Enhancement of the Homomorphism Encryption Based Mechanism by Solving the Integrity Problems Effectively on Multi Keyword Search in Cloud Computing

Vishal, Bikrampal Kau, Surender Jangra

Abstract: ***Cloud computing gives part of advantages to endeavors to offload their information and software services to cloud administrations by saving them lot of money that must be spent on foundation setup cost. Despite that, while offloading the information privacy and protection is a vital concern. Correspondingly for example when the emergency clinics and medicinal services association transfer their patient's subtleties to cloud and when the information is included, it will influence the protection of the patients. So privacy is a vital concern while offloading the information to cloud. Most of solutions for privacy depend on encryption and information to be offloaded is encoded and put away in cloud. Lots of encryption techniques like cryptography, symmetric key etc. methods are used for encrypting the information before uploading to cloud. However, the reaction in this encryption system is that, the encrypted information isn't structure preserving and it is not suitable for ranking or searching. To solve the complications of this searching and ranking due to disordering a homomorphism encryption based mechanism is proposed but this solution also suffers from the problem of integrity attacks on result. So this paper discussed the enhancement of the homomorphism encryption based mechanism by solving the integrity problems effectively.***

Keywords: AES- Advanced Encryption System, Cloud Computing, Cryptography, Data privacy, Integrity Attacks, Multi Keyword Search.

I. INTRODUCTION

Cloud computing provides lot of benefits to enterprises to offload their data and software services to cloud saving them lot of money that has to be spent on infrastructure setup cost. Enterprises wanted to offload their data to cloud and save on their infrastructure cost.

But when offloading the data security and privacy is a important concern.Nowadays there are lot of cloud service providers in the market.

The two requirements of security and privacy are the important decision criteria for enterprises in choosing the cloud service providers.For security many encryption protocols were proposed which encrypt the enterprise data and offload to cloud, so that for other cloud users the data is difficult to decrypt and make sense.

Cloud service providers encrypt the enterprise data and save in cloud. But the problem is when some enterprise user want to search for some documents based on keyword, it becomes difficult for him to download all documents, decrypt and search. The cost of downloading the documents and decrypting them to search every time is costly. Also the time needed for search is high in this way.In the paper work [1], author has proposed a solution called MRSE for multi keyword search on cloud. Among various multi-keyword semantics, authors choosed the efficient principle of “coordinate matching”, i.e., as many matches as possible, to capture the similarity between search query and data documents. Specifically, they use “inner product similarity” [2], i.e., the number of query keywords appearing in a document, to quantitatively evaluate the similarity of that document to the search query in “coordinate matching” principle. During index construction, each document is associated with a binary vector as a sub-index where each bit represents whether corresponding keyword is contained in the document [3]. The search query is also described as a binary vector where each bit means whether corresponding keyword appears in this search request, so the similarity could be exactly measured by inner product of query vector with data vector. However, directly outsourcing data vector or query vector will violate index privacy or search privacy. To meet the challenge of supporting such multikeyword semantic without privacy breaches, we propose a basic MRSE scheme using secure inner product computation, which is adapted from a secure k-nearest neighbor (kNN) technique [4], and then improve it step by step to achieve various privacy requirements in two levels of threat models.

The problem with this approach is that the search result can be attacked and ranking be modified. Say an example of user searching for hotels nearby a city, the search results comes in rank of closet distance, an attacker in middle can modify the search result to move a selected hotel up in order even though it was not in top 10 or so. In this project we discuss about this problem and propose a effective solution against this integrity attack. To solve it in work [1], author has proposed homomorphic encryption based mechanism but the solution suffers from integrity attack on search result.

Manuscript published on November 30, 2019.
* Correspondence Author

Vishal, Assistant Professor Department of computer Science and Engineering, Jai Parkash Mukand Lal Innovative Engineering and Technology Institute, Radaur.

Dr.Bikram Pal Kaur, Assistant Professor, Department of Computer Science, GTB College, Bhawanigarh (Sangrur),

Dr.Surender, ssistant Professor Department of computer Science and Engineering, Jai Parkash Mukand Lal Innovative Engineering and Technology Institute, Radaur.

In this paper, we discuss the integrity problem and solve it effectively.

Cloud security offers the wide scope of solutions for technologies and policies used to ensure information in any association using the cloud computing model [1].

Based on the requirement for the security these solutions can be classified as:

1. To verify activity of end mobile users
2. To improve security of cloud server centres and protect them from attacks

In mobile cloud security different kinds of encryption systems have been actualized. Reference to these techniques can be found in the literature, which incorporates dividing the information to different cloud as per the dimension of integrity and secrecy required by them and connected to encryption strategies. Today, Data owners have apprehensive from CSP for their information security and unapproved access [2].

The main objectives and proposed work are:

1. To analyze the existing approach of security in mobile cloud computing environment.
2. To design a security system for Mobile Cloud Computing environment.
3. To evaluate the performance of developed mobile cloud security framework.

The issue with this approach is about privacy protection that leads to the integrity attack on searching and ranking results [3]. To solve it in work , author has proposed a homomorphic encryption based component however this solution also experiences integrity assault on query item[4]. In this paper, we talk about the integrity issue and propose a solution against this integrity attack. In this project we discuss about this problem and propose an effective solution against this integrity attack [5].

