International Journal of Security and Its Applications
Vol.9, No.11 (2015), pp.323-336
http://dx.doi.org/10.14257/ijsia.2015.9.11.30

Cryptography Based Dynamic Multi-Keyword Ranked Search Using ECC/B+TRE

Prasanna B T, C B Akki

EPCET, Bengaluru, SJBIT, Bengaluru
prasi.bt@gmail.com,
akki.channappa@gmail.com

Abstract

Today, Cloud computing is becoming a household technology. In cloud, a shared pool of computing resources can be accessed on demand through internet and web applications. Since outsourced data is in third party premises i.e. cloud, accountability of user data is paramount. To preserve privacy and security of user data in cloud, many cryptographic techniques have been proposed by many scientists. One among them is searchable encryption. Searchable encryption allows secure search over encrypted data. In our system, a noble approach has been made using the Elliptic Curve Cryptography (ECC), a cryptographic techniques to reduce the overall computation overhead. Dynamic B+ tree data structure is used to perform multi-keyword search over the encrypted data. To retrieve appropriate data files, ranking will be done based on relevance score. Finally, we compare the effectiveness and efficiency of our proposed scheme with our previous work on CRSA/B+ tree through extensive experimental evaluation using Microsoft azure platform.

Keywords*: Cloud computing, Searchable Encryption, Elliptic Curve Cryptography, Microsoft Azure, B+ Tree*

1. Introduction

Cloud computing, uses web technologies and remote servers to store and maintain the user data. In [1], Gartner views cloud computing as a service where in scalable and elastic IT-enabled capabilities are delivered using internet technologies. Reduced cost for storing data in and retrieving data from cloud is the biggest driver for its expected growth. The use of cloud computing is growing, and by 2016 this growth will increase to become the bulk of new IT spend [1]. 2016 will be a defining year for cloud as private cloud begins to give way to hybrid cloud, and nearly half of large enterprises will have hybrid cloud deployments by the end of 2017[1].

Cloud computing, uses web technologies and remote servers to store and maintain the user data. In [1], Gartner views cloud computing as a service where in scalable and elastic IT-enabled capabilities are delivered using internet technologies. Reduced cost for storing data in and retrieving data from cloud is the biggest driver for its expected growth. The use of cloud computing is growing, and by 2016 this growth will increase to become the bulk of new IT spend [1]. 2016 will be a defining year for cloud as private cloud begins to give way to hybrid cloud, and nearly half of large enterprises will have hybrid cloud deployments by the end of 2017[1].

The remainder of this paper is organized as follows: In Section II, literature review of existing systems is discussed. Section III discusses the problem on which the research is carried out. In Section IV, the design goals of proposed system are given. Results of the proposed system are analysed in section V. Finally, conclusions are given in Section VI.

ISSN: 1738-9976 IJSIA

2. Related Work

Many researchers does extensive study on preserving privacy of user data in cloud. In [7] for the first time, authors have proposed practical symmetric searchable encryption method. In this scheme the file is encrypted word by word. To search for a keyword user sends the keyword with same key to the cloud. The drawback of this scheme is that the word frequency will be revealed. In [15], the first public key encryption with keyword search (PEKS) was proposed. The scheme suffers from inference attack on trapdoor encryption method. In [5,9-18] efficient techniques have been proposed to preserve privacy of user data in cloud. Later, the schemes in [3, 6, 19] proposed use of multi keyword search techniques for searching. [21] [22] are two survey papers wherein authors have discussed different techniques available in literature that work on encrypted data, along with comparative study of different searchable and homomorphic encryption schemes. In [20], authors have proposed a new scheme based CRSA and B+ tree and its performance is analyzed.

As an outcome of this literature review, we have proposed a novel system of privacy preserving multiple-keyword ranked search over encrypted cloud data which is supported with efficient search method. Thus, a high performance security model with multi-keyword search evaluation mechanism is proposed.

3. Problem Formulation

The confidentiality and privacy of user's data can be maintained by using Searchable Encryption (SE) techniques, which does search on encrypted data. Multiple techniques like cryptography, searching, storage *etc*. are used for designing efficient, secured, SE schemes. The secure search model involving data owner, data user and cloud server as discussed below.

System Model: The SE architecture consisting of 3 participants, the data owner, data user and the cloud server as shown in Figure 1.

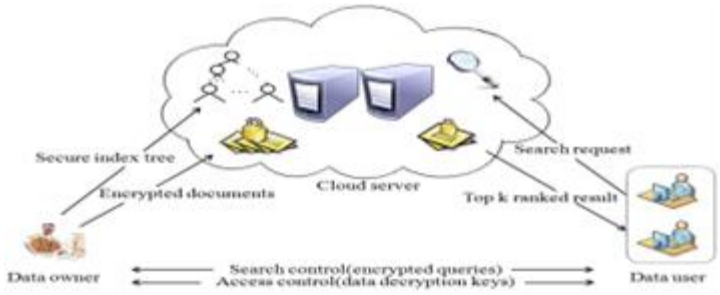


Figure 1. Searchable Encryption Architecture ECC/B+ Tree

1. The encrypted data files along with encrypted index file are uploaded to cloud server. ECC algorithm is considered for encryption of data. B+ tree data structure is used for indexing keywords and to search. 2. Cloud server uses encrypted keywords (trapdoor) and returns ranked relevant data files to the user. 3. Data user search for the relevant files in encrypted cloud dataset by sending encrypted keywords (trapdoor). The encrypted keywords preserve the security and privacy of user data in cloud.

Design Goals: The solution addressed by our proposed system for the requirements are 1.Secured privacy preserving search on encrypted document by cloud server 2.The search results must be ranked in order of relevance. To enable ranked searchable encryption for effective utilization of outsourced and encrypted cloud data under the aforementioned model, our system design should achieve the following security and performance guarantee. Specifically, we have the following goals: 1) Ranked keyword search: to

explore different mechanisms for designing effective ranked search schemes based on the existing searchable encryption framework; 2) Security guarantee: to prevent cloud server from learning the plaintext of either the data files or the searched keywords, and achieve the "as-strong-as-possible" security strength compared to existing searchable encryption schemes; 3) Efficiency: above goals should be achieved with minimum communication and computation overhead.

