Improving Efficiency of TRSE Scheme by Employing Public Key Compression Technique for Fully Homomorhphic Encryption over the Integers

Mr. Sunil A. Kumbhar[1], Mr. Chetan J. Awati[2]

[1] *Department of Technology, Shivaji University, Kolhapur, India*
[2] *Department of Technology, Shivaji University, Kolhapur, India*

Abstract

Day by day popularity of cloud computing technology is increasing. It provides various services & reduces the IT cost. Data outsourcing is an important feature provided by cloud. But to store a data on remote location may leak data privacy. Two Round Searchable Encryption scheme has been proposed that supports top k multi keyword retrieval over encrypted cloud data. In TRSE, vector space model provides search accuracy & homomorhphic encryption provides the data privacy. Ranking is done at user side and score calculation is done at server side. TRSE scheme achieve data privacy and efficiency. However, that efficiency is achieved with the cost of large public key size. So the new homomorhphic encryption is needed, that will reduce the size of key thus improve the efficiency. So the proposed scheme focused on improving the efficiency of existing TRSE scheme by using Public key compression technique for fully homomorhphic encryption scheme over the integers & also comparative efficiency analysis of both existing & proposed TRSE is carried out in terms of communication overhead.
Keywords: *cloud, data privacy, ranking, fully homomorhphic encryption, vector space model, relevance scoring.*

1. Introduction

As Cloud computing can significantly reduce the cost and complexity of owning and operating computers and networks of an organization by providing a various cloud services like SAAS, PASS and IAAS. So organization are now thinking to move on to the cloud. However, Privacy and security are the major concerns in cloud. Organization (data owner) or data users (Authorized user of organization) are expecting not only a high data security but also efficiency from cloud service provider. Sometimes data owner wants to perform operation on data (addition multiplication etc) for that, cloud server needs a raw data. But cloud server itself is not a trusted one. So, data owner expects that no one including cloud service provider leak data privacy. To achieve data privacy, data owner encrypt the data before storing it to the remote location. Data owner frequently perform the searching operation on data. This data should be retrieved with more relevant result to the search query. But searching on encrypted data makes it a difficult task. To tackle this problem, SSE scheme [1] has been proposed which supports Boolean keyword search. Boolean Keyword search often result in either too few (=0) or too many (1000s) results without considering a relevance between keyword in query and files.

To provide a high data privacy, Homomorhphic encryption technique has been emerged which provides the operations on encrypted data. User request to perform an operation on the data files. Server computes that operation on encrypted data without knowing the secret key. So server does not learn about the plain text. Hence data privacy is preserved. Homomorhphic encryption may be additive or multiplicative. Additive homomorhphic encryption performs addition operation and Multiplicative homomorhphic encryption performs multiplication operation on encrypted cloud data [2].

To achieve high data privacy and balance between efficiency & security, Two Round Searchable encryption Scheme [3] has been proposed which provides a data privacy through homomorhphic encryption and use vector space model to provide search accuracy.

So far discussion about existing scheme, we observed two issues: 1. this scheme has used

"Modified Fully Homomorhphic Encryption over the integer" to achieve the efficiency; but public key size is very large and this leads to the greater communication overhead when data user encrypt the key (ex science) for searching a data on cloud. 2. Frequent operation on to the cloud data also makes very challenging task to maintain the efficiency of TRSE scheme. On the basis of experimental result of existing scheme, if the file set contains 500 files and 1000 distinct keyword then size of one encrypted result takes 78.125ms to reach the user. So the efficiency of existing scheme can be further improved by reducing the size of the public key and thus we can reduce communication overhead.

In proposed scheme we focused to reduce the size of the key by using public key compression technique for fully homomorhphic encryption scheme over the integers [4]

2. Problem Statement

Data user or data owner encrypts a keyword (ex. Computer) using public key having larger public key size for searching files on the cloud. If encrypted data is large, there is large communication overhead between cloud server and data user. This reduces the efficiency of existing TRSE scheme.

The main objectives are as follows:

1. To implement Existing TRSE scheme using "modified fully homomorhphic encryption scheme over the integer which fulfills the secure multi keyword top-k retrieval over encrypted cloud data".

2. To Design & implement Proposed TRSE scheme using "Public key compression technique for fully homomorhphic encryption over the integers which fulfills the secure multi keyword top-k retrieval over encrypted cloud data".

3. Comparative efficiency analysis of existing & proposed TRSE scheme.

3. Literature Review

As demand of high data security, some researchers have focused to develop an Encryption technique which enables to perform operations on encrypted cloud data.

Gentry[5]described somewhat homomorhphic encryption which supports limited number of addition and multiplication operations on encrypted cloud data because there is a noise factor in every cipher text and any homomorhphic operation applied on to cipher text increases the noise in the resulting cipher text. Once this noise crosses a certain bound, the resulting cipher text doesn't decrypt any more.

So, to support unlimited addition and multiplication operation on cipher text, noise should be reduced in the cipher text. This process is called a cipher text refresh. Gentry's provided solution as bootstrapping technique in which cipher text is refreshed by homomorphically evaluating the decryption circuit on the cipher text bits, using an encryption of the secret-key bits. This gives us encryption of plain text bit rather than plaintext bit. If the depth of decryption circuit will be small enough, then noise in the cipher text can be smaller than in the original cipher text, hence a cipher text refresh. This scheme is based on the ideal lattice.

Van Dijk, Gentry, Halevi and Vaikuntanathan's described (DGHV) [6] fully homomorphic encryption over the integers. This scheme is conceptually simpler than Gentry's scheme, because it operates on integers instead of ideal lattice. This scheme is achieved simplicity with cost of large public key size O (λ^{10}).

Jean-S_ebastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi [7] described fully homomorhphic encryption over the integers with shorter public key size. This scheme is reduced the public key size of DGHV scheme [6] from O (λ^{10}) to O (λ^{7}).

An effectively search over encrypted cloud data some authors have focused on single keyword retrieval.

Song, D. Wagner, and A. Perrig described the cryptographic schemes in order to tackle searching problem on encrypted data and provided various security proofs for the resulting crypto systems [8]. Scheme supports controlled hidden search and query isolation. Scheme is provably secure that server cannot learn anything about the plaintext when only given the cipher text. Scheme focuses on security definitions and encryption efficiency but these work support only Boolean keyword retrieval without ranking

Swami Nathan & et al. have designed confidentiality preserving top-k retrieval framework which provides confidentiality to the document and query through OPE (Order Preserving Encryption) [9].Framework provided confidentiality to documents and indices (collection of files) through the order preserving encryption scheme. So data center as well as intruder could not learn anything about the plain text. Relevance scoring methods are used to assign score to the query and document pair. Design provided efficient and accurate search capabilities to securely rank-order documents in response to a query.

Later on Zerr & et al. have proposed a Zerber+R ranking model to preserve a privacy of shared document exchange among the enterprises. Model preserves privacy of shared documents while retrieving a top k documents from an outsourced inverted index. [10].They proposed a relevance score transformation function to calculate relevance scores of different terms. Due to which they have improved the security of the indexed data. This model focused on only on single term query.

Cong Wang & et al. have focused on security and efficiency of ranked keyword search over outsourced encrypted cloud data based on Searchable Symmetric Encryption [11]. Problem addressed with this scheme are ranking done at the user side so communication overhead is increased. To reduce this problem, one-to-many Order Preserving Mapping has designed to further improve the efficiency but security guarantee and retrieval accuracy are slightly weakened.

An effectively search over encrypted cloud data some authors have focused on multi keyword retrieval are given below.

P. Golle & et al. have proposed several schemes supporting Boolean multi keyword retrieval [12]. They have designed two protocols for conjunctive search for which it is provably hard for the server to distinguish between the encrypted keywords of documents of its own choosing. Protocols support secure conjunctive search with small capabilities. Drawback of the protocols are partially solves the problem of secure Boolean search on encrypted data.

N. Cao, C. Wang & et al. [13] made the first attempt to define and solve the problem of top-k multi keyword retrieval over encrypted cloud data. They have measured a similarity of coordinate matching. That is matching of query keywords with the document data. To evaluate the similarity measure inner product similarity is used. Author proposed basic MRSE scheme by using secure inner product computation and provided improved MRSE scheme in two different threat models to satisfy a strict privacy requirement.

N.P. Smart and F. Vercauteren have designed a fully homomorhphic encryption scheme with Relatively Small Key and Ciphertext Sizes as compared with Gentry's original scheme[14].

4. Proposed System

4.1 System model

Proposed TRSE scheme can be shown in fig 1.

In proposed scheme, three different parties are involved.

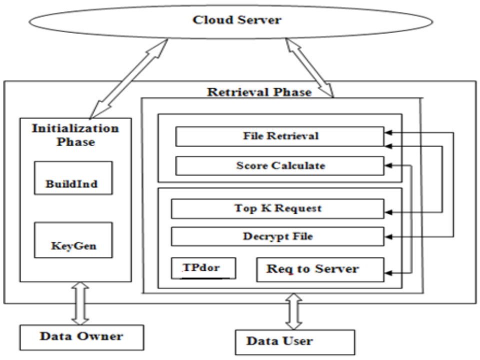

Fig 1.Proposed Scheme

***4.1.1 Cloud Server*:** Cloud server is responsible to store large amount data files of a data owner in an encrypted form. Cloud server is not fully trust worthy for storage.

4.1.2. Data Owner: Data Owner responsible to store encrypted collection of files and encrypted searchable index on cloud.

4.1.3. Data user: Authorized data user first generates the query in the encrypted form & submits to cloud server. Cloud server returns the relevant files with score. Data user selects top k highest scoring files and request to cloud server. Cloud server returns the result.

4.2 Relevance Scoring

Score is the base on which ranking is done. Various techniques are available from information retrieval community through which we can calculate a score & measure the relevance between query and document pair. Score measures how well document & query match.

In proposed scheme following technique will be used given below:

$$idf_t = \log(N/df_t)$$

Where idf_t =Inverse Document Frequency of a term t, N-No of files in the collection df_t=Number of a documents which contains a term t. Inverse document frequency is taken because term which occurs less frequently is more informative. Weight of term t in the document f is given by:

$$tf\text{-}idf_{tf} = t_{t,f} * idf_t \quad (1)$$

Where t *f* is the no of times a term occur in a file f

4.3 Vector Space Model

Vector space model [15] helps us to assign score to a file on the basis of multi keyword. In this model every file is treated as a vector. Each Dimension of the vector depends on the separate term for example If the terms are to be considered as words then dimension is nothing but number of words in the data set. If a term found in a document, then value of that term in document is assumed a non zero otherwise it is considered zero. This is a simple model which allows ranking documents according to their possible relevance. Model helps to continuously capture relevance between files and queries. Model is based on the term weight .It doesn't support binary representation. Score of file f on query q is given by the inner product of the two vectors. First vector for file & the vector for query containing multiple keywords.

$$score_{f,q} = v_f * q \quad (2)$$

Proposed scheme contains initialization and retrieval phase.

4.4. Initialization phase

In this phase data owner and cloud server are involved. Data owner generate the secret SK and public key PK set for homomorhphic encryption by using KegGen and Extract p keywords from collection of files. Then Calculate Term Frequency (TF) & Inverse Document Frequency (IDF) from collection of files. For each file Data owner generate dimensional vector & from collection of vector searchable index is generated. Searchable index is encrypted by using reduced public key & file collection is encrypted with other encryption technique. Encrypted index & Encrypted collection of files are stored on to the cloud.

4.5 Retrieval phase

In this phase data user and cloud server are involved. Data user generates the secure trapdoor using reduced public key. Generates a keyword set as request for searching. From keyword set query vector is generated. Generate secure trapdoor & send it to the cloud server. Server calculates the inner product & returns the vector containing identity of file and product. Data user decrypts result & select top k scoring identifiers. These top k elements identifiers are transmitted to cloud server. Then cloud server transmits the top k required encrypted files.

4.6 Public key compression technique for fully homomorhphic encryption over the integers

Public key compression technique is applied on the DGHV scheme [6] and key size is reduced from O (λ^7) to O (λ^5).

For security parameter λ following parameters are considered:

γ is the bit length of xi's

η Bit length of secret key p

$\mathcal{P}$ bit length of noise ri's

t number of the xi's in public key

$\mathcal{P}$' secondary noise parameter used for encryption

Step 1: KeyGen (1^λ): Generate a random prime integer p as a secret key of size η bits. Random odd integer q0 from interval [0,2^γ/p) and x0, =q0 * p. PRNG (pseudo-random number generator f) with random seed se is initialized. This PRNG generate the set of integer Xi from interval [0, 2^γ) for 1≤i≤t. Compute the small correction to the δi to the Xi's such that xi=Xi-δi is small modulo p, store only the small correction δi in the public key, instead of full xi.

Step2: Encrypt (pk, m ∈{0, 1}: Choose the random integer vector bi from interval $[0, 2^{\alpha})^t$ for1≤i≤t. and random integer r from interval $\left(-2^{\mathcal{P}'}, 2^{\mathcal{P}'}\right)$ and output cipher text as c=$m + 2r + 2\sum_{i=1}^{t} b_i * xi \ mod x0$ where m denotes the plain text of the integer, c denotes the cipher text of the integer

Evaluate and decrypt steps are same as in Original DGHV Scheme [6] but cipher texts are reduced modulo x0. In the above scheme to reduce the size of

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 1, January - 2014

public key apart from generating x_i as $x_i=q_i*p+2r_i$ in DGHV [6] Scheme, in proposed scheme first generates same size of pseudo-random Xi and small correction $\delta_{i,}$ is computed such that $x_i=Xi-\delta_{i,}$ is small modulo p. Then only small corrections in the public key are stored with the seed of the Pseudo Random Generator So size of public key is definitely reduced as compared to the Homomorhphic encryption scheme used in the existing TRSE scheme.

5. Implementation

Experiment is conducted by using file set of National Research Awards Abstracts 1990-2003 [16] on two machines. From NSF file set only 18 files are used from 1990 Abstract file collection & 30000 keywords are used from word.txt file. First machine is act as Data owner & Data user with Pentium dual core CPU running at 2.50 GHz, 0.99 GB of RAM & Second machine act as cloud server with configuration Intel core i3 CPU running at 2.3 GHz, 2 GB RAM. Ubuntu 10.04(eucalyptus cloud ubuntu 10.04) is deployed on the second machine.

Simple Web application is created containing login for data owner as well as admin. After login, data owner is validated by admin. Validation part is implemented in Java Script. Servlet & JSP are used as server side technology. MySQL data base is used to store details of registration and login. Net Beans IDE tool is used for development. Website is deployed on to the cloud. Data owner perform keygen & BuildIndex task by accessing TRSE scheme application developed in java language. Encrypted Index is stored on the cloud by data owner. Following are some screenshots given below.

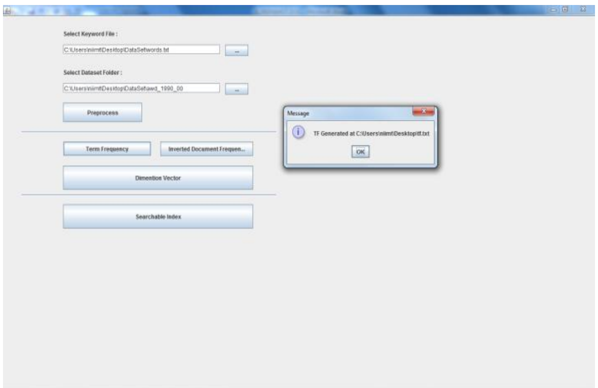

Figure 2. TF calculation

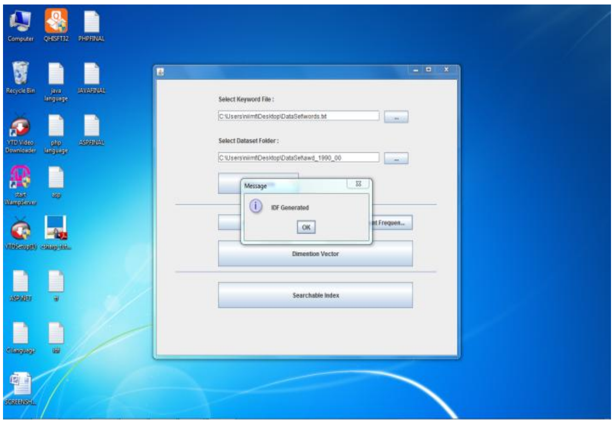

Figure 3. IDF Calculation

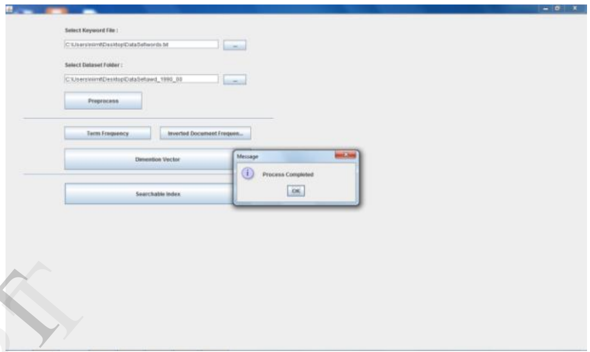

Figure 4. Searchable Index Generation

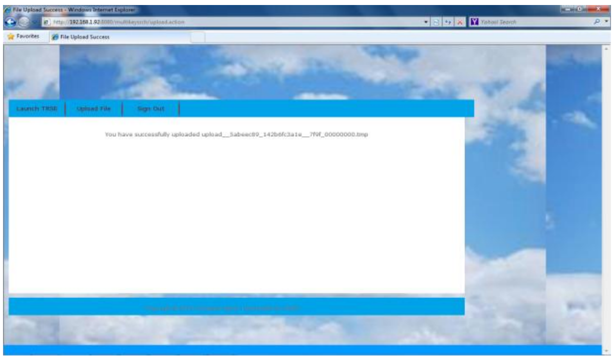

Figure 5.Encrypted Index Upload to the cloud Server

6. Conclusion & Future Work

In this paper we focus to overcome a first issue & improve the efficiency of TRSE scheme using public key compression technique for fully homomorhphic encryption over the integers .Fully homomorhphic encryption provides a data privacy & vector space model achieve search accuracy. Scheme supports secure multi keyword top k retrieval of documents over encrypted cloud data.

Initialization phase is implemented by using Modified fully homomorhphic encryption scheme for existing TRSE scheme. In this phase keygen and buildindex algorithms have been implemented. Next retrieval phase will be implemented by using modified fully homomorhphic encryption scheme for existing TRSE scheme. Communication overhead between data user & cloud server will be measured. Both initialization phase & retrieval phase will be implemented by using public key compression technique for fully homomorhphic encryption over the integers in proposed TRSE scheme. Experimental result will be taken and comparative efficiency of both schemes will be analyzed.

In future, Efficiency of Proposed TRSE Scheme can further improved by overcoming second issue.

REFERENCE

[1] R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption Improved Definitions and Efficient Constructions," *Proc. ACM 13th Conf. Computer and Comm. Security (CCS)*, 2006

[2] Maha TEBAA, Saïd EL HAJJI, Abdellatif EL GHAZI "Homomorphic Encryption Applied to the Cloud Computing Security" *Proceedings of the World Congress on Engineering 2012 Vol I*

[3] jiadi Yu, Peng Lu, Yanmin Zhu, GuangtaoXue,IEEE Computer Society, and Minglu Li "Toward Secure Multikeywod Top-k Retrieval over Encrypted Cloud Data "*IEEE Transaction on Dependable and Secure Computing* , VOL. 10, NO. 4, JULY/AUGUST 2013

[4] Public Key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integers by Jean- S_ebastien Coron, David Naccache, and Mehdi Tibouchi

[5] Gentry, "Fully Homomorphic Encryption Using Ideal Lattices,"*Proc. 41st Ann. ACM Symp. Theory of computing (STOC)*, pp. 169- 178, 2009.

[6] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," *Proc. 29th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques*,pp. 24-43, 2010

[7] J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully Homomorphic Encryption over the Integers with Shorter Public Keys," CRYPTO '11: *Proc. 31st Ann. Conf. Advances in Cryptology*, 2011.

[8] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," *Proc. IEEE Symp. Security and Privacy*, 2000

[9] A. Swami Nathan, Y. Mao, G.-M. Su, H. Gou, A.L. Varna, S. He, M. Wu, and D.W. Oard, "Confidentiality Preserving Rank-Ordered Search," *Proc. Workshop Storage Security and Survivability*, 2007.

[10] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Top-k Retrieval from a Confidential Index," *Proc. 12th Int'l Conf. Extending Database Technology: Advances in Database Technology (EDBT)*, 2009.

[11] "Cong Wang, Ning Cao, KuiRen, Wenjing Lou "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data", *IEEE Transaction On Parallel and Distributed Systems*,Vol. 23 No.8 ,2012

[12] P. Golle, J. Staddon, and B. Waters, "Secure Conjunctive Keyword Search over Encrypted Data," *Proc. Second Int'l Conf. Applied Cryptography and Network Security (ACNS)*, pp. 31-45, 2004.

[13] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multikeyword Ranked Search over Encrypted Cloud Data," *Proc. IEEE INFOCOM*, 2011.

[14] N. Smart and F. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Cipher text", *Public Key Cryptography (PKC)*, 2010

[15] D. Dubin, "The Most Influential Paper Gerard Salton Never Wrote," Library Trends, vol. 52, no. 4, pp. 748-764, 2004.

[16] "NSF Research Awards Abstracts 1990-2003," http://kdd.ics.uci.edu/databases/nsfabs/nsfawards.html,2013