IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

Volume 3, Issue 2, September 2023

Integrated Approach for Providing Data Security Verification Over Encrypted Cloud Data

Mr. Kuna Suresh

Department of Computer Science and Systems Engineering (A), Andhra University, Visakhapatnam, India

Abstract*:* *The cloud has recently attracted a lot of user attention from both small and large businesses, including those in software, BPO, healthcare, schools, colleges, and many other industries. For data storage and access from remote places linked to one another from a central server with the aid of the internet, all of these organisations attempt to adopt this centralised cloud server. Because all data is stored remotely and only occasionally accessed locally, the confidentiality of data is crucial to cloud service providers. As is common knowledge, no cloud service provider currently offers data privacy through encryption and message digests to enable data authorisation. almost all businesses An untrusted user can query data files of interest by sending encrypted keywords as a search query to the cloud server. Try to search the data in a secure manner over encrypted cloud data. In this dishonest cloud environment, the returned query results may occasionally be correct, incorrect, or partial. As everyone is aware, cloud servers today nearly actively withhold some qualified results in order to conserve computational resources and communication overhead. In this research, we presented and analysed a safe, practical, and fine-grained query results verification mechanism, by which the query user, given an encrypted query results set, not only can confirm the integrity of each data file, but also determines the overall number of qualifying data files, which are not returned if the set is not finished before the decryption procedure. This served as our primary inspiration for creating a brand-new secure verification object for encrypted cloud storage. Here, the short signature key is created using the message digest method MD5, which is also used to confirm the data's authenticity. We tested our proposed model in a variety of ways, and the results demonstrate that it is a useful and effective system.*

Keywords: cloud

I. INTRODUCTION

An authorised user can query data files of interest using secure search techniques over encrypted cloud data by sending encrypted search terms to the cloud server in a privacy-preserving way. However, in the dishonest cloud environment, the given query results may actually be inaccurate or lacking. For instance, in order to conserve computational resources and communication overhead, the cloud server may purposefully ignore some qualified results. Therefore, a well-performing secure query system should offer a mechanism for query results verification so that the data user can confirm the results. We create a safe, practical, and fine-grained query results verification technique in this work so that, given an encrypted query results set, the query user can not only confirm the accuracy of each data file in theIf the set is not complete before decryption, it is also possible to examine how many or which qualified data files are not returned.

Any secure query approach can very easily incorporate the verification scheme because it is loosely coupled to actual secure search methodologies. By creating secure verification objects for encrypted cloud data, we succeed in our purpose. Additionally, a verification object request approach is described to allow the query user to safely receive the appropriate verification object, as well as a brief signature technique with an incredibly low storage cost to ensure the authenticity of the verification object. Performance analysis demonstrates that the suggested schemes are workable and effective.

DOI: 10.48175/IJARSCT-13028

1.1 ENVIRONMENTAL CLOUD

Cloud computing: What is it?

Utilising computer resources (hardware and software) in the cloudare provided across a network (usually the Internet) as a service. The name is derived from the widespread use of a cloud-shaped symbol in system diagrams as a metaphor for the intricate infrastructure it holds. Cloud computing entrusts the data, software, and processing of a user to remote services. Hardware and software resources are made accessible via the Internet as managed third-party services in cloud computing. These services often give users access to cutting-edge server networks and sophisticated software programmes.

How Does the Cloud Work?

Using traditional supercomputing, or high-performance computing power, typically used by military and research facilities to perform tens of trillions of calculations per second, in consumer-oriented applications like financial portfolios, to deliver personalised information, topowering massive, immersive computer games or serving as data storage.

Networks of massive clusters of computers, often running low-cost consumer PC technology with specialised connections, are used in cloud computing to distribute data processing tasks among them. Large networks of interconnected systems make up this shared IT infrastructure. The power of cloud computing is frequently maximised through the usage of virtualization techniques.

Models for characteristics and services:

- •According to the National Institute of Standards and Terminology's (NIST) definitions, the key aspects of cloud computing are as follows:
- On-demand self-service: A customer can automatically supply computing resources as needed, such as server time and network storage, without needing to deal with the supplier of each service in person.
- Wide network access: Capabilities can be accessed and made available across the network.by use of common mechanisms that encourage usage by diverse thin- or thick-client platforms (such as cell phones, PCs, and PDAs).
- Resource pooling: Using a multi-tenant approach, the provider's computing resources are combined to serve numerous customers, with distinct physical and virtual resources being dynamically assigned and reassigned in response to customer demand. The customer typically has no control or knowledge over the precise location of the resources given, but they might be able to designate location at a higher level of abstraction (for example, country, state, or data centre). This gives the customer a perception of location independence. Storage, computation, memory, network bandwidth, and virtual machines are a few examples of resources.

Cloud computing advantages

1. Achieve economies of scale to boost productivity or volume output with fewer workers. Your price perunit, project, or item falls apart.
2. Cut back on infrastructure spending for technology. Maintain simple access to your data with little initial outlay. Depending on demand, pay as you go (weekly, quarterly, or annually).
3. Cheaply globalise your staff. Anyone with an Internet connection can access the cloud from anywhere in the globe.
4. Simplify procedures. Work more efficiently and with fewer personnel.
5. Lower capital expenses. There is no need to spend a lot of money on licencing costs, hardware, or software.
6. Make accessibility better. Your life is made so much easier by having access whenever and wherever you want!
7. Improved project oversight. Keep costs under control and ahead of the completion cycle times.
8. Less staff training is required. Fewer employees are required to complete more work on acloud, with a low learning curve for software and hardware problems.

DOI: 10.48175/IJARSCT-13028

II. LITERATURE SURVEY AND RELATED WORK

INTRODUCTION

The most crucial stage of the software development process is the literature review. Determine the time factor, economics, and company strength prior to building the tool. The next steps are to decide which operating system and language were utilised to construct the tool if these requirements have been met. Once the programmers begin creating the tool, they require a lot of outside assistance. This assistance was gathered from senior programmers, books, or websites. The aforementioned factors were taken into account before constructing the proposed system.

CONNECTED WORK

Challenges with public cloud security K. Ren and C. Wang

The most interesting paradigm shift in computing occurring today in information technology is represented by cloud computing. Security and privacy concerns, nevertheless, are seen as the main barriers to its widespread use. The authors present a number of important security issues here and encourage more research into security options for a reliable public cloud environment.