Existing Systems: In [6,15,8], authors discuss secure multi-keyword search over encrypted data. The similarity measure "coordinate matching" in MRSE [6] has some drawbacks when used to evaluate the document ranking order. First, it takes no account of term frequency *i.e*. any keyword appearing in a document will present in the index vector as binary value 1 for that document, irrespective of the number of its appearance. Obviously, it fails to reflect the importance of a frequently appeared keyword to the document. Second, it takes no account of the term scarcity. Usually a keyword appearing in only one document is more important than a keyword appearing in several ones. In addition, long documents with many terms will be favored by the ranking process because they are likely to contain more terms than short documents. Hence, due to these limitations, the heuristic ranking function, "coordinate matching", is not able to produce more accurate search results. More advanced similarity measure should be adopted from plaintext information retrieval community. On the other hand, the search complexity of MRSE is linear to the number of documents in the dataset, which becomes undesirable and inefficient when a huge amount of documents are present. In [20], author proposed an efficient SE scheme which overcomes the drawbacks of MRSE [6].

Proposed system: B+ tree is used as an indexing data structure to match between search query and data documents. Specially, we use inner data correspondence, *i.e*., the number of query keywords appearing in document, to evaluate the similarity of that document to the search query. Each document is indexed in a balanced B+-tree and is encrypted using ECC. The Encrypted keywords (trapdoors) created by user is used by cloud server to perform search on encrypted dataset. Our aim is to design and analyze the performance of multiple-keyword ranked search scheme using ECC for encryption and B+ tree data structure for search. We used Microsoft's Azure platform to emulate the proposed system and to study its performance.

4. SE Framework Using ECC/B+ Tree

4.1. Encryption Framework Using ECC

Researchers worked on exploring new cryptographic methods that would serve as good trapdoor functions apart from factoring algorithms like RSA and Diffie-Hellman. ECC was introduced by Victor Miller and Neil Koblitz in 1985 [17] [26]. Quicker, shorter and more efficient cryptographic keys are created based on ECC which relies on elliptic curve theory. Instead of traditional way of calculating the product of two very big prime numbers, ECC invokes keys through the properties of elliptic curve equation.

In [23], a cryptographic algorithm is proposed which is based on an esoteric branch of mathematics called elliptic curves. The advantage of ECC lies in requirement of a smaller key size compared to other asymmetric algorithms like RSA and Diffie-Helmann. Smaller key size used in ECC reduces the storage and transmission requirements, which in turn leads to faster processing. ECC is useful for implementing encryption on data with less power, CPU time and memory. Thus, the elliptic curve should be recognized for its computational and bandwidth support while it offers comparable security.

The ECC is based on elliptic curve discrete log problem. The elliptic curve is represented as shown in the Figure 2. It is a curve which intersects two axes and is based on the properties of the equation that extract from points where the line bisects the axes. Points A and B are added, then we get a point C on curve E. Even though knowing

original point and the result, it is hard to find what number was used to perform the product. Equations based on elliptic curves have a unique objective such as it is correspondingly easy to perform in one direction but intensively difficult to reverse.

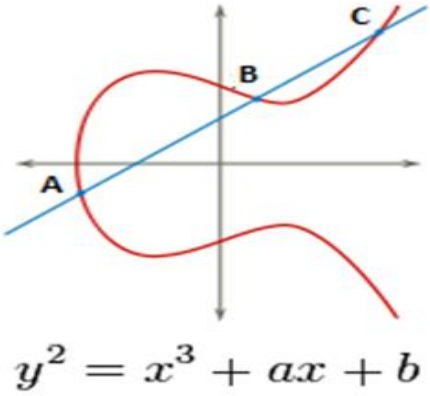


Figure 2. Elliptic Curve

An elliptic curve E (Fp) over a finite field Fp is defined by the parameters a, b ϵ Fp (a, b satisfy the relation $4a^3 + 27b^3 \neq 0$), consists of the set of points (x, y) ϵ Fp, satisfying the equation $y^2 = x^3 + ax + b$.

Key Generation:

Here, it should generate both the keys *i.e*. public key and private key. Encryption takes place with the user's public key and the decryption takes place based on the user's private key. Let us select a number n within the range of I (Maximum limit prime number)

Public key is calculated using the equation B= n × A, Where, n is the random number chosen within the range (1 to i-1). A is the point on the curve, B is the public key and n the random number is the private key.

Encryption:

1. Character is read from a file (for example 'a').
2. Store characters ASCII value into an integer variable (K=97).
3. Point is selected on the Elliptic Curve corresponding to the integer ({for example E (17, 30)} for K=97)
4. The point is encrypted with user's public key B= n × A and the corresponding encrypted point be E´ (*i.e*.(21,124)).
5. Map point E´ is to the database to obtain new integer. The new integer corresponding to E´ is K^1 (say 43).
6. This K^1 is converted to data consisting of two parameters:

a. Printable ASCII character ($) which acts as an index.
b. Page no. (*e.g*. N=1) to which the corresponding index belongs to.

7. Two parameters are sent into two different files which are transmitted.

The server will encrypt the message with the user's public key. Let the message that has to be sent is s. This message s should be represented on the curve. Let us consider s has the point $\mathcal{S}$ on the curve E. Now, select q randomly from $[1 - (i - 1)]$. The two cipher texts will be generated. Let the cipher texts be T_1 and T_2. $T_1 = q \times \mathcal{A}, T_2 = \mathcal{S} + q \times \mathcal{B}$. Thus, finally T_1 and T_2 will be sent.

