International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 06 | June 2019 www.irjet.net p-ISSN: 2395-0072

A Key-Policy Attribute based Temporary Keyword Search Scheme for Secure Cloud Storage

Dr. Aziz Makandar[1], Mrs. Rashmi.Somshekhar[2] and Miss. Soukhya Homey[3]

[1]Professor, Dept. of Computer Science, KSAWU, Vijayapura
[2]Research Scholar, Dept. of Computer Science, KSAWU, Vijayapura
[3]Student, Dept. of Computer Science, KSAWU, Vijayapura

Abstract - *Temporary keyword search on confidential data in a cloud environment is the main focus of this project. The cloud providers are not fully trusted. So, it is necessary to outsource data in the encrypted form. In the attribute - based keyword search schemes, the authorized users can generate some search tokens and send them to the cloud for search operation. These search tokens can be used to extract all the cipher texts which are produced at any time and contain the corresponding keyword. Since this may lead to some information leakage, it is more secure to propose a scheme in which the search tokens can only extract the cipher texts generated in a specified time interval. To this end, in this project, we introduce a new cryptographic primitive called a key-policy attribute-based temporary keyword search (KP-ABTKS) which provide this property. To evaluate this security of our scheme, we formally prove that our proposed scheme achieves the keyword secrecy property and is secure against selectively chosen keyword attack.*

Key Words: Attribute based encryption, Token, Temporary keyword search, cloud security.

1. INTRODUCTION

Cloud computing plays an important role in our daily life, because it provides efficient, reliable and scalable resources for data storage and computational activities at a very low price. However, the direct access of the cloud to the sensitive information of its users threatens their privacy. A trivial solution to address this problem is encrypting data before outsourcing it to the cloud. However, searching on the encrypted data is very difficult.

In Key-Policy Attribute Based Temporary Keyword Search (KP-ABTKS) schemes, the data owner generates a searchable cipher text related to a keyword and the time of encrypting according to an arbitrary time interval and generates a search token for intended keyword to find the cipher text. Then, he/she sends the generated token to the cloud to run the search operation. By receiving the token, the cloud looks for the documents contain the intended keyword. The search result on a cipher text is positive, if (i) the data user's attributes satisfies the access control policy, (ii) the time interval of the search token encompasses the time of encrypting, and (iii) the search token and the cipher text are related to the same keyword. To show that the proposed notion can be realized, we also propose a concrete instantiation for this new cryptographic primitive based on bilinear map.

We introduce the novel notion of KP-ABTKS, and propose a concrete construction for this new cryptographic primitive which can be applied in the cloud storage services. The proposed concrete scheme is designed based on bilinear pairing. In the proposed KP-ABTKS, each user is identified with an access control policy. The data owner selects an attribute set, and runs the encryption algorithm with regard to it. If a data user's attributes set satisfies the access tree of the data owner, then he/she can generate a valid search token. The cloud applies the generated search token to find the corresponding cipher texts which have been encrypted in a time interval specified by the data user.

We formally define two security definitions for KPABTKS in the standard model. One of them defines its security against selectively chosen keyword attack (KPABTKS-SCKA), and the other one defines the keyword secrecy of KP-ABTKS. We formally prove that our proposed scheme satisfies these security definitions under the hardness of the Decisional Diffe-Hellman (DDH) assumption.

We evaluate the performance of the proposed construction of KP-ABTKS in terms of both computational complexity and the execution time. The performance evaluation shows the practical aspects of our proposal.
Algorithms:

The project consists of five algorithms, Setup, Key Generation, Encryption, Token Generation, and Search. These algorithms are described as follows:

- (msk,pp) ← Setup(1λ):This algorithm is run by the TTP. It takes the security parameter λ as input and generates the master secret key msk and the public parameter pp.
- sk ← KeyGen(msk,Tr): This algorithm generates a secret key sk for the user with the access tree, Tr. The TTP determines the access tree Tr and runs this algorithm.
- cph ← Enc(ω,ti,Atts,pp): This algorithm generates a searchable ciphertext related to the keyword ω

and time of encrypting ti according to an attribute set, Atts which is determined by the data owner.

