Cryptographically Imposed Model for Efficient Multiple Keyword-based Search over Encrypted Data in Cloud by Secure Index Using Bloom Filter and False Random Bit Generator

Devi Thiyagarajan, R. Ganesan

(Corresponding author: Devi Thiyagarajan)

School of Computing Science and Engineering, VIT University
Chennai Campus, Vandalur - Kelambakkam Road Chennai - 600 127, India
(Email: devi.t2013@vit.ac.in)

(Received Feb. 11, 2016; revised and accepted May 7 & June 5, 2016)

Abstract

Resources such as storage and network as a service to organizations and customers in reduced cost by cloud computing. The documents stored in cloud are of huge size and due to its sensitivity arises security and storage problems. Data is stored on remote location in cloud and cryptographic techniques resolve problems of data security. To ensure data to be secure in cloud, the client encrypts data and then stores on cloud by employing Hyper Elliptic Curve cryptography (HECC). Search over encrypted data makes the process of file retrieval from cloud more difficult. The paper proposes architecture of multiple keyword search by building index using Bloom filter and also pair key generation by false random bit generator. Bloom filter takes constant time for searching O (N) on large encrypted file systems without the need of document decryption thereby speeding up the process of ciphertext retrieval on user side. Bit array representing keyword information is only stored by data owner on cloud where the server is unable to find file content or query information. The experimental results show that Bloom filter based indexing is faster than traditional indexing schemes, the multiple keyword based search algorithm is effective in case of the response time of query and scalability of the system in case of size of data.

Keywords: Bloom filter, false random bit generation, HECC, secure index

1 Introduction

Cloud computing is one of the emerging field which replaces the burden of IT industry from spending huge expenditure on resources such as storage and network. Remote storage and easy accessibility of data combined with characteristics such as on-demand self-service, broad network access, resource pooling, rapid elasticity and measured services. Cloud is deployed as public, private, hybrid and community cloud with service delivery models such as SaaS (Software-as-a-Service), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). Resources in cloud are offered to both industries and individuals.

Though, cloud has many advantages certain issues such as security, privacy and interoperability do exist. The driving force for cloud computing is virtualization which enable multiple virtual machines to run with the help of single physical machine. As the infrastructure is being shared by several VMs, security issues do arise. Various surveys conducted portray security as one of the major challenge in cloud environment. Latest report on cloud computing challenges compares the issues and portrays that security stands as the first challenge.

Data confidentiality and privacy issues do arise due to multi-tenancy characteristic of cloud. To protect clients from such issues, cloud service provider need to follow certain mechanisms to keep data safe. But, malicious insider may act on behalf of the provider and send indelicate data. Such a situation makes a clear point that the security models cannot be build based on the trust of provider. Client needs to protect their data from malicious attacks both externally and also from internal adversaries. Clients outsource their files containing sensitive information to cloud for effective retrieval at the necessary time. Google search allows search over plaintext data. Such data is being stored in plaintext form in the cloud service which is vulnerable to attacks by adversaries. Protection of data from such malicious activities is prevented by storing encrypted data on cloud. Encryption methods are classified as symmetric (AES) [11, 21] and asymmetric (HECC) encryption algorithms [17, 23]. HECC is proven to be the robust method of encrypting and decrypting files due to the hardness of hyperelliptic discrete logarithm problem. Incorporating such an algorithm in cloud enhances the data security in cloud environment.

Data owners share their files with authenticated users through retrieval mechanisms. Traditional methods retrieve the entire collection of files for a single search request from data user. This incurs more time for search reply and wastage of bandwidth. Selective retrieval of file based on data user request makes use of keywords. Search of plaintext data is not suited for cloud environment and encryption of data limits the search capability. Searchable encryption allows building an index with keywords and corresponding documents. Trapdoors along indexing enable data users to search over encrypted data in a secure manner and maintains privacy of document information as well as keywords. Such techniques remain unsuitable for cloud scenarios due to employment of symmetric encryption methods along with single keyword search. With these pitfalls it is necessary for an effective mechanism to download encrypted documents from cloud. The proposed model (Figure 1) incorporates asymmetric encryption such as HECC and position-based multiple keyword search scheme to maintain data security and privacy in cloud.

Retrieval of files from encrypted content need to be given attention as delivering the correct content to registered user is the ultimate goal. Search over encrypted data has to be done in an efficient way to reduce the overhead experienced by the users while decrypting their content from cloud. Hyperelliptic curve cryptography is employed for encryption of documents and search over encrypted files is carried by Bloom filter (BF) and False Random Bit Generator (FRBG).

Our contributions for providing data security and privacy includes:

1) Key pair generation by False Random Bit Generator (FRBG);

2) Building safe index using Bloom filter;

3) Search on encrypted files using Bloom Filter. Huge numbers of experiments were conducted to analyse the efficacy of the proposed architecture.

Paper organization is as follows: Literature review is presented in Section 2. Section 3 describes the architecture of proposed scheme. Section 4 discusses security analysis, Section 5 deals with implementation results and Section 6 concludes the paper.

2 Related Work

Searchable encryption [2, 3, 4, 5, 6, 7, 10, 27, 29] plays a major role in cryptography. Two-layer encryption of every keyword in a file is the first work on searchable encryption [27]. Usage of Bloom filter for construction of index of files along with trapdoor is stored on server. The request for search is followed by a trapdoor construction and the server on other hand tests with Bloom filters and sends the identifiers of files as response [10]. For effective search, one encrypted index was built for entire document collection where every entry in index contains keyword trapdoor along with file identifiers in encrypted state [6, 7]. Public-key based searchable encryption [4] played a major role in retrieval systems for privacy. Public key users store data on servers whereas authenticated users with the private key perform the operation of search. For efficient querying purpose conjunctive search [1], fuzzy keyword search [20, 31] and similarity keyword search [26] were proposed. A widely accepted retrieval technique namely private information retrieval [22] also helps in retrieving the items incurring complex computations.

Usage of k-nearest neighbour algorithm for searching documents along with ciphertext policy attribute-based encryption provides security and privacy for data [19, 30]. Hourglass function was utilised for the purpose of verifying the encrypted files [14] and identity based encryption is also employed for attaining data privacy [16]. Elgamal encryption [12] supporting fuzzy keyword search over encrypted data prevents inconsistencies occurring during search [28]. Ranked keyword search returned top-k files by using multiple keywords along with homomorphic encryption [32]. Attacks against ciphertexts are being analysed keyword search in cloud environment [15]. Hierarchical predicate encryption [24] along with access control also achieved keyword based search over encrypted data. The problems in bog specific search engines have been identified to optimize the search mechanism in cloud [25].

Authenticated users [13] retrieve files from cloud storage as authentication plays a major role in preventing illegitimate user access due to the application of discrete logarithms [18]. Hyperelliptic curve cryptography is combined with Advanced Encryption Standard and MD-5 algorithm in digital envelope for securing e-commerce channel [8]. The usage of HECC in cloud enhances the security of sensitive data by preventing from exposure.

A notable point is that all the existing schemes lack in certain functionality and remains unsuitable for cloud environments. Our work focusing on retrieving the exact document from cloud server based on Bloom filter and false random bits add robustness to the data security framework.

3 Construction of Secure Keyword Search Scheme

3.1 Background

1) **False Random Bit Function.**
$F : \{0,1\}^{Fi} \times \{0,1\}^{Fj-n} \rightarrow \{0,1\}^n$ be a false random function where sequence of F$_i$ bits key is taken along with random F$_j$-n bit string and mapped to random n-bit string which is publicly known to all users.

2) **Bloom Filter.**

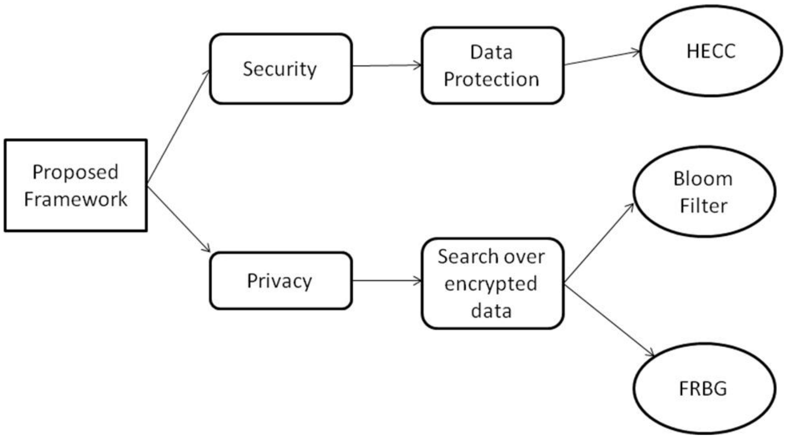

Figure 1: Proposed framework

A data structure named Bloom filter is used to identify the membership of a file in the file collection F and is used for the purpose of querying. A Bloom filter consists of set of 'm' elements $S_t = \{x_1.\ x_2, \cdots, x_m\}$ with 'n' hash functions $H_f = \{h_1, h_2, \cdots, h_n\}$. The r-bit Bloom filter is initially set to 0. Every element $x_k \in S$, $k = 1, 2, \cdots, m$, $h_i(x_k)$ for $i = 1, 2, \cdots, n$ is attained where $0 \leq$ h$_i$(x$_k$) $<$ r thereby respective bits to hash index is set for element $BF[h_i(x_k)]$. With input 'a', 'n' hash indices $h_i(a)$ for $i = 1, 2, \cdots, n$ is attained where $0 \leq h_i(a) < r$. Usage of hash functions may provide a positive answer while querying for an element which may not be the member of set [9]. Such condition is called as false positive.

3) **Hyperelliptic Curve Discrete Logarithm Problem.**
Given hyperelliptic curve of genus g over finite field F_q, point $P \in JC(K)$ of order n, point $Q\varepsilon < P >$, obtain integer $l\varepsilon[0, n-1]$ so $Q = lP$ where l is integer and is discrete logarithm of Q to base P, represented as $l = \log QP$.

4) **Multiple Keyword Search Scheme.**
With set of encrypted files C, MKS scheme returns the file identifiers (file_id) of those requested files to authenticated users. Multiple keyword set along with indexing (ID_{BF}) by Bloom filter speeds up the retrieval process of the users.

5) **Trapdoors.**
By the application of one-way hash functions, trapdoors are generated. With secret key (s_k) and keyword KW_i, trapdoor of KW_i is calculated as $T_{KWi} = f(s_k, KW_i)$.

3.2 System Model

Three participants of model are client (upload), cloud server (storage) and data users (file retrieval). Set of files after encryption by HECC ($C = C_1, C_2, \cdots, C_n$) are stored in cloud server along with keyword set $KW = \{KW_1, KW_2, \cdots, KW_{mo}\}$. Users registered with clients are authenticated and provided with the access over encrypted files C. In order to retrieve selective files, data user provides the multiple keywords of interest. The mapping between the search request from multiple users and files is the responsibility of CS as every file gets indexed with an unique file identifier (ID_f) and set of keyword. Multiple keyword based search scheme proceeds files containing specified keywords as the result of search to the authenticated users. Figure 2 depicts construction of multiple keyword search scheme.

3.3 Construction of Multiple Keyword Search Scheme

The multiple keyword search scheme performs the following three steps:

1) publicly known false random bit generation (FRBG) algorithm for pair_ key generation

2) index building using Bloom filter IB_{BF} with keyword set and

3) index search algorithm IS_{BF}.

The steps involved in the proposed scheme are:

1) Client sends the set (0, 1) along with files and keywords to FRBG.

2) Authenticated users send set (1, 0) to FRBG.

3) If the two different random bit sets matches, it returns the pair_ key and sets the bit as 1 else rejected.

4) Role of FRBG is string matching along with bit (0, 1) sets.

5) The cloud server CS contains the file identifier (id), bit matching and keywords.

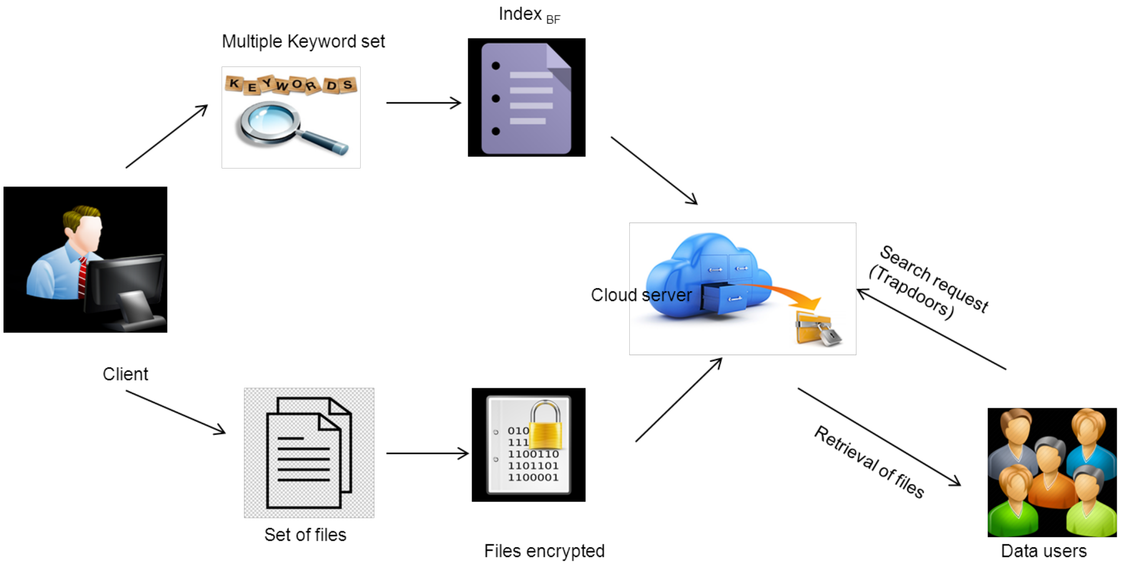

Figure 2: Construction of multiple keyword search scheme

6) Client runs the IndexBuild algorithm to build the secure index with Bloom filter and stores in server.