4.2. B+ Tree Search Algorithm Framework Using Microsoft Windows Azure

Balanced B^+ tree algorithm proposed enables effective, efficient and secure multi-keyword ranked search over encrypted cloud data. Multi-keyword search management tends to be cumbersome if it is done manually. In order to automate the multi-keyword

search management we need a common syntax and a common searchable encryption to interoperate. In this section, we introduce the searchable encryption through ECC method. The working of the B^+ tree search algorithm is as follows.

Firstly the cloud user will upload the data files on to the cloud. Whenever the user wants to retrieve the relevant data files based on the user's requirement, the user will search through the documents with the set of keywords. The user may input many keywords to search a document. The set of all multi-keyword search is represented by S_K. The s_{k_a} represents the a^{th} search keyword in many keywords. The tree structure algorithm consists of the encrypted tree data R_K, the encrypted keyword contents O_K and the tree builder function f_O. The s_{k_a} is derived from both the encrypted tree data $r_{k_a} \in R_K$ and the encrypted keyword contents $o_{k_a} \in O_K$ of the Tree Structure Algorithm (TSA). o_{k_a} is the contents derived from the r_{k_a} of R_K by the tree builder function f_O. The tree builder function extracts all the related keywords of r_{k_a} present in the encrypted tree data R_K of the TSA.

The workers in the B^+ tree provide search encryption services which support the multi-keyword search application. The workers are represented as WR and Wr_n is the n^{th} search provided by the worker. Each search provided by the worker possess the encrypted tree data based documents/records represented by R_{KB}. rkb_r is the r^{th} encrypted tree data record available with the azure cloud search provided by the worker $Wr_n \in WR$ on the n^{th} search. The encrypted keyword contents of r encrypted tree data records is given by O_{KB}. The implementation of all these are as follows.

$$S_K = \{s_{k_1}, s_{k_2}, s_{k_3}, \ldots\ldots\ldots s_{k_a}\}$$
$$s_{k_a} = f_k(r_{k_a}, o_{k_a})$$
$$o_{k_a} = f_O(R_K, r_{k_a})$$
$$R_K = R_{K1} \cup R_{K2} \cup R_{K3} \cup \ldots\ldots\ldots R_{Kn}$$
Where $R_{K1} \neq R_{K2} \neq R_{Kn}$
$$R_K = \{r_{k_{11}}, r_{k_{12}}, \ldots\ldots r_{k_{1a}}\} \cup \{r_{k_{21}}, r_{k_{22}}, \ldots\ldots r_{k_{2a}}\} \cup \ldots\ldots\ldots \{r_{k_{n1}}, r_{k_{n2}}, \ldots\ldots r_{k_{na}}\}$$
$$O_K = O_{K1} \cup O_{K2} \cup \ldots\ldots \cup \ldots\ldots\ldots O_{Kn}$$

Where R_{Kn} and O_{Kn} are the encrypted set available with n^{th} search service provided by the worker. The locally available encrypted data could be defined as $O_{Kn} \propto R_K$.

In the current search deployments available, there exists a problem where the encrypted data available with search service n which is provided by the worker, may not contain all the possible keywords, as the complete encrypted tree data set R_K. The purpose of the B^+ tree is to overcome the short comings by using efficient searching algorithms and search encryption compositions. The huge data of the cloud search provided by the worker, constitutes both the encrypted tree data and encrypted keyword contents. A search executed on huge databases would affect the response time. Since it involves numerous disk read and disk write operations in the search operation. To compress the data and create cache the B^+ tree utilizes a hierarchical data ordering algorithm.

$$R_{KB} = \{rkb_1, rkb_2, rkb_3, rkb_4, \ldots\ldots\ldots rkb_r\}$$
$$O_{KB} = \{okb_1, okb_2, okb_3, okb_4, \ldots\ldots\ldots okb_r\}$$
$$Cache_{s_{k_a}} = < s_{k_a}, r_{k_a}, e_{k_a} >$$
$$rkb_r = < trkb_{r_{sub}}, trkb_{r_{prd}}, trkb_{r_{obj}} >$$

Based on the record, rkb_r could be represented as mentioned above which are said to consist of triplets. Where $trkb_{r_{sub}}$ is the subject triplet, $trkb_{r_{prd}}$ is the predicate triple and $trkb_{r_{obj}}$ represents the object triplet. The keywords extracted from the encrypted tree data include some complex relations. Complex relations cannot be represented in encrypted tree data alone. Hence the B^+ tree presented here adopts representation of the encrypted keyword contents through tree structure builder due to its benefits. As the

number of keywords increases, the relations, the data size and number of disk operation will increase for the search operation. The number of occurrences of a keyword in an encrypted data is directly proportional or equivalent to the number of relations r_{k_a} of a keyword. It can be said that the number of relations (f_{num_r}) r_{k_a} of a keyword s_{k_a} and a function of the tree depth (f_{edg_dpth}) of a keyword s_{k_a} is equivalent to a constant m. Even if the number of relations r_{k_a} of a keyword s_{k_a} increases, the cache size does not increase by a great extent.

$$f_{num\_r}\left(r_{k_a}\right)_{x=m} \times f_{edg\_dpth}\left(r_{k_a}\right)_{x=m} \approx m$$

$$r_{k_a} \approx \sum_{x=1} m/x \approx m \int_{x=1} 1/x\,dx = m \ln m$$

The cloud space required to store the keyword s_{k_a} is given by S_{Util}. The space utilized in storing the cache defined above is given by

$$\sum_{r_{k_a}} (2 + f_{num\_r}(r_{k_a})) \approx S_{Util}(2 + \ln S_{Util})$$

The keywords require $2S_{Unit}$ cloud storage space per keyword (s_{k_a}) and also only one entry of a s_{k_a} keyword is allowed in the cache. To compare the normal caching strategy with the caching strategy used in B^+ tree search, the comparison ratio is established as

$$\frac{2S_{Util}\left(1 + {\ln S_{Util}}/{2}\right)}{2S_{Util} \ln S_{Util}} = 1/\ln S_{Util} + 1/2$$

Hence the proposed caching strategy improves the cloud storage space utilization by approximately 50% .

The azure cloud access cost for the caching strategy is defined as

$$ACost_{Cache} = \sum_{\{r_{k_a}: f_{num\_r}(r_{k_a}) \le t\}} f_{num\_r}(r_{k_a}) \approx \int_{S_{Util}/t}^{S_{Util}} \frac{S_{Util}}{f_{edg\_dpth}}\, df_{edg\_dpth}$$
$$= S_{Util} \ln t$$

Where $f_{num\_r}(r_{k_a}) \le t \leftrightarrow f_{edg\_dpth}(r_{k_a}) \ge S_{Util}/t$

The probability of $AProb$ finding the keyword s_{k_a} in the encrypted data is defined as $AProb_{Cache} = \ln t / \ln S_{Util}$

The access time of the cache to search for a keyword s_{k_a}within the encrypted data with a probability $AProb_{Cache}$ is defined as

$$ATime_{Cache} = AProb_{Cache} \log_b ACost_{Cache} + (1 - AProb_{Cache})(\log_b ACost_{Cache} + 1)$$
$$= \log_b ACost_{Cache} + (1 - AProb_{Cache})$$

Where b represents the branching factor of the encrypted tree.

To access faster the cache created based on the encrypted tree data and encrypted keyword content is encoded in a binary format. The encrypted relevance score is a ratio between the query keyword and the response keyword based on the encryptions constructed. The encrypted relevance score is used by the search application in ranking the search responses received by the n search service provided by workers considered in the B^+tree search. The search query SS_Q could be defined as a set of keywords and relational operators. The search query ss_Q could be represented as a $p \times q$ matrix where p represents the number of keywords queried for and q represents the number of relations, logical operators and special characters defined for querying amongst the p keywords. The search response ss_R could also be represented as a $r \times r$ matrix where r the number of responses obtained for the search query in ss_Q. The encrypted relevance score is represented by $ors_{R_{ss_R}}$.

$ss_Q = < s_{k_{SS_Q}}, R_{K_{SS_Q}} >$

$ss_R = < s_{R_{ss_R}}, ORS_{R_{ss_R}} >$

$$ors_{R_{ss_R}}(ss_Q, ss_R) = \frac{\sum_r s_{r_{ss_R}}, ss_Q}{\| ss_Q \| \| s_r \|}$$

Normalization is considered to represent the encrypted relevance score to a scale of $0\ to\ 1$.

The encrypted relevance score is $ors_{R_{ss_R}}(ss_Q, ss_R) = s_{r_{ss_R}}', ss_Q$ Where $s_{r_{ss_R}}' = \frac{s_{r\,ss_R}}{s_r}$

Let us consider search keyword set S_K and two keywords $s_{k_x} \in S_K$ and $s_{k_y} \in S_K$. There exists 4 possible relations amongst keywords s_{k_x} and s_{k_y}. The possible relations could be defined by using the subsume represented by Sb_{sum} and defined as $Sb_{sum} : (S_K \times S_K) \mapsto \{T, F\}$, Where T represents the conditional true relation and F represents a conditionally false relation.

Let us consider a parameter p_x of the search service provided by the worker Wr_x and a parameter p_y of the search service provided by the worker Wr_y . If the parameters $p_x = p_y$ then the cloud search service could be called if only $Sb_{sum}(Wr_x, Wr_y) = T$.

$Sws_{Wr} = \{ sws_{Wr_1}, sws_{Wr_2}, \ldots\ldots\ldots sws_{Wr_n} \}$, Where sws_{Wr_n} represents the n^{th} cloud search service offered by search service provided by the worker Wr_n.

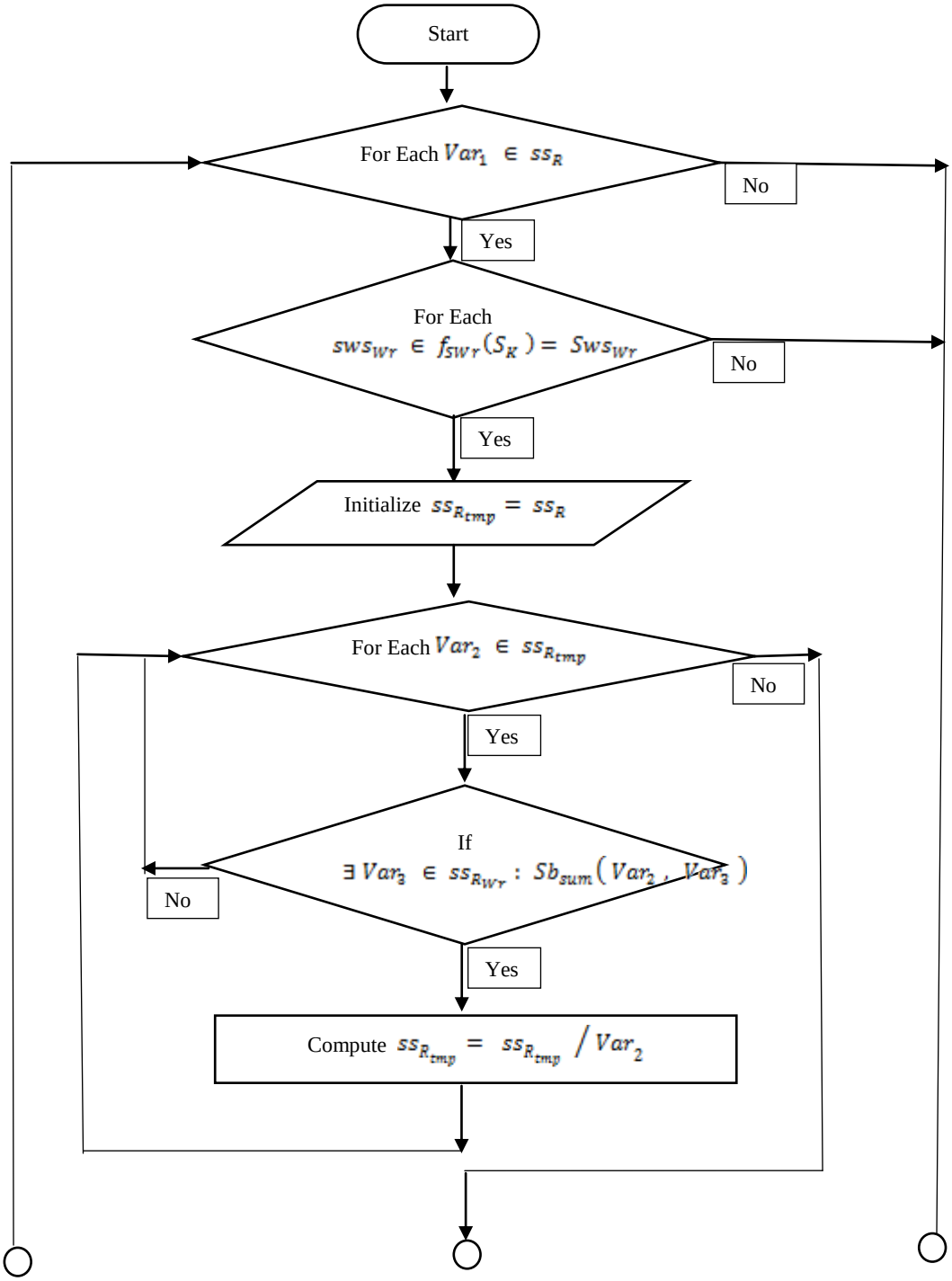
Each cloud search service offered by search worker Wr_n required a set of inputs denoted as ss_{QWr_n} and if the set of inputs is provided in an orderly fashion the cloud search service provides a set of output keywords denoted by ss_{RWr_n} and $ss_{RWr_n} \in S_K$. The efficient ranked keyword search cloud service composition algorithm discovers the cloud search services available on Sws_{Wr}. Let the ranked keyword search cloud service composition be represented as $Comp_{Sws}(Sws_{Wr})$ then the cloud search service composition is said to successfully process all the requests if

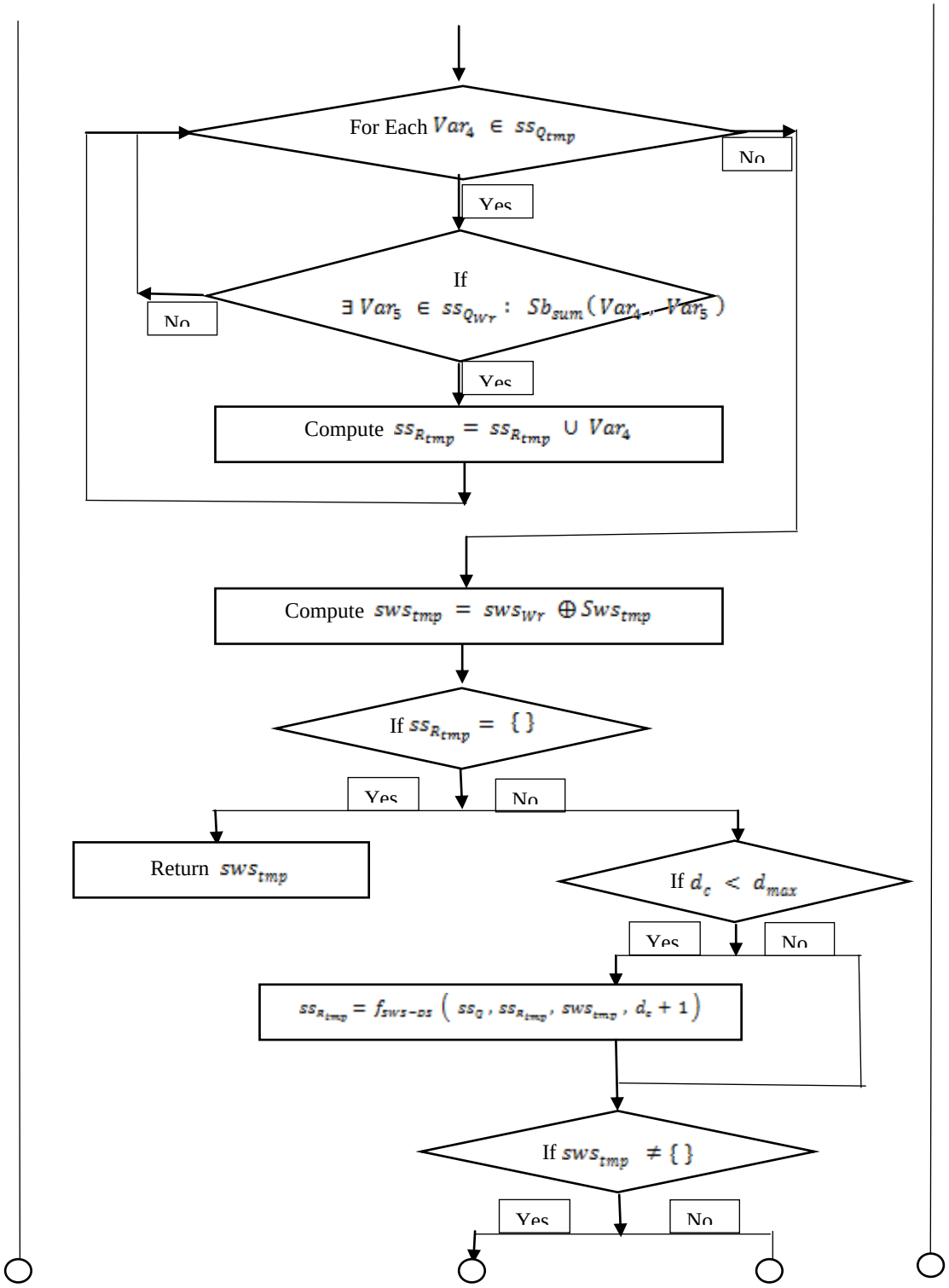
$$Comp_{Sws}(Sws_{Wr}) \leftrightarrow \forall \mathcal{X} \in ss_{QWr_1}\ \exists\ \mathcal{Y} \in SS_{QWr} : Sb_{sum}(\mathcal{X}, \mathcal{Y}) \wedge \forall \mathcal{X} \in ss_{QWr_2}, z \in \{2, 3, \ldots, n\}$$

$$\exists\ \mathcal{Y} \in SS_{QWr} \cup ss_{RWr_{z-1}} \cup \ldots . \cup ss_{RWr_1} : Sb_{sum}(\mathcal{X}, \mathcal{Y}) \wedge \forall \mathcal{X} \in ss_{RWr_1} \cup \ldots \cup ss_{RWr_n} \cup SS_{QWr} : Sb_{sum}(\mathcal{X}, \mathcal{Y})$$

Let f_{SSWS} represent a service provided by worker on search function based on a keyword S_K which provides all the set of cloud search services available defined as $f_{SWr}(S_C) = Sws_{Wr}$ Also it could be stated that $\forall s_{k_a} \in f_{SWr}(S_C)\ \exists\ ss_{RWr_a} \in SS_{RWr} : Sb_{sum}(S_C, ss_{RWr_a})$

Let us define a function f_{SWS-DS} which performs the B^+ tree search algorithm is defined as $f_{SWS-DS}(ss_Q, ss_R, Sws_{tmp}, d_c) = sws$, Where ss_Q represents the input query set, ss_R is the desired response, Sws_{tmp} represents the current temporary cloud services identified, d_c represents the height and sws represents the resultant cloud service identified. The flowchart for f_{SWS-DS} is drawn below.





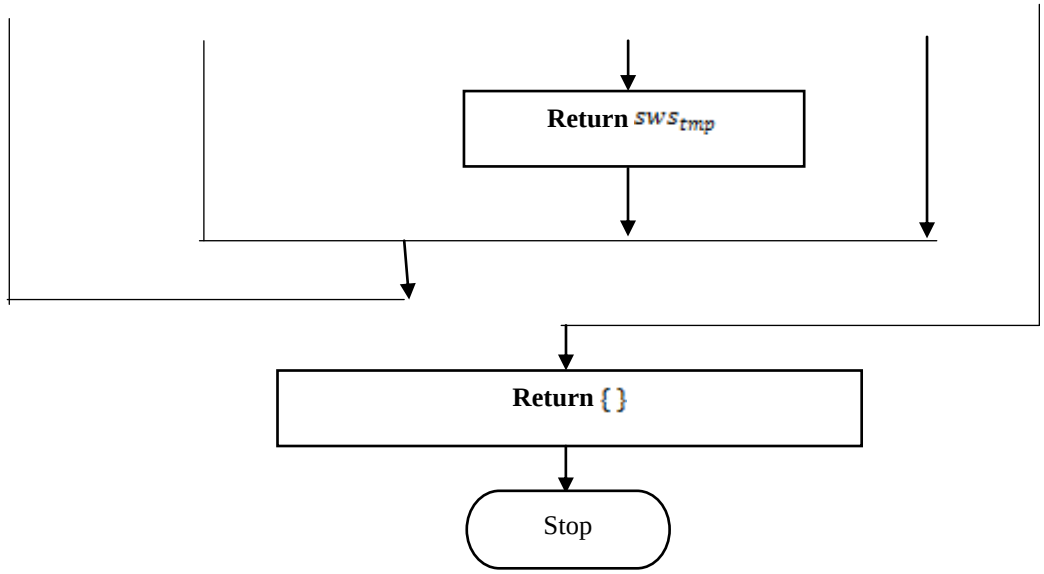


Figure 3. Flowchart for Search Algorithm Using B+ tree

5. Results

The privacy preserved multi-keyword search based on the encrypted cloud data has been implemented. The system model presented has been developed on Visual Studio 2010 framework 4.0 with C#. The overall system has been developed and implemented with Microsoft Azure platform. Different parameters like computation overhead, computation time, and bytes overhead have been considered to study and compare the performance of our proposed scheme with our previous work that was based on CRSA and B+ tree [13].

Search Computation Time:

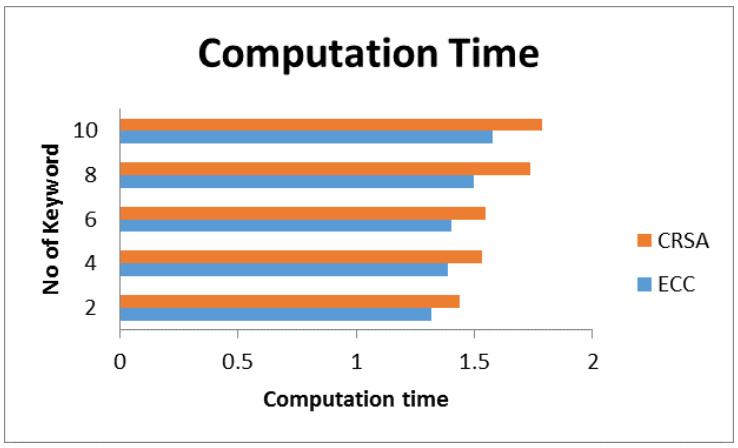


Figure 4. Computation Time for Keyword Search in MS

Figure 4, depicts the computation time ((the length of time required to perform a computational process) in seconds against up to 10 keywords for analysis. As per the experimental result the computation time for proposed ECC / B+ tree scheme is reduced

by 0.173 sec from 0.868 sec or 173 ms from 868 ms approximately over CRSA / B+ tree scheme. The practical result proves that the proposed ECC/B+ tree search algorithm performs better than CRSA / B+ tree search algorithm.