- st ← TokenGen(sk,ω,[ts,te]): The data user runs this algorithm to generate the search token st for searching the ciphertexts which are encrypted in the time interval [ts,te], and contain the keyword ω, according to its secret key sk.
- {0,1} := Search(cph,st): For each stored ciphertext cph and the received search token st which is associated with specific keyword ω and attribute set Atts, this algorithm returns 1.

2. Architecture

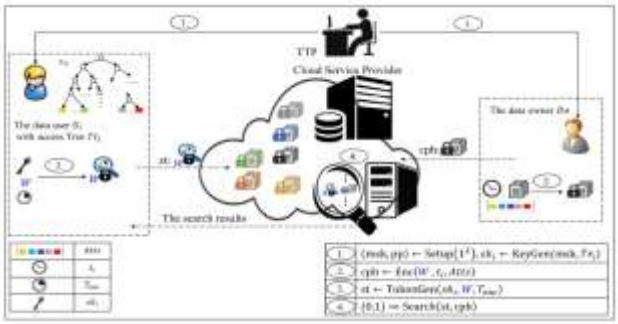


Fig -1 Architecture

Data Owner register for file upload after registration successfully the data owner will login with the help of user name and password. To upload the file, Data owner will require master key. After the registration of the data owner successfully master key goes to TTP. Data Owner's request for the response of TTP. The TTP will login with the help of username and password. After the login Data Owner's request will be displayed. TTP will analyse pending request lists after TTP will response to the Data Owner. At that time master key will upload the file into the cloud successfully.

Data User firstly register. After the completion of registration successfully, Data User will login with the help of username and password. After the registration successfully the master key request goes to the TTP. The TTP will login with the help of user name and password. After the login, Data User's request details will be displayed. TTP will analyse the pending request lists after TTP will response to the Data User. At that time master key will go to the Data User's mail. Data User with the help of master key Data User will search the files. After the activation for the required file, Data User needs a token. Token behaves like one time password. Once the token is used it will used for the another time. It acts like one time password.

After that Data User will put that particular token, the list of Data Owner files will be displayed. The files will be in the form of encrypted. The files which are in the encrypted form cannot readable that is in the non-readable form. To convert the files from non-readable to readable so, we need a keyword. A keyword is used for conversion of non-readable to readable form and it can be used for download the file. It is also used for the specific interval of time for the download of file. If the Data User needs decrypted and download the file. The Data User send a request to the cloud.

The cloud firstly login with the help of username and password. After that cloud will know about the keyword request details. Based on the particular keyword request, Cloud will response that keyword. The Data User will get the keyword and put that particular keyword to decrypt the file and download that file successfully.

If the download process will not completed with the given interval of time. It will send the alert message as "Time Over". At that time Data User cannot download again request will do for the security purpose.

2.1 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.

jQuery:

jQuery 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.

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.

HTML:

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

Eclipse:

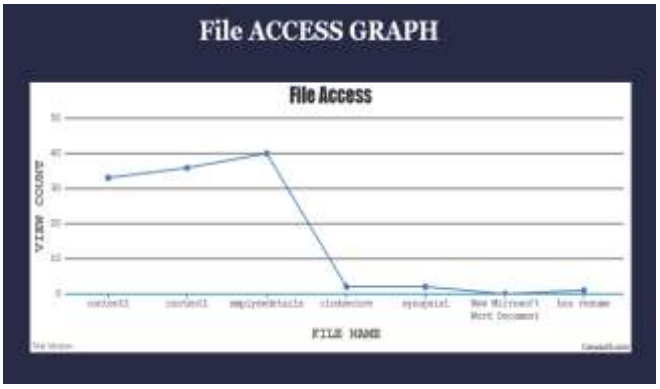
It is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly Java and its primary use is for developing Java applications in other programming languages.

MySQL:

It is open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-

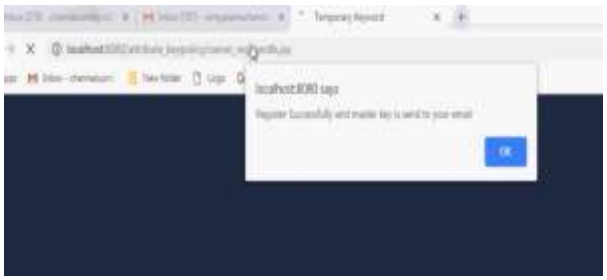
founder Michael Widenius's daughter and "SQL", the abbreviation for Structured Query Language.