1) Cloud storage using cryptography S. Kamara and K. Lauter wrote the book.

We take into account the challenge of constructing a secure cloud storage service on top of a public cloud architecture when the consumer does not fully trust the service provider. In order to accomplish our goal, we briefly discuss a number of architectures that combine modern and unusual cryptographic primitives. We review the advantages that such an architecture would offer to users and service providers, as well as present an overview of recent developments in cryptography that are specifically driven by cloud storage.

2) Useful methods for decrypting data searches AUTHORS: D. Song, D. Wagner

To lower security and privacy risks, it is preferable to keep data on data storage servers, such as mail servers and file servers, in encrypted form. However, this typically implies that functionality must be given up for security. It was previously unknown how to allow the data storage server to do the search and provide the query without compromising the secrecy of the data, for instance, if a client wants to obtain just documents that include a specific set of terms. We present security proofs for the resulting crypto systems as well as a description of our cryptographic solutions to the challenge of searching encrypted data. Our methods offer several very important benefits. They may be proven to be safe since they offer encryption with provable secrecy, meaning an unreliable server cannot They offer query isolation for searches, which means that an untrusted server cannot learn anything about the plaintext other than the search result; they offer controlled searching, which prevents the untrusted server from searching for any word without the user's permission; they also support hidden queries, which allow the user to ask the untrusted server to look for a secret word without disclosing the word's identity. The described techniques are easy to use, quick (for a document of length n, the encryption and search algorithms only need O(n) stream cypher and block cypher operations), and virtually overhead-free in terms of both space and communication.

3) Fuzzy public-key encryption A proven secure method that can withstand a keyword guessing assault
AUTHORITIES: P. Xu and H. Jin

A flexible tool is public-key encryption with keyword search (PEKS). It enables a third party with knowledge of a keyword's search trapdoor to look up encrypted documents containing that keyword without having to decrypt the documents or know the keyword itself. However, it is demonstrated that if the keyword space is of a polynomial size, a malicious third party could compromise the keyword under a keyword guess attack (KGA). Public-key encryption with fuzzy keyword search (PEFKS), a form of PEKS with keyword privacy enhancement, is how we solve this issue. Each keyword in PEFKS has both an exact and a fuzzy keyword search trapdoor corresponding to it. at least two keywords similar fuzzy keyword trapdoors. Only the fuzzy keyword search trapdoor is made available to the third party, or the searcher, in order to search encrypted documents that include a particular keyword. Because of this, even if the keyword space is limited, PEFKS prevents malevolent searchers from learning the precise keyword to be searched. We provide an all-encompassing transformation that turns any IBE (identity-based anonymous encryption) system into a

DOI: 10.48175/IJARSCT-13028

safe PEFKS scheme. Using the generic architecture, we create the first PEFKS scheme that has been demonstrated to be secure under KGA in the scenario when the keyword space is polynomial in size.

4) Symmetric encryption that is concurrent and dynamically searchable S. Kamara and C. Papamanthou are the authors. A client can outsource a set of encrypted documents using searchable symmetric encryption (SSE). without disclosing information about the contents of the documents and inquiries, but still maintaining the capacity to do keyword searches. Although there are effective SSE structures, the existing solutions are rather sequential. This is primarily because the inverted index approach (Curtmola, Garay, Kamara, and Ostrovsky, CCS '06), the only technique currently available for achieving sub-linear time search, requires the search algorithm to access a series of unpredictable memory locations stored at the previous location in the sequence. We offer a new technique for creating sub-linear SSE schemes that is inspired by developments in multi-core architectures. Our method is very dynamic and parallelizable. Searches for a keyword w are performed with roughly a logarithmic number of cores in situ. Our algorithm runs in parallel in o(r) time, where r is the number of documents that include the keyword w (additional cores allow for a lower bound of O(logn), which is independent of the result size r). This time complexity is faster than the optimum (r) sequential search time; the updates are bound by a similar bound. Additionally, our plan accomplishes the following crucial characteristics: In comparison to other sub-linear dynamic SSE schemes (such as Kamara, Papamanthou, and Roeder, CCS '12), our scheme has three strong security features: (a) security against adaptive chosen-keyword attacks; (b) updates that do not leak information other than that which can be deduced from prior search tokens; and (c) efficient implementation in external memory (with logarithmic I/O overhead).

III. PROPOSED WORK AND ALGORITHM

We explicitly propose the verifiable secure search system model, threat model, and construct a fine-grained query results verification scheme for safe keyword search over encrypted cloud data as the main contributions of this research. We suggest a quick signature method based on public-key cryptography without certificates to ensure the veracity of the verification objects themselves.

We develop a novel Paillier encryption-based verification object request method where the cloud server has no idea what data the user is asking or which verification objects will be delivered to the user.

BENEFITS OF THE PROPOSED SYSTEM

1) In order to assess the precision and effectiveness of our suggested scheme, we offer the formal security definition and proof and carry out thorough performance experiments.
2) Our system can confirm the accuracy of Find out precisely how many or which qualifying data files are returned by the dishonest cloud server per each encrypted query result.
3) A quick signature method is intended to ensure the validity of the verification object itself.

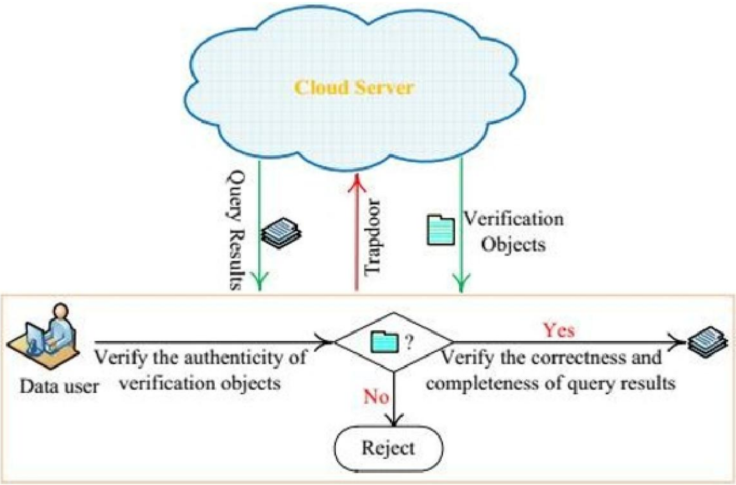

Cloud Server
Query Results
Trapdoor
Verification Objects
Data user
Verify the authenticity of verification objects
?
Yes
Verify the correctness and completeness of query results
No
Reject

Fig 1:System architecture

IV. METHODOLOGIES

Goals JDBC

Software programmes are rarely created without objectives in mind. One that influenced the creation of the API was JDBC because of its diverse objectives. These aims have helped to shape the JDBC class library into a reliable platform for creating database applications in Java, along with early reviewer criticism.

