International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 05 | May-2016 www.irjet.net p-ISSN: 2395-0072

IMPROVED APPROACH FOR ENCRYPTED CONTENT SEARCH IN MOBILE CLOUD SERVICE

Sasikala.K[1], Karthik.S[2], Santhana Mani.J[3]

[1] *Associate professor, Information Technology, VMKV Engineering College, Salem, India*
[2] *Associate professor, Information Technology, VMKV Engineering College, Salem, India*
[3] *PG Student, Information Technology, VMKV Engineering College, Salem, India*

Abstract - *In today's world the cloud service is rapidly growing popularity to store the data. However, this will lead to the risk to consumers unless the data is encrypted for security. The encrypted data for mobile client should be effectively searchable and retrievable without any privacy leaks. Many security issues have solved in recent research but architecture cannot be applied on mobile services under the mobile cloud service. The main problem arises due to wireless network, such as low transmission, poor connectivity and latency sensitivity. This causes more search time and large network traffic costs while using traditional search schemes.*

This paper addresses some issues by proposing an improved approach for encrypted content search in mobile cloud service. The new scheme uses a lightweight trapdoor (encrypted keyword) compression method, which optimizes the data communication process by reducing the trapdoor size for traffic efficiency. Ranked serial Binary Search (RSBS) algorithm and Trapdoor Mapping Table (TMT) module are the two optimization methods used for the document search; to speed the search time is the main objective of this paper. The search time and traffic network is reduced in improved approach for encrypted content search.

Key Words: **Mobile User, Encrypted content Search, Optimization, Trapdoor, Cloud Storage.**

1. INTRODUCTION

Cloud Computing refers to manipulating, configuring, and accessing the applications online. It offers online data storage, infrastructure and application. It is not to install a piece of software on our local PC and this is how the cloud computing overcomes platform dependency issues. Hence, the Cloud Computing is making our business application mobile and collaborative. In other words, Cloud is something, which present at remote location. Cloud can provide services over network, i.e., on public networks or on private networks, i.e., WAN, LAN or VPN. Applications such as e-mail, web conferencing, customer relationship management (CRM), all run in cloud. Many data provider is not sending the data directly to the user instead they were using the advanced cloud service to send the data to the user. The Encrypted content store provides content encryption at rest capability. This is done by scrambling plain text into cipher text (encryption) and then back again (decryption) with the help of symmetric and asymmetric keys. When a document is written to the Encrypted content store, the Encrypted content store uses symmetric encryption to encrypt the document before it is written to the wrapped content store. A search engine is an information retrieval system designed to help find information stored on a computer system. The search results are usually presented in a list and are commonly called hints.

2. RELATED WORK

To protect data security and improve search efficiency many research have focused on encrypted search schemes. The encryption algorithms and noise methods are mainly used for the data security and for performance efficiency, including the Boolean keyword search algorithm and the Ranked keyword search algorithm. In earlier, encryption algorithm foe data security cannot directly apply to mobile cloud because it is hard to achieve efficient network traffic and search time to address the important issues for mobile cloud. Information leaks even a one to one mapping order preservation encryption method [21]. A one-to-many mapping order preserving encryption method that requires a complex computation process, and therefore is not suitable for the mobile cloud [13]. To retrieve data from encrypted cloud data using an order preserving encryption which preserved security perfectly [14],[22],[23]. Fully homomorphism encryption methods proved themselves secure and accurate enough for searching encrypted data purpose [15],[24],[25],[26]. In traditional encryption search adding noise in the trapdoor is a popular method to protect trapdoor security.

Adding duplicate trapdoor into target trapdoors the optimized noised method is easily obtained [16].An efficient encryption algorithm, fast accumulated hash (FAH)[1], [2], [3], to encrypt document's index and keywords in improved approach for encrypted content search. Earlier for performance efficiency, many research focused on search algorithm in the server side. Information retrieval will be divided into the Ranked keyword search and Boolean keyword search. The Boolean keyword search returns every document or nothing [28][29][30].So the Boolean search is cannot be used for the reducing the network traffic. The relevance document will be return according to the ranked keyword search. A single round trip encrypted search scheme, but their system is not secure enough, as it leaks the keyword and associated document information from multiple keyword searches [13]. A multi keyword search method, but when the field of the record becomes large, their index

building procedure would be extremely time-consuming and their trapdoor vector would be very large [12]. By using the Multi-keyword search methods traffic and search time inefficiency occur due to two network round trips intake in this method [4],[16].To overcome this problem ,a ranked keyword search algorithm and optimize it with binary tree principle is used for the search efficiency. To reduce the two networks round trip into single network round trip the trapdoor mapping table is used.

2. TRADITIONAL ENCRYPTED SEARCH DESIGN

2.1 Architecture of Traditional Encrypted Search System

The document and indexes are basically encrypted before sending to the cloud for search and to provide data security [11],[12]. When users need to retrieve particular document, they first send keywords to the original data provider. The encrypted keywords (trapdoor) are created by provider and return the trapdoor to user. The trapdoor will be send by the user to cloud. When receiving the trapdoor, the Cloud uses a search algorithm to select a desired encrypted document based on the encrypted indexes and by trapdoors. Finally, the user receives these encrypted search result and uses the private key from the provider to decrypt the documents. This architecture, as shown in Fig-1**,** protects data security while entitling the provider to use computation and storage power of the Cloud for document searches. In privacy preserving search system this architecture has been well adopted [12],[13],[14],[15],[16].

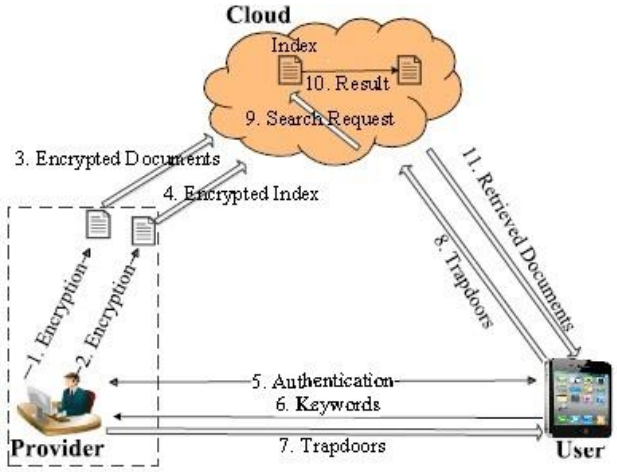

Fig -1: Architecture of Traditional Encrypted Content Search System

In today's life, the users highly utilize mobile devices to request document search services. The mobile devices are mostly connected to the Internet mainly through wireless networks such as WiFi/3G/4G/LTE, which causes some challenges as compared to traditional wired networks. The challenges are;

1) Latency sensitivity: these causes longer network latency, which results in the slow down a single search request if the search request requires many network round trips for traditional design shown in Fig-1. , three round trip is taken place for a single search request and results in the notable latency for wireless communication.

2) Poor connectivity: as the mobile devices are mostly for energy saving purpose, it cannot be connected for the long time with Cloud. There will be more number of request will be occur while the operation is carried out and it requires the extra authentication costs.

3) Low network transmission rate: The transmission rates are slower because of the mobile device are equipped with less power transmission components.

The traditional system in Fig-1, Shows the user and the provider, requires two networks round trip for one for authentication and another for trapdoor generation. The mobile users cannot uses the privacy preserving searches due to longer search delay and more bandwidth consumption. The measurement of a search request in the traditional system could produce trapdoors with a size up to 1.2MB [18].

This paper mainly focuses on the traffic and search time inefficiency issues over the mobile cloud service. This problem can be overcome by the improved approach for encrypted content search in mobile cloud service. For network traffic, the pre-computes trapdoor for common search keywords and cancel the one network round trip for re-computing trapdoor per request. The process have been still modified several mechanism to compress trapdoors and the pre-computed trapdoor table has a size of 0.33 MB and it can be effectively stored and loaded in mobile device memory. According to the binary tree principle, the Ranked Serial Binary Search (RSBS) algorithm will reduce the command time in the cloud.

In traditional encrypted search architecture, network traffic and search time, shows that the conventional approach is not applicable in mobile-cloud environments. The improved approach for encrypted content search in mobile cloud service is used to overcome these problems. The architecture uses the trapdoor compression method to reduce the traffic while request is send, as well as a Trapdoor Mapping Table (TMT) module and RSBS algorithm to reduce search time.

3.2 Issues in Traditional Encrypted Search Content System

As shown in Fig-1, the traditional encrypted search content system over the cloud is composed of three different participants, Provider, Cloud and User, which are as follows below,

- The Provider provides a set of documents and their indexes. This will be send to the cloud and from there the user have to search the content retrieve the document.

- The Cloud is something, which present at remote location. The large amount of the data can be stored for the later use by the user at the time of retrieval of document.

- The User is someone who submits keywords to search documents that contain these keywords. In our scenario, users would use mobile device such as Smartphone's and tablets to submit search requests. **Fig-1** consists of three steps: documents and indexes uploading process trapdoor generation process and document retrieval process.

Documents and indexes uploading process

The provider will store all words in these documents in the cloud and retains these terms. Then every term is encrypted and considered as one index's keyword. [19], [20]. In document index the frequency of each term in the document set is counted and then written into it. The provider encrypts this index and outsources it to the cloud with the encrypted documents. This index is a word frequency table encrypted by the computable encryption algorithm. Few researches have used the Fast Accumulated Hash (FAH) algorithm to achieve these purposes [1], [2], [3].

Trapdoor generation process

The user has to authenticate with the provider when they have to do some search request. The provider would send its secret key to the user to decrypt the documents stored in cloud during authentication. After authenticated the user would send the search keywords to the provider. The provider then computes trapdoors, commonly with FAH algorithms and replies back. In this process, two round trips are required (trapdoor generation and authentication) for a user to obtain the trapdoor for the search keywords.

Document retrieval process

In this method, the noised trapdoor is send to the cloud by user. The cloud then removes noise in the trapdoor and searches the indexes with a search algorithm. When documents are found, the cloud ranks them according to each document's score. Then relevant documents are chosen and sent to user. The Ranked Serial Search (RSS) algorithm [4], [5] is chosen as the search algorithm.

3.3 Search Time Inefficiency Problem

The trapdoor generation time and document search time results in search delay in the process. Trapdoor generation time faces challenges in mobile wireless networks: high communication latency, poor connectivity and low network transmission rate.

3.4 Network Traffic Inefficiency Problem

The provider will provides data security by trapdoor generating process .Although the trapdoor should be transmitted twice per request (between the provider and the user plus between the user and cloud). **Fig-1** trapdoor generation process and document retrieval process are the two network communication round trip for traditional system.The authentication process as well as transmitting target documents from the cloud to the user. So the total network traffic of the traditional system depends on network traffic cost when generating trapdoors.

4. IMPROVED APPROACH FOR ENCRYPTED CONTENT SEARCH DESIGN

4.1 Architecture of the improved approach for encrypted content search system

The new trapdoor generation process is introduced in the improved approach for encrypted content search design. While comparing with the improved approach for encrypted content search system Fig-2 with traditional system Figure 1, the main difference is that (1) network traffic is reduced by a single round trip information exchange and the trapdoor compression method; and (2) the search time is reduced by the RSBS algorithm and the TMT module; and (3) the computing burden for generating trapdoors is also offloaded by the TMT module. Aforementioned performance benefits are enabled by a new trapdoor generation process and new search algorithm.

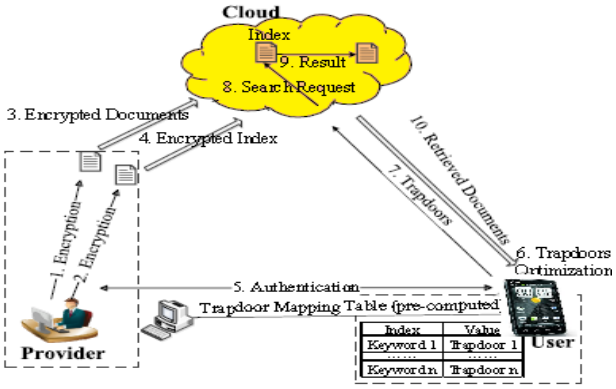

Fig -2: Architecture of Improved Approaches for Encrypted Content Search System

To reduce search delay and network traffic new trapdoor generation process and the cloud search algorithm are used in this system. For trapdoor generation, the improved approach for encrypted content search stores a pre-computed Trapdoor Mapping Table (TMT) in mobile devices, which maps common English words to corresponding trapdoors. When the mobile device initiates a search request, the trapdoor is looked up from the table instead of being requested from the provider. This optimization saves one network round trip for the trapdoor generation.

4.2 Overview of Substitute New Trapdoor Generation Process

This process includes the trapdoor mapping table and the trapdoor compression algorithm. The new trapdoor generation process, it is not necessary for an authenticated

user to calculate pure trapdoors. After a keyword is combined, a user can just query the trapdoor mapping table for the trapdoors, as shown in Algorithm 1. Since the trapdoor mapping table stores the information needed for mapping and search, the heavy computation for generating trapdoors is not needed to be conducted online. This not only avoids the recalculation if the term is found, but also reduces the number of necessary round trips from two to one. A lightweight trapdoor compression method is used to extract each trapdoors characteristic bits, record as well as accumulate location of each characteristic bit in order, and transmit the compressed trapdoor to the cloud. The compressed trapdoor will lead to additional reduced traffic cost for transmitting the trapdoors to the cloud.

ALGORITHM 1 Trapdoor Generation Process

Input:

Keyword: K
Mapping function in FAH algorithm: M ()
Hash function in FAH Algorithm: H ()
Noise set: N

Output:

Index: Compressed trapdoor t

1. Extract the term t from k
2. **if** the term t hits in the TMT module **then**
3. Obtain its pure trapdoor without any noise
4. **else**
5. Hash it by H() and Map it by M() ,there will be some bits found
6. **end if**
7. Choose f noises from the noise set N to form a subset
8. Each characteristic bit 0 to calculate the location to get a compressed Trapdoor
9. **return** t

4.2.1 Trapdoor Compression

The trapdoor compression method is used in this system. This trapdoor compression method is that utilize the location of each trapdoor's characteristic bit to represent this trapdoor, since characteristic bit 0 can show all the features of the trapdoor and also occupy a much smaller proportion compared with non-characteristic bit 1.

4.2.2 Trapdoor Mapping Table Module

The trapdoor from provider side will make a long calculation time. The calculation of generating a trapdoor of a given keyword is constituted by term combination, encryption and adding noise by the provider in a traditional system. Figure 3 displays three columns, denoting the total calculation time for generating trapdoors for one keyword, two keywords and three keywords respectively. As shown in Chart 1, the encryption time occupies nearly 82% of the total calculation time. This is because that the encryption operation requires more computing resources than others, as it accumulates all terms together to generate a hash code.

The trapdoor generation process utilizes a Trapdoor Mapping Table (TMT), which stores a large amount of frequently-used trapdoors (since an English vocabulary of just 3,000 words provides coverage for around 95% of common texts [6], here we assume a proper size of keywords is about 3,000 words) calculated offline. The key for this trapdoor mapping table is a term from stemmed keywords, while its value corresponds to encrypted terms (a pure trapdoor without any noise). Encrypted keywords have a small size. So we selected 5,893 different words (including 3,000 common words and 2,893 rare/uncommon words [7]) as keywords to be encrypted, and then stored them in TMT module. Chart-1, which display three columns, denoting a single keyword, two keywords and three keywords respectively.

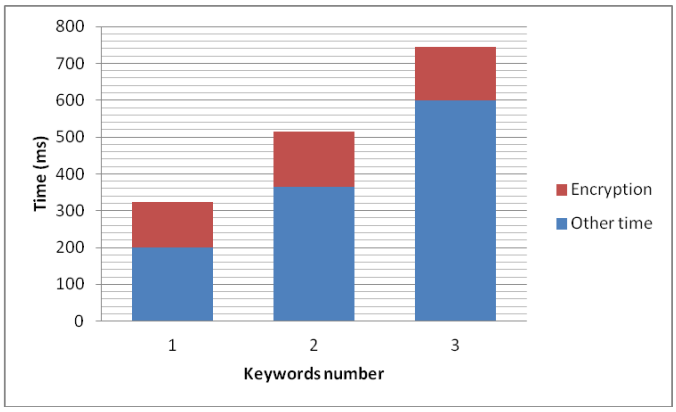

Chart -1: Trapdoor calculation time

4.3 Efficient Search Algorithm

4.3.1 Document Index Construction

The provider will construct document indexes and send to the cloud. Then cloud uses the indexes produces by the provider for quickly search document. In common, there are two main matrices are commonly used to generate the index of document [8]. The Inverted Document Frequency (IDF) matrix represents the significance of rare terms that are used to distinguish documents. The Term Frequency (TF) matrix gives the frequency of each term in document. The new matrix A will be formed in the multiplication of these two matrixes. Rather than TF and IDF matrix, matrix A will encrypted and outsourced to the cloud.

4.3.2 Index Encryption

This is another process for the provider were, before sending the index to the cloud they encrypts each index with a given FAH algorithm by encrypting each index's slices. This process is based on the privacy preserving searching system

[16],[1],[5].By using this FAH algorithm ,the en crypt slices of each index. This encryption process, the characteristic bits in each term is preserved, and checks the existence of the characteristic bits to determine whether a given keyword exists in the index stored in the cloud.

4.3.3 Binary Search Tree for Indexes

It is inefficient to search a trapdoor among indexes. To accelerate the search time the binary search tree for indexes is composed. By using the FAH algorithm, each document's index is processed as a hash code comprised by accumulated terms. To speed up the entire process the binary tree is used to slice the indexes. In this data structure, all accumulated terms are compressed in top level hash code. Next level, the accumulated terms of half of the index is descendant. All descendants contain the accumulated terms of half of that from its parent. The provider will add noise to the index to prevent respect to privacy leakage .In this method, each noise is accumulated to an index. The provider divides each index into slices after the plain text document indexes are produced according to the score value.

4.3.4 RSBS Algorithm

The indexes provided by the provider while receiving a trapdoor, the cloud would perform a privacy preserving search. Then it pick up top-k documents that contain the given search keywords. The Algorithm 2 explain how the RSBS algorithm is achieved.

Algorithm 2 Ranked Serial Binary Search (RSBS) algorithm.

Input

Noised trapdoor: t1

The number of document to return: k

Encrypted document indexes: E

Document request: D

Output

1. Create the scores as an N zeros
2. **for** i = 1 **to** N **do**
3. **for** n = 1 **to** E **do**
4. find the keywords appears in any of the s slice of the document
5. **end for**
6. **end for**
7. receive the top-k documents
8. **return** D

The binary search will start from the binary tree we constructed and descend to a slice that contains the keyword or find that the keyword does not appear in the document. If the keyword appears in the document, then the score will be calculated and updated to the Scores array. Otherwise, a zero will be recorded.

5. RESULT AND DISCUSSION

5.1 Experimental Environment

To evaluate the improved approach for encrypted content search system's is implemented on the private cloud with Openstack Essex [9].A virtual machine 8 GB memory for the cloud. To search and return the retrieved documents to the user, RSBS algorithm is used and written in Python program. the user utilized a mobile device utilized an Android tablet with a Cortex-A7 Quad 1.7GHz CPU, and 3GB memory. The tablet is connected to a mobile network with 64Mbps rate. The trapdoor mapping table is pre-computed on a PC and uploaded to the mobile device before experiments, which consumes 0.28MB of device storage. The encrypted document set used here is the corpus of 2,386 VOA news [10] extracted from the web site covering subjects such as politics, education, economy, military, etc.

5.2 Search Time Evaluation

The trapdoor mapping table module and RSBS algorithm are used to reduce the search time and to improve the calculation efficiency. The TMT module and the RSBS algorithm, choose 11000 keywords to search target documents and evaluate the search time for the three schemes with the document size ranging from 1KB to 10KB.In Chart-2 improved approach for encrypted content search system's saves about 45% of the time compared to Traditional text for 1 KB documents, and by 31% for 10KB documents. These results benefit from RSBS algorithm, but also the TMT module which reduces one network communication round trip.

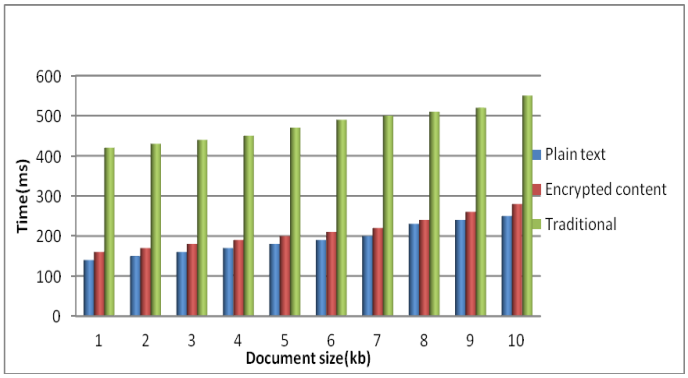

Chart -2: Performance Comparisons in Search Time

5.3 Network Traffic Evaluation

In the improved approach for encrypted content search systems, this benefits from the trapdoor compression method and the TMT module and reduced network traffic significantly. Evaluate and analyze the overall system network traffic reduction and the performance of the trapdoor compression method.

5.3.1 Performance in Network Traffic

The trapdoor compression method and the TMT module, improved approach for encrypted content search costs fewer networks than the traditional system. Chart-3 shows the throughput comparison of plain text, encrypted content, and traditional system. The transmission speed for the 1KB-size document is most effective, and the speed increases from 35 KB/s to 60 KB/s.

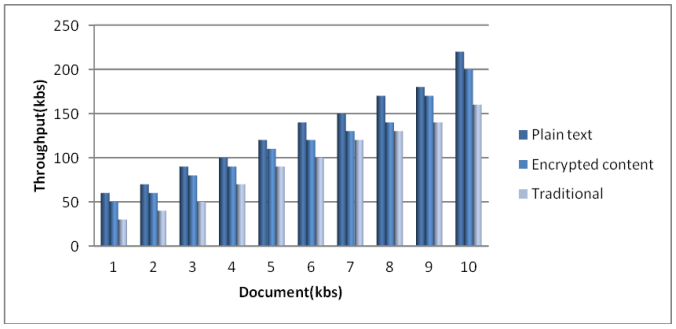

Chart -3: The Throughput Comparisons

5.3.2 Performance of Trapdoor Compression

Even after adding the some noises into the pure trapdoor, it cannot get the length of the noised trapdoor. The different number of pure trapdoors to add noise, which is randomly selected from noise, set N. Compare the size of plain text with that of the corresponding trapdoor in the traditional system and the encrypted content search system. The average length of the trapdoor in encrypted content search is reduced by 90%, while even in the worst case scenario; the length of the compressed trapdoor is equal to its corresponding plain text. To reduce network traffic costs, this simple compression method is highly effective.

6. CONCLUSIONS

The network traffic and search time efficiency improves when compared with the traditional system over the cloud. The traditional encrypted search system is fully analyzed in the mobile cloud like search time inefficiency and network traffic. To overcome this problem the improved approach for encryption content search system architecture is designed. The TMT module, reduce network traffic by trapdoor compression method. The RSBS algorithm will reduce the search time.

REFERENCES

[1] K. Nyberg, "Fast accumulated hashing," in Proc. Int. Workshop Fast Softw. Encryption (FSE), Feb. 1996, pp. 83–87.

[2] Nyberg and Kaisa, "Commutativity in cryptography," in Proc. Int. Workshop Funct. Anal., 1995.

[3] J. Benaloh and M. De Mare, "One-way accumulators: A decentralized alternative to digital signatures," in Advances in Cryptology- EUROCRYPT 1993, 1994, pp. 274–285.

[4] C. O¨ rencik and E. Savas¸, "An efficient privacy-preserving multikeyword search over encrypted cloud data with ranking," Distrib. Parallel Databases, vol. 32, no. 1, pp. 119–160, Mar. 2014.

[5] P. Wang, H. Wang, and J. Pieprzyk, "An efficient scheme of common secure indices for conjunctive keyword-based retrieval on encrypted data," pp. 145–159, 2009.

[6] S. Gendreau, "How many words do i need to know? the 95/5 rule in language learning, part 2/2," http://www.lingholic.com/ how-many-words-do-i-need-to-know- the-955-rule-in-language/ -learning-part-2.

[7] Manythings.org, "English vocabulary," http://www.manythings. org/vocabulary/lists/l/.

[8] J. S. Culpepper, G. Navarro, S. J. Puglisi, and A. Turpin, "Top-k ranked document search in general text databases," in Proc. Annu. Euro. Conf. Algorithms (ESA), Sep. 2010, pp. 194–205.

[9] R. cloud computing, "Openstack cloud software," http://www.openstack.org.

[10] VOANEWS.COM, "Voice of american," http://www.voanews.com.

[11] D. Huang, "Mobile cloud computing," IEEE COMSOC Multimedia Commun. Tech. Committee (MMTC) E-Letter, vol. 6, no. 10, pp. 27– 31, 2011.

[12] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in Proc. Int. Conf. Comput. Commun. (INFOCOM), Apr. 2011, pp. 829–837.

[13] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data,"IEEE Trans. Parallel Distrib. Systems, vol. 23, no. 8, pp. 1467–1479, 2012.

[14] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS), Jun. 2010, pp. 253–262.

[15] C. Gentry and S. Halevi, "Implementing gantries fully homomorphicencryption scheme," in Advances in Cryptology– EUROCRYPT 2011, 2011, pp. 129–148.

[16] C. O¨ rencik and E. Savas¸, "Efficient and secure ranked multi-keyword search on encrypted cloud data," in Proc. Joint EDBT/ICDT Workshops, Mar. 2012, pp. 186–195.

[17] Gartner, "Worldwide traditional pc, tablet, ultra mobile and mobile phone shipments on pace to grow 7.6 percent in 2014," http://www.gartner.com/newsroom/id/2645115.

[18] Trellian, "Keywords number," http://www.keyworddiscovery. com/keyword-stats.html?date=2014-03-01.

[19] V. Rijmen and J. Daemen, "Advanced encryption standard," Federal Information Processing Standard, pp. 19–22, 2001.

[20] X. Lai, "On the design and security of block ciphers," Ph.D. dissertation, Diss. Techn. Wiss ETH Z¨ urich, Nr. 9752, 1992. Ref.: JL Massey; Korref.: H. B¨ uhlmann, 1992.

[21] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in Proc. ACM

SIGMOD Int. Conf. Manag. Data (COMAD), Jun. 2004, pp. 563–574.

[22] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L. Varna, S. He, M.Wu, and D.W. Oard, "Confidentiality-preserving rank-ordered search," in Proc. ACM Workshop Storage Secur. Survivability (StorageSS), Oct. 2007, pp. 7–12.

[23] A. Boldyreva, N. Chenette, Y. Lee, and A. Oneill, "Order preserving symmetric encryption," in Advances in Cryptology- EUROCRYPT 2009, 2009, pp. 224–241.

[24] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[25] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in Advances in Cryptology-EUROCRYPT 2010, 2010, pp. 24–43.

[26] D. Stehl´e and R. Steinfeld, "Faster fully homomorphic encryption," in Advances in Cryptology-ASIACRYPT 2010, 2010, pp. 377–394.

[27] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE Symp. Secur. Priv. (SSP), May. 2000, pp. 44–55.

[28] E.-J. Goh et al., "Secure indexes," IACR Cryptology ePrint Archive, vol. 2003, p. 216, 2003.

[29] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Advances in Cryptology- Eurocrypt 2004, 2004, pp. 506–522.

BIOGRAPHIES

J. Santhana Mani was received his Master of Science in Information Technology from Anna University, India and he is currently studying the Master of Technology in Information Technology from Vinayaka Missions University, India.

S. Karthik was received his PhD degree in the area of Communication and Networking at Anna University, India in 2015. He is currently an associate professor in the department of Information Technology at the VMKV Engineering College, Salem, India. His research interests focus on protocols to improve Internet performance and improving existing ones.

K.Sasikala received the PhD degree in the faculty of Information and Communication Engineering; in the domain of wireless networks from Anna University, Chennai, Tamil Nadu, and India in 2015 She is working as an Associate Professor, Department of IT, V.M.K.V Engineering College, Salem, Tamil Nadu, India.