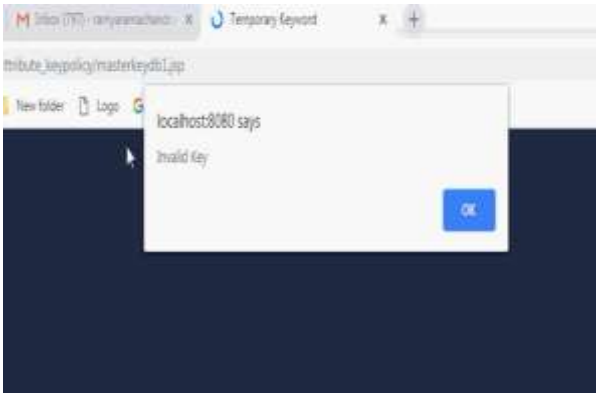
3. Results:



3.1 Screenshots:

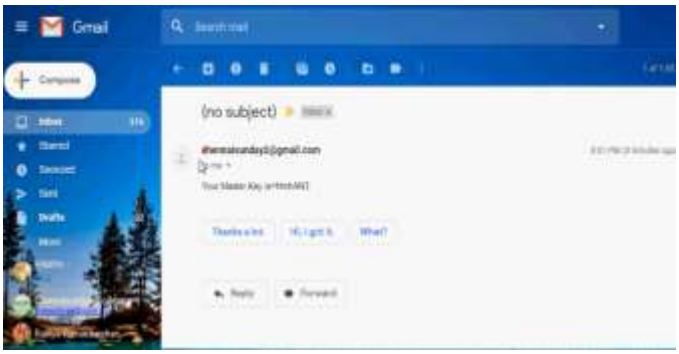
DATA OWNER REGISTRATION:





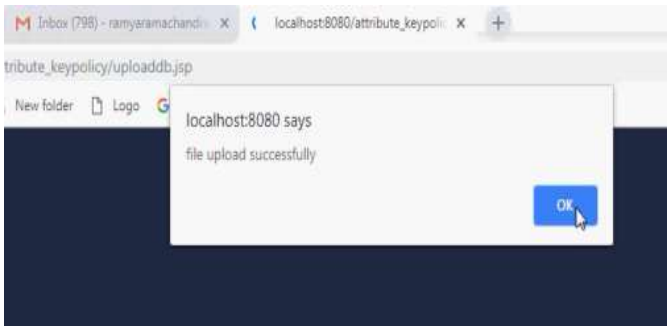
TTP LOGIN:

DATA OWNER DETAILS:

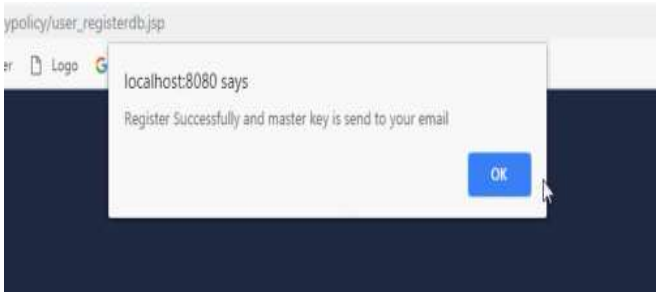


MASTER KEY FOR DATA OWNER:

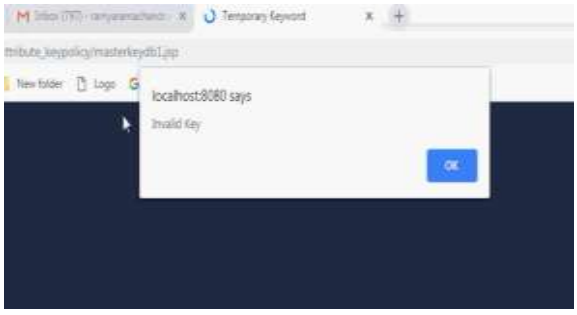
FILE UPLOAD:



DATA USER REGISTRATION:

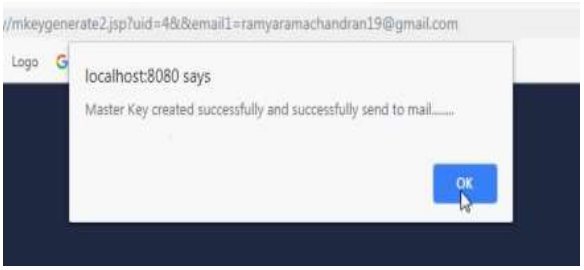


DATA USER LOGIN:



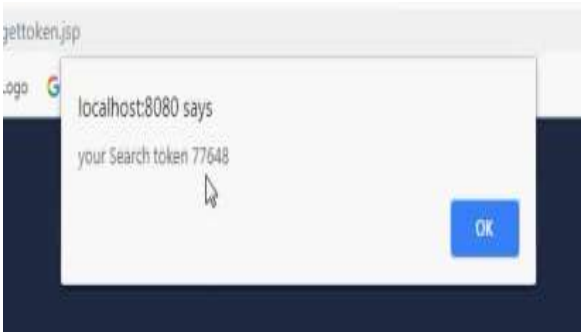
TTP LOGIN:

DATA USER DETAILS:



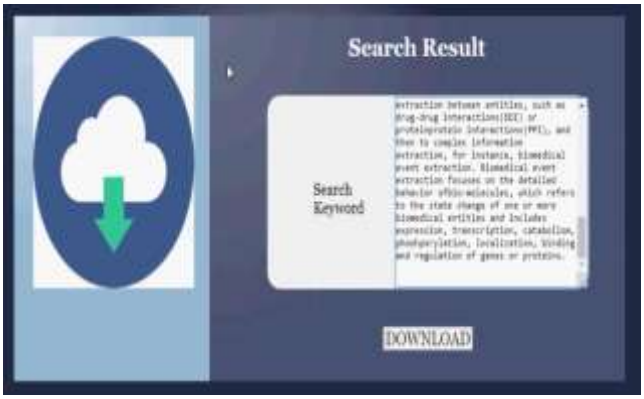
MASTER KEY FOR DATA USER:

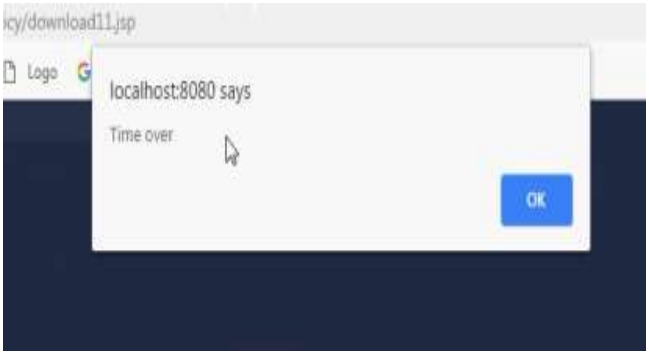
SEARCH TOKEN:



CLOUD FILES:

ORIGINAL FILE:





4. CONCLUSION

Securing cloud storage is an important problem in cloud computing. We addressed this issue and introduced the notion of key-policy attribute-based temporary keyword search (KPABTKS). According to this notion, each data user can generate a search token which is valid only for a limited time interval. We proposed the first concrete construction for this new cryptographic primitive based on bilinear map. We formally showed that our scheme is provably secure in the random oracle model. The complexity of encryption algorithm of our proposal is linear with respect to the number of the involved attributes. In addition, the number of required pairing in the search algorithms is independent of the number of the intended time unit's specified in the search token and it is linear with respect to the number of attributes. Performance evaluation of our scheme in term of both computational cost and execution time shows the practical aspects of the proposed scheme.

References:

[1] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search".
[2] Q. Zheng, S. Xu, and G. Ateniese, "Vabks: Verifiable attribute-based keyword search over outsourced encrypted data".
[3] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions".