. The malicious uses may collude with the cloud server to search files beyond their retrieval permissions. The cloud server is honest-but-curious, which means that it honestly follows the above protocol but attempts to infer any privacy information about the keyword from the search token and cipher-text.

V. TOOLS USED

Java: Developed by Sun Microsystems, Java is one of the most used programming languages. It is High-Level language. It is Object-oriented language and is very robust in nature. Java is flexible and provides cross-platform support. It was created by James Gosling.

1. Query:
Query is a library of JavaScript. It helps in reducing scripting on client system side. We can operate on multiple platforms using JQuery. It is very small and hence very agile.

2. JavaScript:
It is a lightweight interpreted or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for web pages, many non-browser environments also use it, such as Node.js, etc. JavaScript is a prototype-based, multi-paradigm, dynamic language, supporting object-oriented, imperative and declarative styles.

3. HTML:

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications with Cascading Style Sheets (CSS) and JavaScript.

4. MySQL:
It is open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co- founder Michael Widenius's daughter and "SQL", the abbreviation for Structured Query Language.

VI. CONCLUSION

In this paper, we first discussed the existing KP-ABTKS scheme [1], namely ADMS. By presenting two carefully designed attacks against ADMS scheme, we showed that the cloud server can retrieve the cipher-text created in any time, which causes the ADMS scheme insecure. Then, we presented a secure KP-ABTKS scheme in the prime order groups, which achieves fine-grained search control and temporary keyword search.

In our construction, the uses are allowed to search the cipher-text when their attributes satisfy the access tree, while the cloud server is only allowed to search the cipher-text in a limit time interval. Finally, we gave the security and performance analysis to show that our scheme is secure and practical.

Our proposed scheme is the first public key encryption system supporting attribute-based search control and temporary keyword search simultaneously. However, our scheme is only proved secure against selective adversaries, which have to commit the challenge attributes set and time period before the setup stage. Consequently, our proposed scheme may be not secure against adaptive attacks, where the challenge attributes set and time period are chosen in the challenge stage. We leave it as our future work to construct an adaptively secure KP-ABTKS system in the standard model.

REFERENCES

[1] M. H. Ameri, M. Delavar, J. Mohajeri, and M. Salmasizadeh, "A keypolicy attribute-based temporary keyword search scheme for secure cloud storage," IEEE Transactions on Cloud Computing, pp. 1–1, 2018.
[2] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in International conference on the theory and applications of cryptographic techniques. Springer, 2004, pp. 506–522.
[3] Q. Zheng, S. Xu, and G. Ateniese, "Vabks: verifiable attribute-based keyword search over outsourced encrypted data," in IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE, 2014, pp. 522–530.
[4] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," in IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE, 2014, pp. 226–234.
[5] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2005, pp. 457–473.
[6] V. Goyal, O. Pandey, A. Sahai, and B.Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proceedings of the 13th ACM conference on Computer and communications security. Acm, 2006, pp. 89–98.
[7] J. Bethencourt, A. Sahai, and B.Waters, "Ciphertext-policy attribute-based encryption," in 2007 IEEE symposium on security and privacy (SP'07). IEEE, 2007, pp. 321–334.
[8] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in International Workshop on Public Key Cryptography. Springer, 2011, pp. 53–70.
[9] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2010, pp. 62–91.
[10] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in Annual cryptology conference. Springer, 2010, pp. 191–208.
[11] M. Chase, "Multi-authority attribute based encryption," in Theory of Cryptography Conference. Springer, 2007, pp. 515–534.

Author's Profile

Sindhu Mathuku Pursuing M.Tech at MJR College of Engineering & Technology, Department of Computer Science & Engineering, Piler, Chittoor Dist.

V Dakshayani Working As An Assistant Professor In Mjr College Of Engineering & Technology, Department Of Cse, Piler, Chittoor Dist.

V Subhasini Working As An Assistant Professor In MJR College Of Engineering & Technology, Department Of CSE, Piler, Chittoor Dist.