Computation Overhead (CPU Utilization):

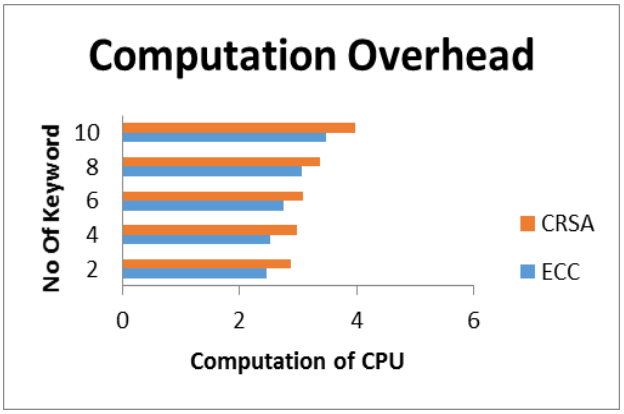


Figure 5. Computation Overhead

Figure 5, depicts the computation overhead (CPU time) in seconds against the keywords in the range 1 to 10. In this study, we compared the performance of our proposed system ECC/B+ tree with the CRSA/B+ tree [13]. Results clearly show that, the computation overhead using ECC/B+ tree is low as compared to the CRSA/B+ tree. For example, CRSA/B+ tree takes approximately 3 seconds for searching 2 keywords, whereas our proposed ECC/B+ tree based scheme takes less than 2.5 seconds. The computation overhead is linear in both schemes. But from Figure 5, it is evident that our proposed ECC based scheme performs better even under increased number of keywords. Thus, the analytical result proves that ECC search computation overhead is less compared to CRSA based search algorithm.

Tree Computation Time:

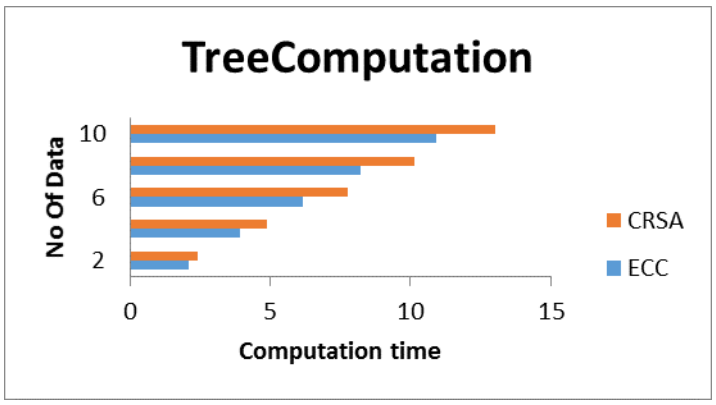


Figure 6. Tree Computation Time

The graph in Figure 6 plotted above makes the comparison of the tree search computation time in seconds of our proposed system against the CRSA/B+ tree [13]. As the number of data files increases, the computation time for search also increases linearly

in both schemes. The experimental result proves that ECC/B+ tree, search computation is reduced by 1.392 sec from 6.96 sec over CRSA/B+ tree search algorithm. This proves that our proposed ECC with B+ tree scheme performs better than CRSA and B+ tree scheme.

Byte Overhead:

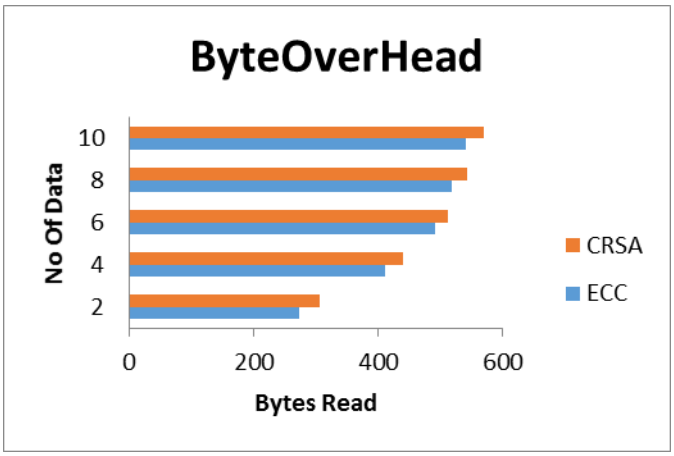


Figure 7. Byte Overhead

The graph in Figure 7 portrays the overhead computation in bytes. For two data files the number of bytes read is around 250 bytes compared to 300 bytes from CRSA/B+ tree [13]. As the number of data files increases the bytes read for search also increases linearly in both schemes. The experimental result proves that our proposed ECC/B+ tree scheme reads less bytes into memory for search over CRSA on B+ tree scheme.

6. Conclusion

The insights of privacy-assured searchable cloud data storage services are discussed in this paper. Despite the popularity of cloud services and their wide adoption by enterprises and governments, cloud providers still lack services that guarantee both data privacy and privacy preserving search operation on encrypted data. Here we tried to address the security issues considering a large set of cloud data and users based on preserving the privacy of multi keyword search over an encrypted data. We have designed an efficient cryptographic scheme using ECC and B+ tree. The B+ tree search algorithm is adopted for the ranked search technique. The C-RSA cryptographic technique induces low computation overhead with the asymmetric key. This is improved with the use of ECC technique, yielding much reduced computation overhead without compromising security.

Detailed analysis which examines the privacy and search efficiency of our proposed model is given. The experimental results prove that our proposed model induces low overhead on the overall system. Using the ECC, the computation overhead and byte overhead is much reduced compared to other cryptographic methods. Based on this comprehensive performance analysis, we conclude our scheme using ECC/B+ tree is more secure, efficient and practical than existing schemes.

References

[1] A. Singhal., 'Modern information retrieval: A brief overview', IEEE Data Engineering Bulletin, **(2001**), Vol. 24, pp 35–43.
[2] Aviv A. J., Locasto M. E., Potter S. and Keromytis A. D., 'SSARES: Secure Searchable Automated Remote Email Storage.' in Proc. of 23rd annual Computer Security Applications Conference, ACSAC **(2007).** pp 129-139.
[3] C. Wang., N. Cao., J. Li., K. Ren., W. Lou., 'Secure Ranked Keyword Search over Encrypted Cloud Data', in Proc. of IEEE ICDCS '10, **(2010)**, pp 253-262.
[4] C.Gentry., R. Z., 'Single-database private information retrieval with constant communication rate', in Proc. of intl. Colloquium on automata, Languages and programming, ICALP, **(2005)**, pp 803–815.
[5] D. Xiaoding Song., W. D., Perrig A., 'Practical techniques for searches on encrypted data', in IEEE Proc. on Security and Privacy, S&P 2000, **(2000)**, pp 44-55.
[6] D. Boneh., Crescenzo G. D., Ostrovsky R., Persiano G., 'Public key encryption with keyword search', in Proc. of EUROCRYPT, LNCS, Vol 3027, **(2004)** , pp 506-522.
[7] Gartner., 'Analysts Examine Cloud Strategies and Adoption', Gartner Symposium/ITxpo**, (October-2013)**, 21-24, Goa, India, Available: http://www.gartner.com/newsroom/id/2613015.
[8] IT Business Edge., 'Data Privacy, Integration, Rank as Top SaaS Concerns for Large Companies',**(2014)** Available: http://www.itbusinessedge.com/blogs/integration/data-privacy-integration-rank-as-top-saas-concerns-for-large-companies.html.
[9] Li M., Shucheng Yu., Ning Cao., Wenjing Lou., 'Authorized Private Keyword Search over Encrypted Data in Cloud Computing', in Proc. 31st Int'l. Conf. Distributed Computing Systems, ICDCS'10, **(2011)**, pp 383–392.
[10] Neal Koblitz., 'The state of Elliptic Curve Cryptography. Designs, Codes and Cryptography',**(2000)**, Vol 19, pp 173-193.
[11] Ning Cao., Cong Wang., Ming Li., Kui Ren., Wenjing Lou., 'Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data', in IEEE transactions on Parallel and Distributed Systems, **(2014)**, Vol.25, pp 222-233.
[12] Nagamalli A., Krishna A., Keerthi S., Priya B., 'Data encryption using Counting Bloom Filters for cloud security', in 3rd Intl conference on Computational Intelligence and Information Technology, **(2013)**, pp 514-518.
[13] Prasanna B T., C B Akki., 'Performance Study of Cryptography based Dynamic Multi-keyword Searchable Security Algorithm in Cloud using CRSA/B+ tree', Global Journal of Computer Science and Technology, **(2015)**,Vol 15, Issue 1, Version 1.0 pp 5-16.
[14] Prasanna B T., Akki C B., 'A Survey on Homomorphic and Searchable Encryption Security Algorithms for Cloud Computing', in IJARCSSE, **(2015)** ,Vol 4, Issue 5,pp 1012-1021.
[15] Prasanna B T., Akki C B., 'Dynamic multikeyword ranked searchable security algorithm using CRSA and B Tree', **(2015)**, IJCSIT, Vol 6(1), pp 826-832.
[16] Richard Brinkman., 'Searching in encrypted data', University of Twente, PhD thesis, **(2007).**
[17] Sun W., Wang B., Cao N., Li M., Lou W., Hou Y T., Li H l., 'Privacy-preserving multikeyword text search in the cloud supporting similarity-based ranking', in Proc. of the 8th ACMSIGSAC symposium on Information, computer and communications security, ACM, **(2013)** ,pp 71–82.
[18] Victor Miller., 'Uses of elliptic curves in cryptography', in Proc. Of Advances in cryptology, CRYPTO-85, Springer, **(1986)**, pp 417-426.
[19] Wenjun Lu., Varna A L., Min Wu., 'Confidentiality-Preserving Image Search: A Comparative Study between Homomorphic Encryption and Distance-Preserving Randomization', in IEEE Access, **(2014)**, Vol 2, pp 125-141.
[20] Wai Kit Wong., Cheung D W., Kao B., and Mamoulis N., 'Secure knn computation on encrypted databases', in Proc. Intl. conference on management of data, SIGMOD,**(2009)** , pp 139-152.
[21] Wikipedia., 'Elliptic curve cryptography', **(2015)**, Available: http://en.wikipedia.org/wiki/Elliptic_curve_cryptography.
[22] Zhangjie Fu., Xingming Sun., Zhihua Xia., Lu Zhou., Jiangang Shu, 'Multikeyword Ranked Search Supporting Synonym Query over Encrypted Data in Cloud Computing', in 32nd IEEE Conference on performance computing and communications conference, IPCCC, **(2013)**, pp 1-8.
[23] Zhangjie Fu., Xingming Sun., Linge N., Lu Zhou., 'Achieving effective cloud search services: multi-keyword ranked search over encrypted cloud data supporting synonym query', in IEEE transactions on Consumer Electronics**, (2014)**, Vol.60 pp164-172.
[24] Prasanna B T., C B Akki., 'A comparative study of homomorphic and searchable encryption schemes for cloud', IJASCSE, (indexed in IET-Inspec), **(2015)**, Vol 4, Issue 5.

Authors

Prasanna B T, he received the Bachelor's Degree in Computer Science and Engineering from Siddaganga Institute of Technology, Tumkur, India in 2000; He received his Master's Degree in Computer Science and Engineering from University BDT College of Engineering, Davanagere, India in 2005. He is currently an Associate Professor at East Point College of Engineering and Technology, Bengaluru, India. He has both academic and Industrial experience. His Special interests includes Wireless Communication, Computer Networks and Cloud Computing. Published research papers in reputed journals.

Dr. C.B.Akki, he received the Bachelor's Degree in Electrical Engineering from University Vishvesvaraiah College of Engineering, Bengaluru, India in 1982, He received his Master's Degree and Ph.D in Computer Science and Technology from University of Roorkee (IIT), India in 1990 and 1997 respectively. He is currently a professor at SJB Institute of Technology Bangalore, India. He has both academic and Industrial experience in India and abroad. His Special interests are Wireless Communication, Mobile Computing and Computer Networks. He is a member for several research bodies and published numerous research papers in reputed journals.