A. Mobile Cloud Computing

Mobile Cloud Computing- MCC is a rich mobile technology that uses flexible assets of multiple clouds and network methods toward unrestricted functionality, mobility and storage to serve a multitude of mobile devices anywhere. Zohreh Sanaei et al. [6]. Mobile cloud computing alludes to a unique approach where both the information storage and the security occur outside of the mobile device. Mobile users' requests the data and it is transmitted to the central units associated with their particular servers giving network services. The users' requests are conveyed to a cloud; the cloud controllers process the queries to the relating cloud administrations.

B. Data security and Privacy

In MRSE, for data privacy data is uploaded to cloud after encoding it by using conventional symmetric encryption algorithm and further uploaded to the public cloud server. With the help of metadata set using the keys provided by public cloud server private cloud server develops the index. For data retrieval or downloading process the inverted index is distributed to the public cloud server [7]. The index is secured from the integrity attacks using this index and inverted index approach.

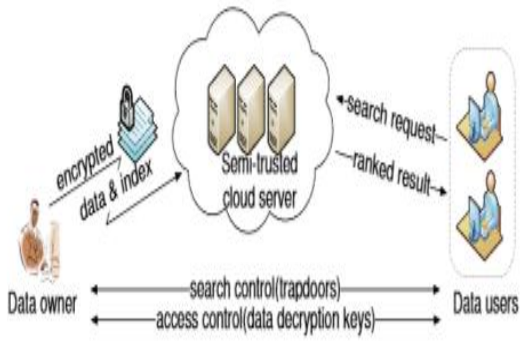

encrypted data & index
Semi-trusted cloud server
search request
ranked result
Data owner
search control(trapdoors)
access control(data decryption keys)
Data users

Fig.1. Architecture of MRSE [7]

The query processed for searching data is also encrypted and delivered to cloud for storage. Here there is a security problem in the MRSE; the search query can be deduced by inference attack by matching the query to the search result. Once the cloud server processes the search query against the index, the ranked search results are returned. MRSE provides the ranked results also in an encrypted form to avoid mapping between the search query and the search results. But the ranking order does not change for the same search query. By exploiting this property, attacker can manipulate the search results for his own selfish needs. Fig. 1 above shows the architecture of MRSE.

In this search control and access control is given more importance and any attacks on this are very difficult. But the order of ranked result can be affected and it is not possible for the data owner to realize the search result is attacked. It is also the case that some other search result can be given to query to mislead the users [8].

II. LITERATURE REVIEW

Below literature survey mainly focus on security concerns in mobile cloud computing and cloud computing environment. As it is the hottest buzzword and has been considered for various types of online solutions. Mobile Cloud computing model leaves the clients vulnerable to different types of attacks and threats.

Jun Zhou et al. proposed a security and privacy needs in wireless communications are data Privacy, efficiency, verifiability, evaluation privacy etc. The analysis of basic techniques used to accomplish privacy preserving data collection in wireless communications from the subsequent 3 traits are one-way (trapdoor) functions, fully homomorphic encryption (FHE), and secure multiparty computation (SMC) [9].

Abdul Nasir Khana et al. discussed the distinctive security structures proposed for the MCC environment and offload processors, address the issues relating to information security, data breach issues, network security, data confidentiality, data locality, authentication, authorization, data integrity, data access, data segregation, web application security, and different factors and how accomplish a safe MCC condition, security risks should be contemplated and tended accordingly [10].

Cong Wang et al. examined the issue of information security in cloud data storage, proposed a viable and adaptable disseminated scheme including update, delete, and modify with explicit dynamic data support. Authors depend on deletion revising code in the document appropriation planning to give redundancy parity vectors and guarantee the data dependability. Conveyed corroboration of removing coded information by homomophic tokens. This plan accomplishes the integration of storage correctness insurance and data error localization [11].

Randeep Kaur et al. demonstrated a general perspective on security, and assuring customers information is put on the protected mode in the cloud. Authorization of customer turns into an obligatory task because of fact that information must not be stolen by the outsider. This paper discussed about quantities of existing strategies used to give security in the field of cloud computing based on various parameters [12].

Shilpi Singh et al. proposed a plan that not just gives security of client's private information of storing and retrieving over the cloud but also additionally verification of the client to the cloud server utilizing elliptic curve cryptography. ECC is depends upon elliptic curves hypothesis. This paper studied the designing of the key cryptographic systems. Koblitz and miller anticipated the idea of ECC that is used and described for elliptic curve diffie-hellman (ECDH) key exchange, elliptic curve encryption/decryption and A3 algorithm [13].

Mohamed Hamdi et al.- Evasion techniques are used by the attackers to bypass preventive and reactive security mechanisms. They break into three categories-Packet splitting- Duplicate insertion, Payload mutation and Shell code mutation. One key fact that have been noticed based on the security surveys is that application-level attacks are, by far, more bandwidth-efficient than network-level attacks. This is mainly because, at the application level, attackers often use script injection tools rather flooding tools. The most important attacks that can be performed at the application layer include SQL injection and Cross Site Scripting (XSS) attacks, and direct node injection. It is easier to target the application logic or framework of an application than the actual server behind the hardened network perimeter. This paper also provides a information on cloud security techniques, tools, and counter measures[17].

Hui Suo et al. - Aiming at Mobile Terminal Security Anti-malware software (Detection and prevention) e.g.- CloudAV. Mobile Cloud Security (Integrating the current security technologies; Key management and data protection, Authentication and access control Privacy)- Protection to Platform Reliability, Data Encryption and Key Management, Authentication and Access Control, Privacy Protection[18].

Xinyi Huang et al. - Provides the ID-based forward secure ring signature (IDFSRS) scheme, and tuple of probabilistic polynomial-time (PPT) algorithms. Concrete design of forward secure ID based ring signature. No previous ID-based ring signature schemes in the literature have the property of forward security, and feature. It is in ID-based setting. The elimination of the costly certificate verification process makes it scalable and especially suitable for big data analytic environment. The size of a secret key is just one integer. Key update process only requires an exponentiation and need not require pairing in any stage[19].

Syam Kumar Pasupuleti et al. -Key contribution is summariesd as propose an efficient and secure privacy preserving approach; it uses probalistic public key encryption technique to reduce computational overhead on owners. This approach uses ranked keyword search on encrypted data to retrieve file back. Through analysis on security demonstrates that purpose scheme can be proved semantically secure under diff. attacks. ESPPA Approach using probalistic public key encryption and ranked keyword search. In this scheme, the data owner creates an index for file collection then encrypts the both index and files. ESPPA consists of three phases 1) setup phase2) retrieval phase3) integrity verification. Setup phase involved with algorithms key generation, Index creation, Privacy Preserving and encryption algorithms trapdoor generation, Ranked search Index, Data Decryption. The experimental evaluation of proposed ESPPA scheme on real data set (RFC) request for comments included user and server. User acts a data owner, authorized user and server acts as CSP. The performance of this scheme evaluated regarding the efficiency in terms of computation and communication costs. Algorithms use both SSL and MATLAB Libraries and server use c-programming on Linux machine[20].

On the basis of literature survey many security mechanism Identified are given below:-

Authentication Mechanisms:-

- Identity based models

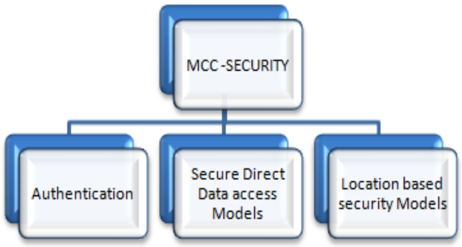

MCC -SECURITY
Authentication
Secure Direct Data access Models
Location based security Models

Fig.2. MCC-Security mechanism

Data Access Mechanisms:-

- Secure Resource allocation methods
- Secure cryptographic schemes
- Web Referral Services methods
- Network security channels
- Privacy preserving approach
- Attribute based encryption

Location Based Services Mechanisms:-

- Outsourced data based model
- Group secure framework
- Separate PPCCP model
- Location based fine grained access control (LFAC)

III. PROPOSED WORK

A. System architecture

The substances in the framework are: user, data owner and the cloud server.

User: this can be explicate as those individuals that can process request to cloud server and can perform search queries over the encoded database.

Data Owner: this can be characterized as a particular individuals who can claims the data while saving his/her privacy.

The system should enable different data owners to encode and upload data to server, through that multiple user can perform search queries. The system should secure the owner's data by using low key management overhead having high versatility. Also, from a user's point of view for per-search operation should be acceptable to have high efficiency.

Cloud Server: this can be defined as that stores the encoded information uploaded by different data owners in a database and process the queries asked for the clients.

B. Proposed model

The CSP is un-trusted, and along these lines the secrecy and integrity of information in the cloud might be in danger. For economic motivating forces and keeping up notoriety, the CSP may hide data loss, or recover stockpiling by disposing of information that have not been or is once in a while accessed. On the other hand, a data owner and approved clients may intrigue and erroneously blame the CSP to get a specific measure of repayment. They may untrustworthily guarantee that information integrity over cloud servers has been damaged [14]. In this research, following are the cryptographic keys to ensure information documents stored on the cloud.

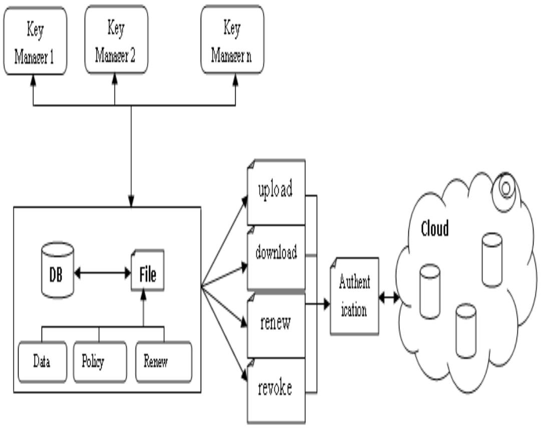


Fig.3. Key management processing module diagram

- Public Key: The Public key is an irregular produced binary key, created and kept up by the Key administrator itself. Especially utilized for encoding and decoding.
- Private Key: It is made by combination of the username, secret key i.e. password and two security question of client's decision. The private key is kept up by customer itself. Utilized for encode/decode the record.
- Access key: It is related with a strategy. Private access key is kept up by the customer. This key is based on quality based encryption. Document can only be read or write using this key.
- Renew key: It is maintained by the customer itself. Everyone has its own renew key. This key is utilized to recreate the approach of every vital document at simple technique.