The objectives for JDBC are significant. They will help you understand why particular classes and functions act in the manner that they do. The eight JDBC design objectives are as follows:

One SQL Level API

The creators believed that creating a Java SQL interface was their primary objective. The database interface level is not the lowest one that can be achieved, but it is low enough for higher-level tools and APIs to be produced. On the other hand, it is competent enough for application programmers to use. By achieving this objective, tool providers in the future will be able to "generate" JDBC code and hide many of JDBC's complexity from end users.

SQL compliance

As you switch between different database vendors, the SQL syntax changes. Any query statement may be provided through JDBC to the underlying database driver in an effort to support a wide range of vendors. By doing so, the connectivity module is able to manage non-standard functionality in a way that is appropriate for its customers.

JDBC needs to be implemented on top of standard database interfaces Other widespread SQL level APIs must "sit" underneath the JDBC SQL API. This objective permits JDBC utilises an existing software interface to access ODBC level drivers. This interface would convert ODBC calls to JDBC calls and the other way around.

Modulae, the System Framework, the Data Owner, the Data User, and the Cloud Server

V. MODULES DESCRIPTION

System Architecture:

In this framework, we design a safe, simple-to-integrate, and fine-grained query results verification mechanism, by which the query user, given an encrypted query results set, can not only verify the accuracy of each data file in the set but also check how many or which qualified data files are not returned if the set is incomplete before decryption. Any secure query approach can very easily incorporate the verification scheme because it is loosely coupled to actual secure search methodologies. We succeed in our mission by creating a secure verification object for cloud data that has been encrypted. Additionally, a verification object request approach is described to allow the query user to safely receive the appropriate verification object, as well as a brief signature technique with an incredibly low storage cost to ensure the authenticity of the verification object. Performance analysis demonstrates the viability and effectiveness of the suggested schemes. The Data Owner, Data User, and Cloud Server modules are used here.

Owner of Data:

Data owners must first register their information in the Data Owner module.Following a successful registration, the data owner can log in and upload files using encrypted keywords and hashing algorithms to a cloud server. He or she can see the cloud-uploaded files. The data owner may accept or reject the transmitted file request. by users of data. Following request approval, the data owner will mail the verification object and trapdoor key.

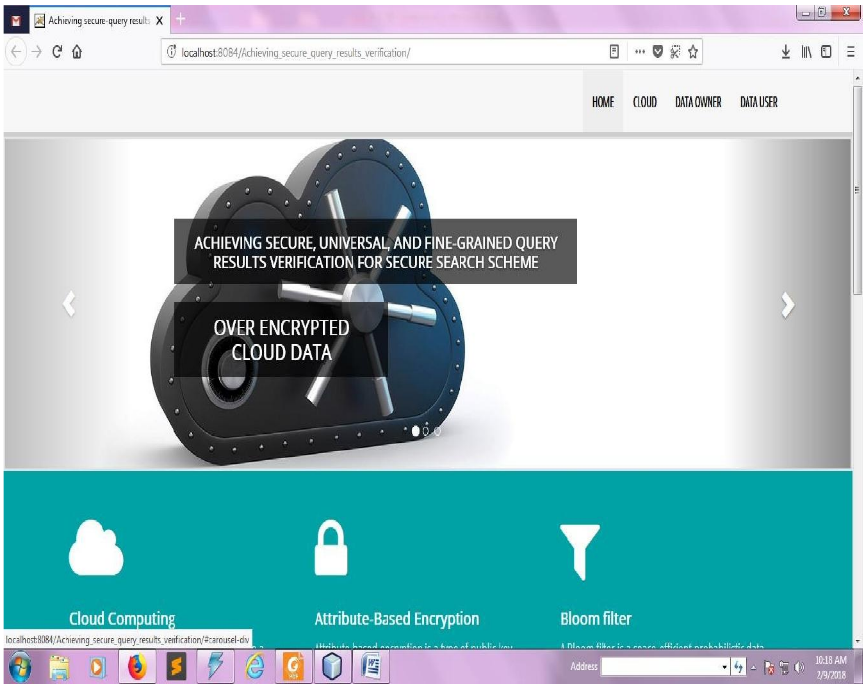
Data User: In the Data User module, Data Users must first register their information before logging in. They then must confirm their login using a secret key. All of the files that data owners submit can be searched by data users. He or she can send requests to the files, which will then send requests to the owners of the data. If the data owner approves the request, the trapdoor, verification object, and decryption key will be sent to them by registered mail.

Online storage:

Cloud Provider can examine all file details in the Cloud Server module. The cloud server can view the download history and alter and update files.

DOI: 10.48175/IJARSCT-13028

VI. RESULTS AND DISCUSSION SCREENSHOTS



Achieving secure-query result
localhost:8084/Achieving_secure_query_results_verification/
HOME
CLOUD
DATA OWNER
DATA USER
Cloud Computing
Attribute-Based Encryption
Bloom filter
localhost:8084/Achieving_secure_query_results_verification/#carousel-div
Address
10:18 AM
2/9/2018

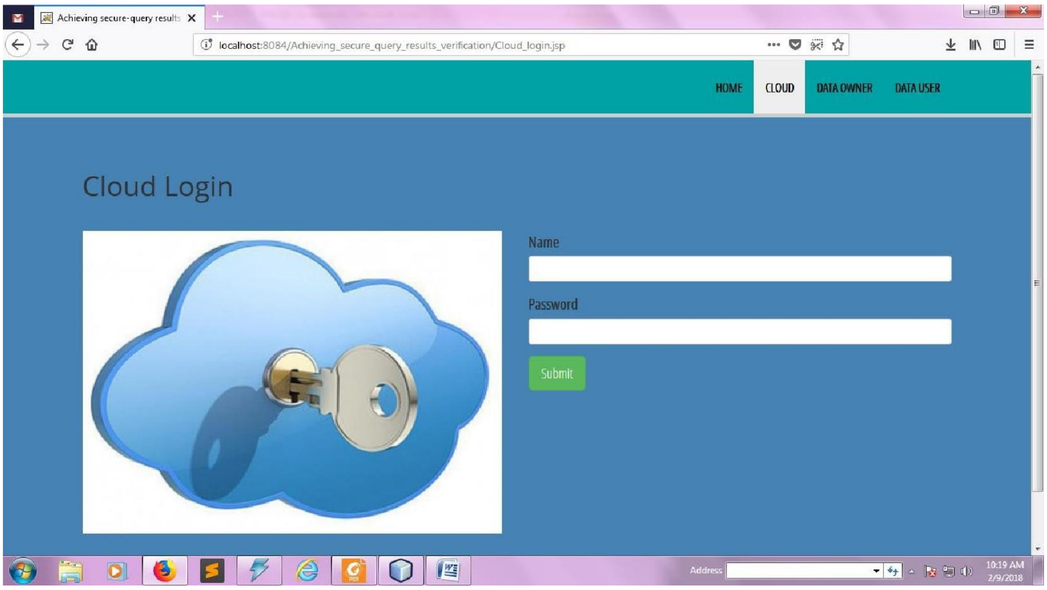
ACHIEVING SECURE, UNIVERSAL, AND FINE-GRAINED QUERY RESULTS VERIFICATION FOR SECURE SEARCH SCHEME

OVER ENCRYPTED CLOUD DATA

Attribute based encryption is a type of public key

A Bloom filter is a space-efficient probabilistic data

Fig 2: home page



Achieving secure-query result
localhost:8084/Achieving_secure_query_results_verification/Cloud_login.jsp
HOME
CLOUD
DATA OWNER
DATA USER
Submit
Address
10:19 AM
2/9/2018

Cloud Login

Name

Password

Fig 3:- cloud login page

DOI: 10.48175/IJARSCT-13028

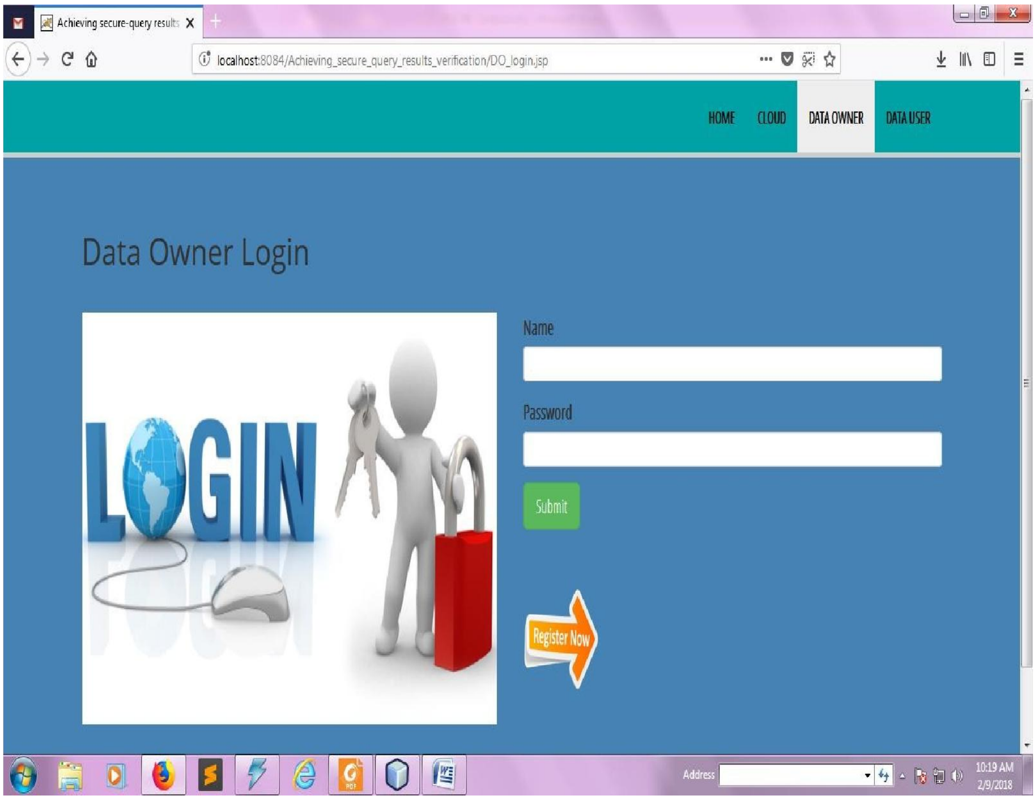
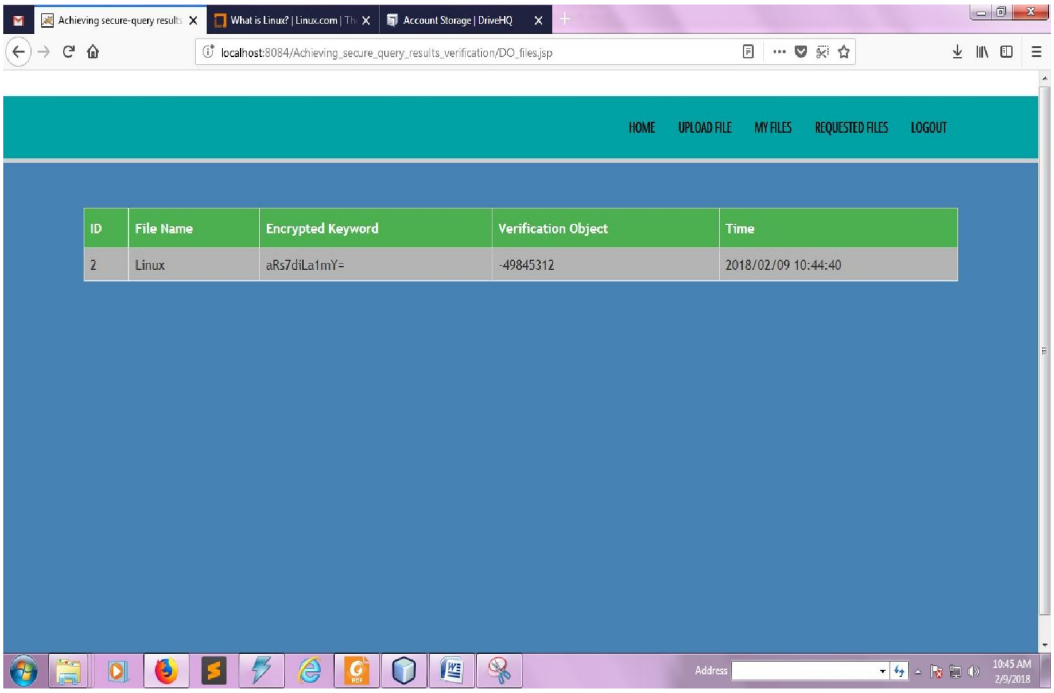


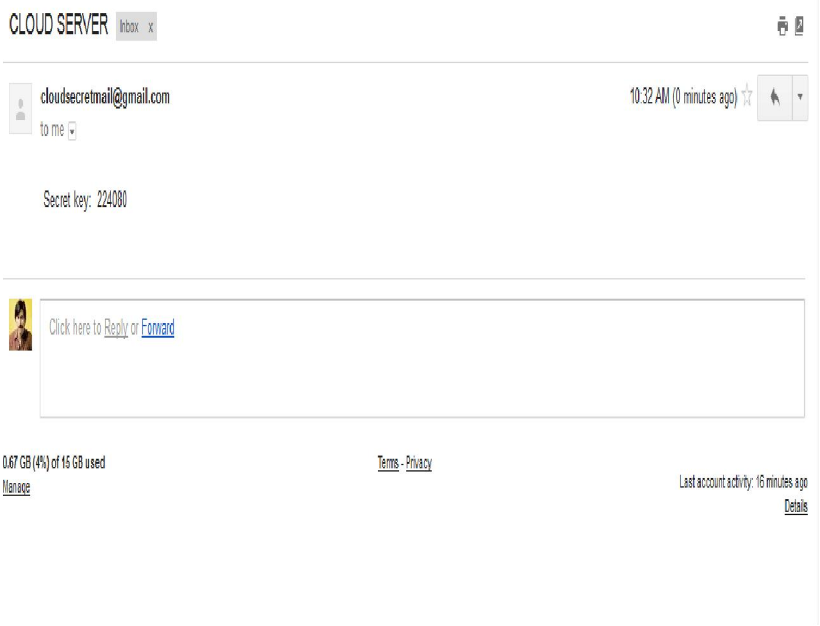
Fig 4:-Data owner login page


Achieving secure-query results
What is Linux? | Linux.com | Th
Account Storage | DriveHQ
localhost:8084/Achieving_secure_query_results_verification/DO_files.jsp
HOME
UPLOAD FILE
MY FILES
REQUESTED FILES
LOGOUT
Address
10:45 AM
2/9/2018

ID	File Name	Encrypted Keyword	Verification Object	Time
2	Linux	aRs7diLa1mY=	-49845312	2018/02/09 10:44:40

Fig 5:- Data owner uploaded file with secret key

DOI: 10.48175/IJARSCT-13028



CLOUD SERVER Inbox x

cloudsecretmail@gmail.com
to me

10:32 AM (0 minutes ago)

Secret key: 224080

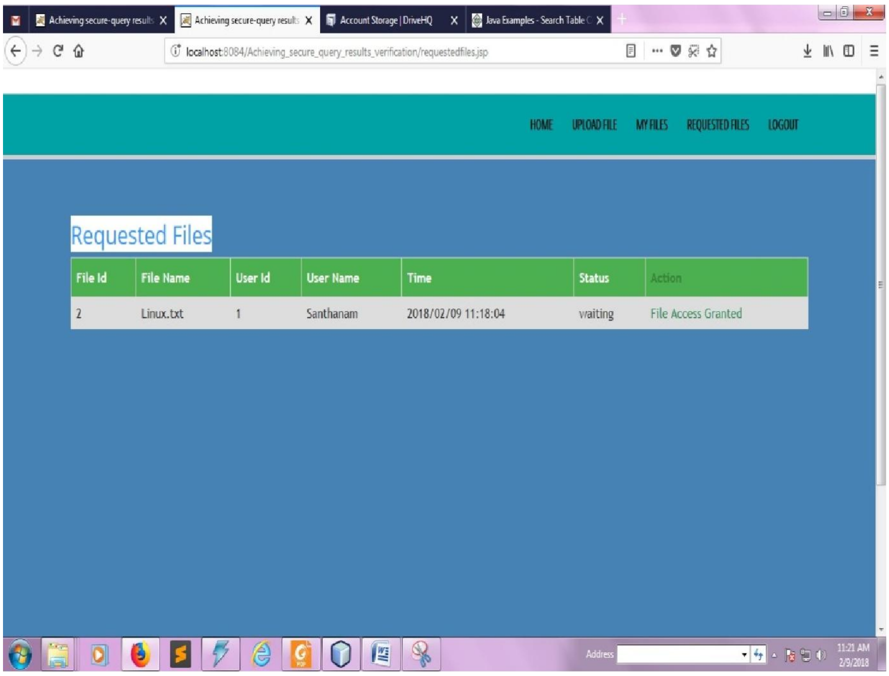
Click here to Reply or Forward

0.67 GB (4%) of 15 GB used
Manage

Terms - Privacy

Last account activity: 16 minutes ago
Details

Fig 5:-Secret key sent to mail

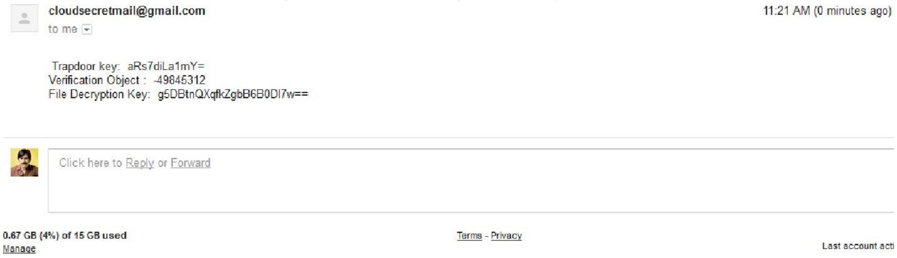


Achieving secure-query results
Achieving secure-query results
Account Storage | DriveHQ
Java Examples - Search Table C
localhost:8084/Achieving_secure_query_results_verification/requestedfiles.jsp
HOME
UPLOAD FILE
MY FILES
REQUESTED FILES
LOGOUT
Address
11:21 AM
2/9/2018

Requested Files

File Id	File Name	User Id	User Name	Time	Status	Action
2	Linux.txt	1	Santhanam	2018/02/09 11:18:04	waiting	File Access Granted

Fig 6:-File access granted by cloud



cloudsecretmail@gmail.com
to me

11:21 AM (0 minutes ago)

Trapdoor key: aRs7diLa1mY=
Verification Object : -49845312
File Decryption Key: g5DBtnQXqfkZgbB6B0DI7w==

Click here to Reply or Forward

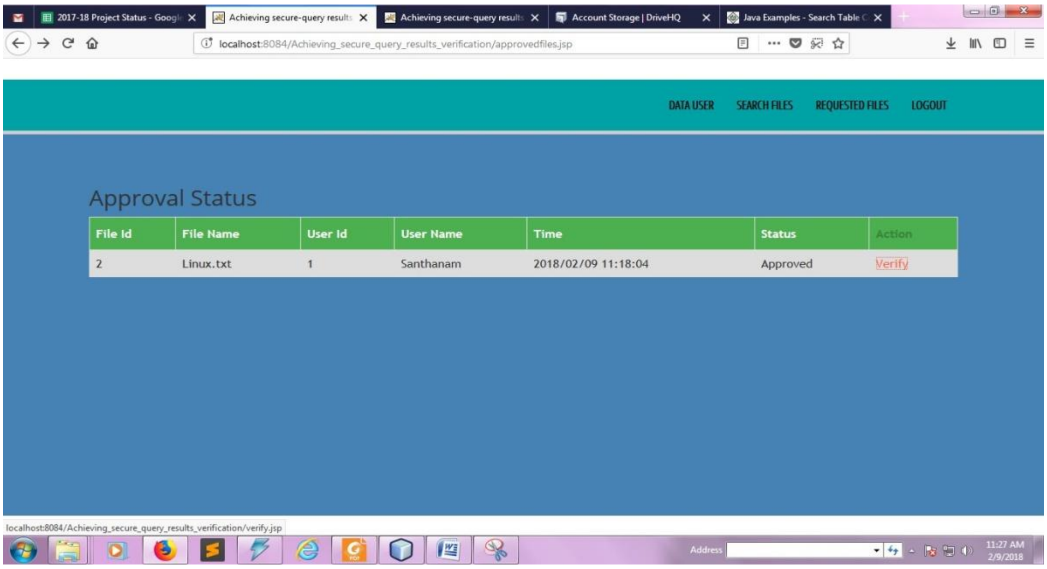
0.67 GB (4%) of 15 GB used
Manage

Terms - Privacy

Last account acti

Fig 6:-verification key and trapdoor key send to user mail id

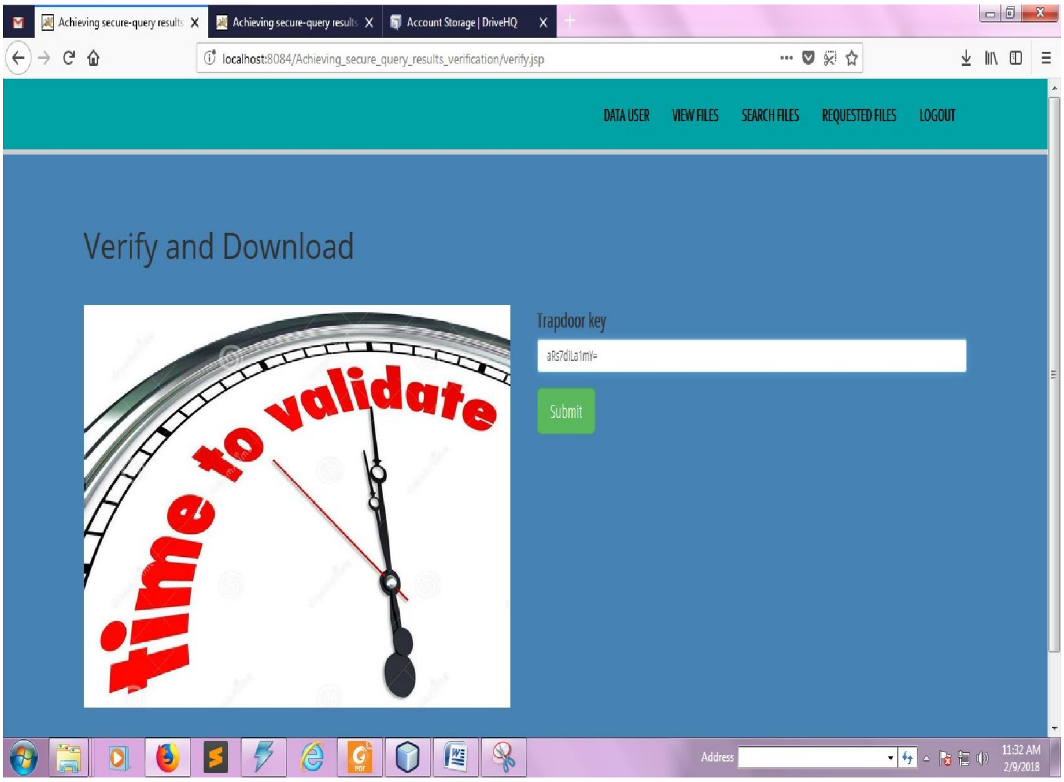
DOI: 10.48175/IJARSCT-13028


localhost:8084/Achieving_secure_query_results_verification/approvedfiles.jsp
DATA USER
SEARCH FILES
REQUESTED FILES
LOGOUT
localhost:8084/Achieving_secure_query_results_verification/verify.jsp

Approval Status

File Id	File Name	User Id	User Name	Time	Status	Action
2	Linux.txt	1	Santhanam	2018/02/09 11:18:04	Approved	Verify

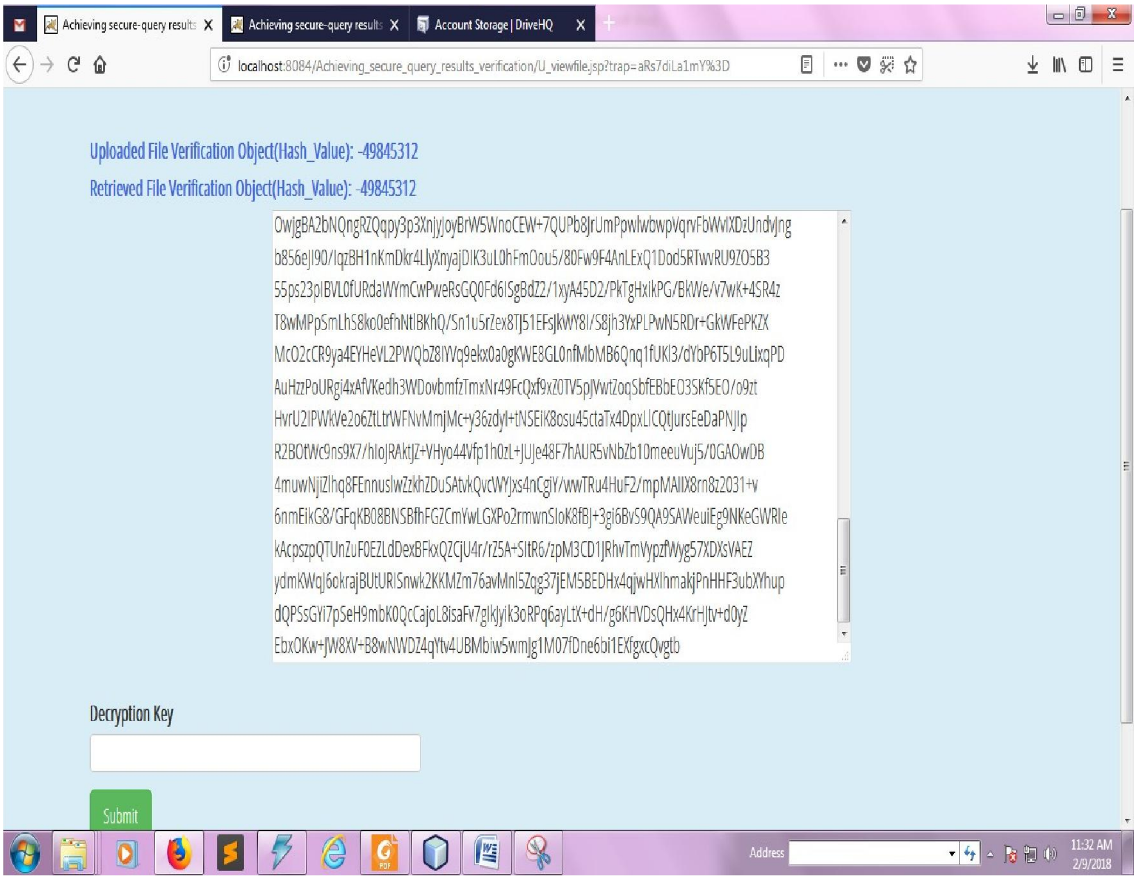
Fig 7:-Approval status form



Verify and Download

Trapdoor key: aRs7diLa1mY=

Fig 8:- Trapdoor key to download


Achieving secure-query results
Achieving secure-query results
Account Storage | DriveHQ
localhost:8084/Achieving_secure_query_results_verification/U_viewfile.jsp?trap=aRs7diLa1mY%3D
Address
11:32 AM
2/9/2018

Uploaded File Verification Object(Hash_Value): -49845312

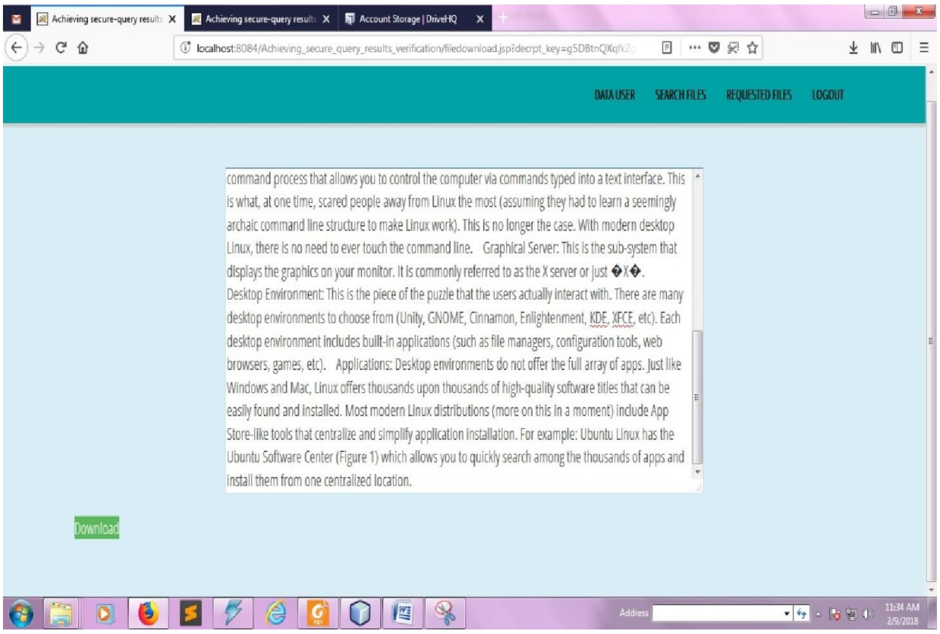
Retrieved File Verification Object(Hash_Value): -49845312

```
OwjgBA2bNQngRZQqpy3p3XnjyJoyBrW5WnoCEW+7QUPb8JrUmPpwlwbwpVqrvFbWvlXDzUndvJng
b856eJI90/IqzBH1nKmDkr4LlyXnyajDIK3uL0hFmOou5/80Fw9F4AnLExQ1Dod5RTwvRU9ZO5B3
55ps23pIBVL0fURdaWYmCwPweRsGQ0Fd6ISgBdZ2/1xyA45D2/PkTgHxikPG/BkWe/v7wK+4SR4z
T8wMPpSmLhS8ko0efhNtIBKhQ/Sn1u5rZex8TJ51EFsJkWY8I/S8jh3YxPLPwN5RDr+GkWFePKZX
McO2cCR9ya4EYHeVL2PWQbZ8IYVq9ekx0a0gKWE8GL0nfMbMB6Qnq1fUKl3/dYbP6T5L9uLixqPD
AuHzzPoURgi4xAfVKedh3WDovbmfzTmxNr49FcQxf9xZ0TV5pJVwtZoqSbfEBbEO3SKf5EO/o9zt
HvrU2IPWkVe2o6ZtLtrWFNvMmjMc+y36zdyI+tNSEiK8osu45ctaTx4DpxLICQtJursEeDaPNJIp
R2BOtWc9ns9X7/hIoJRAktJZ+VHyo44Vfp1h0zL+JUJe48F7hAUR5vNbZb10meeuVuj5/0GAOwDB
4muwNjiZIhq8FEnnusIwZzkhZDuSAtvkQvcWYJxs4nCgiY/wwTRu4HuF2/mpMAIIX8rn8z2031+v
6nmEikG8/GFqKB08BNSBfhFGZCmYwLGXPo2rmwnSIoK8fBJ+3gi6BvS9QA9SAWeuiEg9NKeGWRIe
kAcpszpQTUnZuF0EZLdDexBFkxQZCjU4r/rZ5A+SItR6/zpM3CD1JRhvTmVypzfWyg57XDXsVAEZ
ydmKWqJ6okrajBUtURISnwk2KKMZm76avMnl5Zqg37jEM5BEDHx4qjwHXlhmakjPnHHF3ubXYhup
dQPSsGYi7pSeH9mbK0QcCajoL8isaFv7gIkJyik3oRPq6ayLtX+dH/g6KHVDsQHx4KrHJtv+d0yZ
EbxOKw+JW8XV+B8wNWDZ4qYtv4UBMbiw5wmJg1M07fDne6bi1EXfgxcQvgtb
```