Threat model

This study considered the "honest-but-curious" public cloud server. In particular, the cloud server should honestly pursue the assigned protocol determination, however it must be "curious" to construe and dissect all information accessible on the server to become familiar with supplementary data. With respect to the private cloud server, expecting it is a trusted third party.

C. Integrity with MRSE

The data owner develops a metadata for each information or file during the setup stage. Once setup is done data owner upload the encoded metadata set to his private cloud server using the keys provided by public cloud server. After that the uploaded information is encrypted by using conventional symmetric encryption algorithm and further uploaded to the public cloud server as read only files. Private cloud server develops the inverted file index and their semantic relationship records. For data retrieval or downloading process the inverted index is distributed to the public cloud server.

Following major properties are included in this process:

- (Data Privacy) defined as client personal information is stored by keys and cryptography techniques.
- (Index Privacy) defined as the search index is secured by keys and they are further secured by metadata or keywords.
- (Trapdoor Privacy) defined as single input trapdoor for a lot of keywords that can't be duplicated with another one even by server itself.
- (Non-Impersonation) defined as security of users profile information.

The proposed research work will have great importance for different sectors like research studies, nation, society, corporate sector and decision analysts for providing secure and save computing environment on personal computers as well as on their mobiles. Through mobile computing everyone involve in faster and secure business enterprise adoption and use of social networks. Cloud computing is changing the way of information technology and mobile end users. Businesses have a range of paths to the mobile cloud, including infrastructure, platforms and applications that are available from cloud providers as online services [21].

Security in main issue: Protecting user privacy and data/application secrecy from adversary is a key to establish and maintain consumers' trust in the mobile platform, especially in MCC. MCC faces a privacy issue when mobile users provide private information such as their current location. This problem becomes even worse if an adversary knows user's important information (Mobile contacts and internet banking, social site etc.) transfer to the other unknown database [22].

IV. METHODOLOGY

In this proposed design privacy preserving access control conspire is introduced. A document can be stored securely in the cloud. There are mainly three kinds of clients (Creator, Writer, Reader) having initial level Registration Process at the web end. The clients give their very personal data to this procedure. The server stores the data in its database. Clients get a secured and private token from the trustee. A trustee can be somebody like the central government monitoring the public tax policies [15]. To show the tax number, the trustees are provided with their token.

There is lots of Key Distribution Channel (KDCs) that issue the keys for encoding/decoding and signing files for Clients.

A. Integrity problems

There are two integrity problems which we wanted to solve

- Search Result Replay Attack

To avoid the search result replay this research proposed following modifications in the MRSE. In the search query each time a secure token is computed and sent to the cloud. The cloud server is to echo back a reply token by doing an operation known between the cloud and user alone. Once the search result arrives at the user end, sanity test on the token is done to ensure there is no replay attack.

Step 1: User when registers to the cloud provider, he is given a secure function HF.

Step 2: User takes the pseudo random number R and applies the function HF on it to get a token TK

TK = HF(R)

Step 3: At the receiver end, cloud server applies the inverse HF on the token to get the R

R1 = HF-1(TK)

Step 4: After this cloud server applies a operation using function OPF, which is known only between the user and the cloud service provider, the reply token RK is computed as

RK = OPF(R1)

RK is then sent in the search query result reply to the user.

Step 5: User applies OPF on the R which he generated to get RK1 as

RK1 = OPF(R)

If the RK1 is equal to RK, the search result is taken a integral from the replay attack.

- Search Result Reordering attack

To solve the search result reordering attack, we propose following changes in the MRSE. When the search result is computed at CS (cloud server), a order based token encrypted with a secure hash function is created and send along with search result to the user end. At the user end, encrypted hash token is created once again from the order of search result and if it matches to the secure hash function in the message, then the ranking is integral, otherwise it can be known some corruption has happened by the attacker.

Step 1: Let the search result be S1, S2, S3.. SN at the cloud server for a search query Q. At the cloud server, a merkel hash function is applied on the search results in following manner

H1= HF(Q,S1);
H2=HF(H1,S2);
H3=HF(H2,S3)
…..
…..
HN=HF(HN-1,SN);

The HN computed is sent in the query result to the cloud server. At the cloud server, the Markel hash is computed again in the same order of search result.

Step 2: If any attacker has altered the search result

Say S3, S5, S1,….SN

Then the computed Markel hash for it

H1= HF(Q,S3);
H2=HF(H1,S5);
H3=HF(H2,S1)
…..
…..
HNN=HF(HN-1,SN);

The HNN computed will not match with the HNN arrived in the message, from this user can be sure the search result has be reordered and integrity is failed.

When a user put a request to process a query ˆQ from LTA, then the LTA verifies about the attribute set W based that query ˆQ, for that particular user. This implies to one kind of approach to maintain the database of attribute values for all users in the LTA's domain. However on the basis of a second approach, the LTA can issue a lot of accreditations confirming the client's attribute values, and checks those credentials upon a demand for ability for example on-demand checking criteria. So as to signify its approval on ability, a LTA can issue an identity-based signature with respect to every capacity it produced/designated. The server needs to check that a got capacity has a substantial signature from an enlisted LTA before performing search process for a client.

B. Encrypted Index Generation and query privacy

The basic objective is to maintain the cloud server from taking any additional data from the encoded records, indexes, and the clients' profiles, except from the query results. Security of Query Index alludes to classification of the file, while query protection ensures clients' profiles. Like example shown in above figure 4 classification based on attribute like age, sex, region and provider leads to further query processing by users.

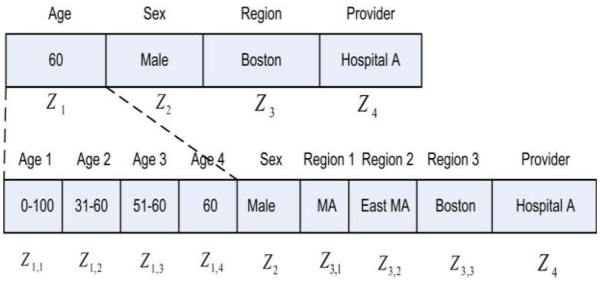


Fig.4. Index conversion among hierarchical fields – for example Age and region

C. Process of MRSE-Multi-dimensional Keyword Search

The proposed framework should support multi-dimensional keyword search for that it must need to implement with conjunctions among various dimensions where for every dimension there can be multiple keywords. The normal process takes *t* time in searching for single encoded file under various n values. That states the proposed design is speedier in search process than basic encryption methods because it only takes n + 3 values of pairing operations.

V. DESIGN AND IMPLEMENTATION

A. Design issues

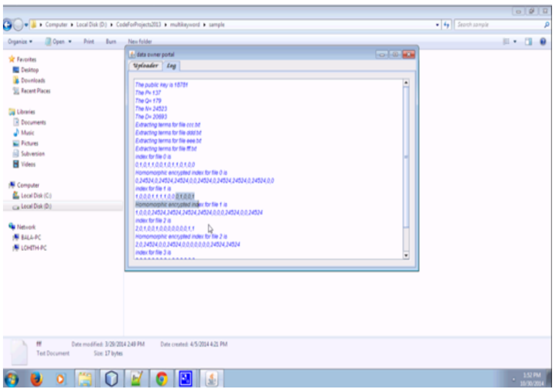
- This project focuses on to the solution in the cloud computing paragon of multiple keywords ranked search over encrypted cloud data (MRSE) while protecting its privacy.
  - To use the groups of issued keywords to verify cloud data storage and access for ranking system

The major modules of proposed design are relied upon giving the security and privacy ensures as follows:

- MRSE: This module is used to configure the searching criteria with multi-keyword query and deliver the required result for current data retrieval with ranking, instead of homogeneous results.
- Privacy-Preserving issues: This module is used to control the data downloading and uploading from the dataset and the record for the cloud server, fulfilling its fundamental protection prerequisites.
- Efficiency: Searching and ranking process ought to guarantee security and furthermore low correspondence and calculation overhead.

B. B. Implementation steps

The proposed work is implemented in NetBeansIDE version 8.0 beta. Following steps are to be taken to implement the proposed work:

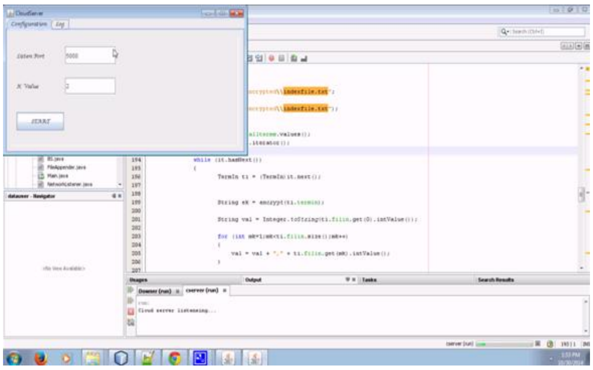


Computer ▸ Local Disk (D:) ▸ CodeForProjects2013 ▸ multikeyword ▸ sample
data owner portal
Uploader
Log

```
The public key is 18781
The P= 137
The Q= 179
The N= 24523
The D= 20693
Extracting terms for file ccc.txt
Extracting terms for file ddd.txt
Extracting terms for file eee.txt
Extracting terms for file fff.txt
index for file 0 is
0,1,0,1,1,0,0,1,0,1,1,0,1,0,0
Homomorphic encrypted index for file 0 is
0,24524,0,24524,24524,0,0,24524,0,24524,24524,0,24524,0,0
index for file 1 is
1,0,0,0,1,1,1,1,0,0,0,1,0,0,1
Homomorphic encrypted index for file 1 is
1,0,0,0,24524,24524,24524,24524,0,0,0,24524,0,0,24524
index for file 2 is
2,0,1,0,0,1,0,0,0,0,0,0,0,1,1
Homomorphic encrypted index for file 2 is
2,0,24524,0,0,24524,0,0,0,0,0,0,0,24524,24524
index for file 3 is
```

Date modified: 3/29/2014 2:49 PM Date created: 4/5/2014 4:21 PM
Size: 17 bytes

Fig.5.Shows the process of profile data generation at current time for different users

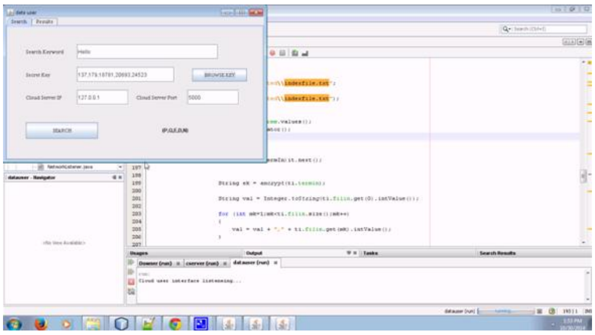


Listen Port: 5000
K Value: 2

```
while (it.hasNext())
{
        TermIn ti = (TermIn)it.next();

        String eK = encrypt(ti.termin);

        String val = Integer.toString(ti.filin.get(0).intValue());

        for (int mk=1;mk<ti.filin.size();mk++)
        {
            val = val + "," + ti.filin.get(mk).intValue();
        }
```

```
run:
Cloud server listening...
```

Fig.6. Shows the process of index generation i.e. Log file calling

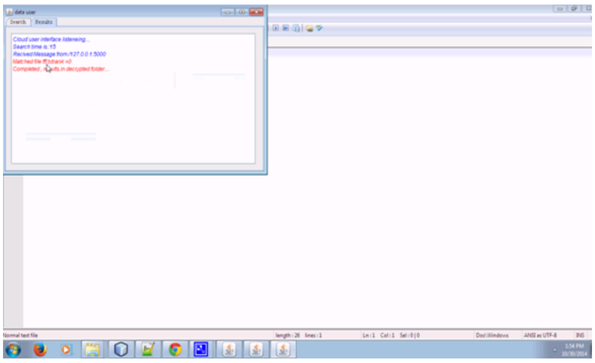


Search Keyword: Hello
Secret Key: 137,179,18781,20693,24523
Cloud Server IP: 127.0.0.1
Cloud Server Port: 5000

```
        String eK = encrypt(ti.termin);

        String val = Integer.toString(ti.filin.get(0).intValue());

        for (int mk=1;mk<ti.filin.size();mk++)
        {
            val = val + "," + ti.filin.get(mk).intValue();
        }
```

```
run:
Cloud user interface listeneing...
```

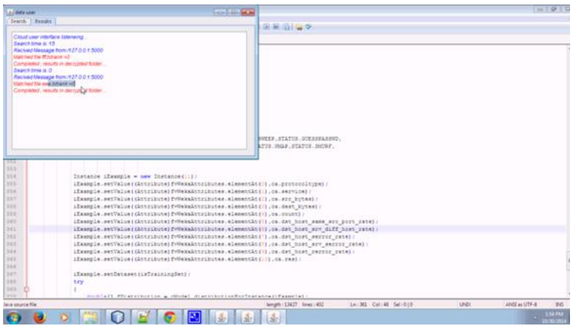
Fig.7. Shows the process of key generation for a particular user's profile



```
Cloud user interface listeneing...
Search time is :15
Recived Message from /127.0.0.1:5000
Matched file fff.txtrank =0
Completed , results in decrypted folder....
```

Fig.8. Shows the process profile search encryption

Figure 5 shows the process of profile data generation at current time for different users. This also generates the key pair for both public and private kind of users. After that by using these keys the file data encrypted into cipher text i.e. encryption shown by figure 6. The keywords generated in figure 6 are indexed for key security in figure 7 i.e. index generation i.e. Log file calling. The consequence of user queries like searching or ranking for encryption process are shown by figure 8.



```
Cloud user interface listeneing...
Search time is :15
Recived Message from /127.0.0.1:5000
Matched file fff.txtrank =0
Completed , results in decrypted folder....
Search time is :0
Recived Message from /127.0.0.1:5000
Matched file eee.txtrank =0
Completed , results in decrypted folder....
```

```
Instance iExample = new Instance([illegible]);
iExample.setValue((Attribute)fvWekaAttributes.elementAt([illegible]),ca.protocoltype);
iExample.setValue((Attribute)fvWekaAttributes.elementAt([illegible]),ca.service);
iExample.setValue((Attribute)fvWekaAttributes.elementAt([illegible]),ca.src_bytes);
iExample.setValue((Attribute)fvWekaAttributes.elementAt([illegible]),ca.dst_bytes);
iExample.setValue((Attribute)fvWekaAttributes.elementAt([illegible]),ca.count);
iExample.setValue((Attribute)fvWekaAttributes.elementAt([illegible]),ca.dst_host_same_src_port_rate);
iExample.setValue((Attribute)fvWekaAttributes.elementAt([illegible]),ca.dst_host_srv_diff_host_rate);
iExample.setValue((Attribute)fvWekaAttributes.elementAt([illegible]),ca.dst_host_serror_rate);
iExample.setValue((Attribute)fvWekaAttributes.elementAt([illegible]),ca.dst_host_srv_rerror_rate);
iExample.setValue((Attribute)fvWekaAttributes.elementAt([illegible]),ca.dst_host_rerror_rate);
iExample.setValue((Attribute)fvWekaAttributes.elementAt([illegible]),ca.res);

iExample.setDataset(isTrainingSet);
try
```

Fig.9. Shows the process of data decryption for profile search by users

These above ramification of user doubt like searching or ranking for decryption process are shown by figure 9. Regardless, the way of implementation it to encode the cloud data in searching system became a very troublesome task because of its rigid privacy and security issues, comprising things like the data assurance, the record insurance, the keyword insurance, and various others.

VI. RESULT AND ANALYSIS

We measured the communication overhead due to above two changes in MRSE by varying the number of searches and plotted the result below

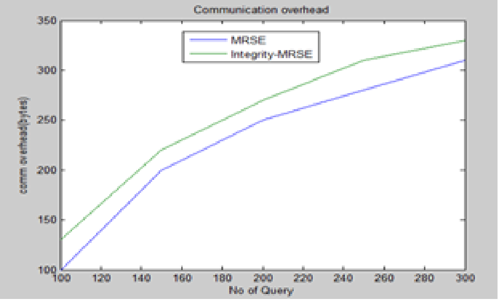


Fig.10. Comparison graph of communication overhead between MRSE and Integrity with MRSE

From the result, we see that the communication overhead in terms of bytes is very less for the added benefit of integrity checking. We also measured the search time against MRSE for different number of queries and plotted the result below.

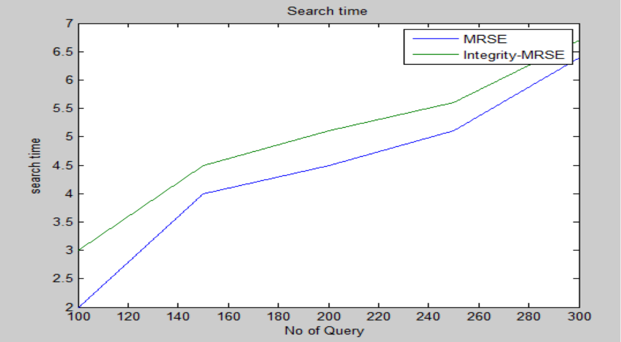


Fig.11. Comparison graph of search time between MRSE and Integrity with MRSE

From the results, we see that there is only a small increment in search time and it is linear increasing. From the result shown in figure 10, we see that the communication overhead in terms of bytes is very less for the added benefit of integrity checking. We also measured the search time against MRSE for different number of queries and plotted the result. From the results shown in figure 11, we see that there is only a small increment in search time and it is linear increasing.
Comparison results in proposed and existing approach:

Table- I: Comparison results in proposed and existing approach for communication overhead

Parameters	MRSE	Integrity+ MRSE
Average delay	0.6155203	0.1934061
Average PDR	0.4293650	0.555238
NRL	7	9
Throughput	123	91

From table 1 result can be interpreted as communication overhead is increased for proposed work as compared to existing work as number of queries increased.
Comparison results in proposed and existing approach:

Table– II: Comparison results in proposed and existing approach for Search time

Parameters	MRSE	Integrity+ MRSE
Average delay	0.6155203	0.1241600
Average PDR	0.4963652	0.5942358
NRL	8	11
Throughput	137	96

From table 2 results can be interpreted as search time is increased for proposed work as compared to existing work as number of queries increased.

VII. CONCLUSION

In this research, we have discussed the integrity issues in the MRSE scheme and proposed two enhancements in MRSE scheme to solve the integrity attacks. Through performance results we also proved that the cost of communication overhead and search time increment is less for the added benefit of integrity check. Currently we are only finding if the search result is integral , in future we will focus on search result order recovery without initiating a new search in case of integral failure.

REFERENCES

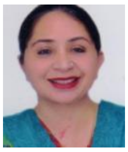
1. L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner. "A break in the clouds: towards a cloud definition", ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 50–55, (2009).
2. S. Kamara and K. Lauter. "Cryptographic cloud storage, in RLCPS", LNCS. Springer, Heidelberg (January 2010).
3. A. Singhal. "Modern information retrieval: A brief overview, IEEE Data Engineering Bulletin", vol. 24, no. 4, pp. 35–43, (2001).
4. H. Witten, A. Moffat, and T. C. Bell. "Managing gigabytes: Compressing and indexing documents and images", Morgan Kaufmann Publishing, San Francisco, (May 1999).
5. D. Song, D. Wagner, and A. Perrig. "Practical techniques for searches on encrypted data", in Proc. of S&P, (2000).
6. E.-J. Goh. "Secure indexes, Cryptology ePrint Archiv\e", (2003), http:// eprint.iacr.org/2003/216.
7. Y.-C. Chang and M. Mitzenmacher. "Privacy preserving keyword searches on remote encrypted data", in Proc. of ACNS, (2005).
8. R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky. "Searchable symmetric encryption: improved definitions and efficient constructions", in Proc. of ACM CCS, (2006).
9. Jun zhou, Zhenfu, XiaoleiI Dong, Xiodong Lin. "Security and Privacy In Cloud-Assisted Wireless wearable Communication, Wireless Communications", IEEE Journals & Magazines, Volume: 22, Issue: 2, Pages: 136 -144, (2015)
10. Abdul Nasir Khana, M.L. Mat Kiah a, Samee U. Khanb, Sajjad A. Madani., "Towards secure mobile cloud computing: A survey", Elsevier Journal, homepage:www.elsevier.com/locate/fgcs, Future Generation Computer Systems 29, Pages: 1278-1299 (2013).
11. Cong Wang, Qian Wang, and Kui Ren. "Ensuring Data Storage Security in Cloud Computing, Quality of Service", 17th International Workshop on Quality of services(IWQoS), Charleston, South Carolina, USA, Pages: 1 - 9, IEEE Conference Publications, 13-15 (July 2009).
12. Randeep Kaur,Supriya Kinger. Analysis of Security Algorithms in Cloud Computing, International Journal of Application or Innovation in Engineering & Management, Volume 3, Issue 3, Pages: 171-176, (March 2014).
13. Shilpi Singh, Vinod Kumar. "Secured User's Authentication and private Data Storage- Access Scheme in Cloud Computing Using Elliptic Curve Cryptography", New Delhi, India, Pages: 791 – 795, IEEE Conference Publications, (11th to 13th March, 2015).

L. Ballard, S. Kamara, and F. Monrose. Achieving efficient conjunctive keyword searches over encrypted data, in Proc. of ICICS, (2005).

14. P. Golle, J. Staddon, and B. Waters. “Secure conjunctive keyword search over encrypted data”, in Proc. of ACNS, pp. 31–45 (2004).
15. W. W. Cohen. Enron email dataset, http://www.cs.cmu.edu/∼enron , and last accessed 2018/7/1.
16. Mohamed Hamdi, “Security of cloud computing, storage, and networking” International Conference on Collaboration Technologies and Systems (CTS), Denver, CO, USA, paper published in IEEEXplore, DOI: 10.1109/CTS.2012.6261019, May 2012.
17. Hui Suo, Zhuohua Liu, Jiafu Wan Keliang Zhou, “Security and Privacy in Mobile Cloud Computing”, 9th International Wireless Communications and Mobile Computing Conference (IWCMC), University of Cagliari, Italy ,Pages: 655 - 659, IEEE Conference Publications, 1-5 July 2013
18. Xinyi Huang, Joseph K. Liu, Shaohua Tang, “Cost-Effective Authentic and Anonymous Data Sharing with Forward Security”, IEEE Transactions on Computers, Vol. 64, no. 4, Pages: 971 - 983 , April 2015.
19. Syam Kumar Pasupuleti, Subramanian Ramalingam, Rajkumar Buyya, “An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing” ,Journal of Network and Computer Applications, Pp:1-11 ,2016.
20. M. Bellare, A. Boldyreva, and A. ONeill, “Deterministic and efficiently searchable encryption,” in Proc. of CRYPTO, 2007.
21. 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,” J. Cryptol., vol. 21, no. 3, pp. 350–391, 2008.

AUTHORS PROFILE

Vishal, is an research scholar at I.K.G. Punjab Technical University,Kapurthala (Punjab) and Assistant Professor at the Department of computer Science and Engineering, Jai Parkash Mukand Lal Innovative Engineering and Technology Institute, Radaur. He is Bachelors of Technology in Computer Engineering field from Kurukshetra University and Master of Technology in Software Engineering from Kurukshetra University. His research interest is in the area of security in cloud computing and mobile devices, ANN, Adhoc network etc. He is an Active researcher who has supervised many B.Tech. Projects, M.Tech / Research Projects, guided/ guiding M.Tech. Dissertations / Thesis. He has more than 15 years teaching experience to teach B.Tech, M.Tech students. At Punjab Technical University he is doing research work on field of security in mobile cloud computing. Author has published around 20 research papers in International Journals and conferences. Mr.Vishal is member of international & professional bodies CSI, IAASSE, IAENG, IAOIP, and ICAICR.



Dr.Bikram Pal Kaur is Professor in the Department. of Computer Science & Engineering and was also HOD of Deptt. Of Computer Application in Chandigarh Engineering College, Landran, Mohali.She holds the degrees of B.Tech., M.Tech, M.Phil., PhD in the Computer Engineering from Punjabi University, Patiala. She has more than 22 years of teaching experience and served many academic institutions. She is an Active Researcher who has supervised many B.Tech. Projects, M.Tech / Research Projects, guided/ guiding M.Tech. Dissertations / Thesis and also guiding PhD to seven scholars . She has contributed more than 28 articles in various national/ international conferences and 43 papers in research Journals. Her areas of interest are Information System, ERP, Parallel Computing, AI, Fuzzy . Dr.Kaur is member of many professional societies such as ICGST, IAENG.

Dr.Surender completed his M.Tech degree in Computer Science and Engineering from Ch. Devi Lal University Sirsa(Hry) in 2006, M.Phil in Computer Science from Alagappa University, Karaikuri (T.N) in 2008, Ph.D in Computer Science and Application from Kurukshetra University, Kurukshetra in 2011 and also qualified UGC-NET in Computer Science and Applications in Oct. 2013. He has more than 10 years teaching experience to teach B.Tech, M.Tech., BCA and MCA Classes and recently working as an Assistant Professor, Department of Computer Science, GTB College, Bhawanigarh (Sangrur), Punjab, India. He has published over 50 publications in different International Journals and Conferences of repute. His research interests lies in Fault Tolerance in Mobile Distributed Systems, Adhoc N/W, Data Mining, Cloud Computing, System Security and Cryptography.