Decryption Key

Submit

Fig 9:-Decryption key to Decrypt


Achieving secure-query results
Achieving secure-query results
Account Storage | DriveHQ
localhost:8084/Achieving_secure_query_results_verification/filedownload.jsp?decrpt_key=g5DBtnQXqfkZg
DATA USER
SEARCH FILES
REQUESTED FILES
LOGOUT
Address
11:34 AM
2/9/2018

command process that allows you to control the computer via commands typed into a text interface. This is what, at one time, scared people away from Linux the most (assuming they had to learn a seemingly archaic command line structure to make Linux work). This is no longer the case. With modern desktop Linux, there is no need to ever touch the command line.   Graphical Server: This is the sub-system that displays the graphics on your monitor. It is commonly referred to as the X server or just �X�.
Desktop Environment: This is the piece of the puzzle that the users actually interact with. There are many desktop environments to choose from (Unity, GNOME, Cinnamon, Enlightenment, KDE, XFCE, etc). Each desktop environment includes built-in applications (such as file managers, configuration tools, web browsers, games, etc).   Applications: Desktop environments do not offer the full array of apps. Just like Windows and Mac, Linux offers thousands upon thousands of high-quality software titles that can be easily found and installed. Most modern Linux distributions (more on this in a moment) include App Store-like tools that centralize and simplify application installation. For example: Ubuntu Linux has the Ubuntu Software Center (Figure 1) which allows you to quickly search among the thousands of apps and install them from one centralized location.

Download

Fig 10:-File decrypted after entering decryption key

DOI: 10.48175/IJARSCT-13028

VII. CONCLUSION

In this research, we suggest a fine-grained query results verification scheme for secure search over encrypted cloud data that is both safe and simple to incorporate. In contrast to earlier research, our scheme can properly determine how many or which qualified data files are returned by the dishonest cloud server, as well as validate the accuracy of each encrypted query result. A quick signature method is intended to ensure the validity of the verification object itself. Additionally, we develop a secure verification object request method where the cloud server is unaware of the verification objects that the data user has requested and really received in the form of returned bytes. Experiments on performance and accuracy show that our suggested approach is effective and legitimate.

REFERENCES

[1] P. Mell and T. Grance, “The nist definition of cloud computing,” http://dx.doi.org/10.602/NIST.SP.800-145.

[2] K. Ren, C. Wang, and Q. Wang, “Security challenges for the public cloud,” IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, 2012.

[3] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in Springer RLCPS, January 2010.

[4] D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in IEEE Symposiumon Security and Privacy, vol. 8, 2000, pp. 44–55.

[5] E.-J.Goh, “Secure indexes,” IACR ePrint Cryptography Archive, http://eprint.iacr.org/2003/216, Tech. Rep., 2003.

[6] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public-key encryption with keyword search,” in EUROCRYPR, 2004, pp. 506–522.

[7] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved deinitions and efficient constructions,” in ACM CCS, vol. 19, 2006, pp. 79–88.

[8] M. Bellare, A. Boldyreva, and A. O’Neill, “Deterministic and efficiently searchable encryption,” in Springer CRYPTO, 2007.

[9] K. Kurosawa and Y. Ohtaki, “Uc-secure searchable symmetric encryption,” Lecture Notes in Computer Science, vol. 7397, pp. 258–274, 2012.

[10] P. Xu, H. Jin, Q. Wu, and W. Wang, “Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack,” IEEE Transactions on Computers, vol. 62, no. 11, pp. 2266–2277, 2013.

[11] S. Kamara and C. Papamanthou, “Parallel and dynamic searchable symmetric encryption,” in Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2013, pp. 258–274.

[12] M. Naveed, M. Prabhakaran, and C. A. Gunter, “Dynamic searchable encryption via blind storage,” in IEEE S&P, May 2014, pp. 639–654.

[13] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, “Secure ranked keyword search over encrypted cloud data,” in IEEE ICDCS, 2010, pp. 253–262.

[14] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” in IEEE INFOCOM, 2011, pp. 829–837.

[15] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, “Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking,” in ACM ASIACCS, 2013.

[16] B. Wang, S. Yu, W. Lou, and Y. T. Hou, “Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud,” in IEEE INFOCOM, 2014, pp. 2112–2120.

DOI: 10.48175/IJARSCT-13028