7) User on requesting the file, server runs the IndexSearch algorithm and in response returns the corresponding file in a secure manner.

Publicly Known False Random Bit Generation (FRBG).
The algorithm takes set of files F$_i$, message given by client which is used for mapping with strings of random bits (0, 1) and 8-bit key as input (See Algorithm 1). Output generated by the algorithm is sequence of false random bits $(y_1, y_2, \cdots, y_n)$. Suppose the input is in collection of files, it is followed by key pairing for client acknowledgement. The string along with the message is processed. Suppose if the paired key matches with the string, sequence of false random bits $(y_1, y_2, \cdots, y_n)$ are generated. By taking two different sets $\{0,1\}^s$ ant $\{1,0\}^{n-m}$ mapping is done. If mapping is correct, positions of bit are set to 1, else 0. Finally $y_{ij} = 1$ position of bit is set to 1 And false random numbers are generated (Figure 3).

Index Building with Bloom Filter.
Followed by pair_ key generation, secure index is constructed by means of Bloom filter. Formalization of the proposed scheme is as follows:

- Pub$_{key}$ (P_k): Represents the public key of the given user (i.e) $P_k \in \{0,1\}^{Fi}$ which is kept publicly known.

- Secret Key (s_k): Represents the secret key for string matching which is to be kept known only to authenticated users.

- False Random Bit Function: $F : \{0,1\}^{Fi} * \{0,1\}^{Fj-n} \rightarrow \{0,1\}^n$ be false random function where sequence of F$_i$ bits key is taken along with random F$_j$-n bit string and mapped to string of random n-bit which is widely identified by all users.

Algorithm 1 Publicly Known False Random Bit Generation (FRBG)

```
Input: Collection of files $F_i$, message $m$, pair_key;
Output: Set of false random numbers $(y_1, y_2, \cdots, y_n)$;
for $(x \in F_i U$ pair_key)
Generate String $S_i$ message;
do (pair_key = = string)
Generate false random bits $y_{ij}$;
if $(F : \{0,1\}^{Fi} * \{1,0\}^{Fj-n} \rightarrow \{0,1\}^n)$ then
    while $(y_{ij} ==$ matching done) do
        return as 1;
    end while
else
    return 0;
end if
End
```

- Trapdoor: Let T$_{ij}$ be a trapdoor whose inputs are public key (P_k), secret key (s$_k$) and keywords (KW) and generates the trapdoors for files F$_i$. T$_{ij}$ $(P_k$, KW) = E$_{Pk}$ (KW) where E is encryption function. Encryption is performed by Hyperelliptic Curve Cryptography for security reasons.

- IndexBuild $_{BF}$ (F$_i$, P_k, KW, H): The algorithm takes files F$_i$, pubic key P_k, keywords KW and hash functions H as input. It generates string using false random bit generator (FRBG) and outputs the index for file F. Finally index is built for searching files (F$_i$) by using the sequence of random numbers generated by FRBG algorithm.

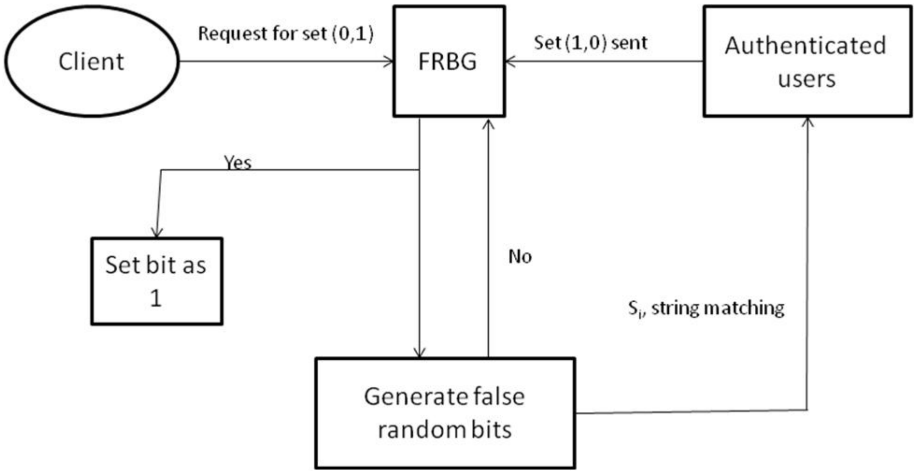

Figure 3: Pair_key generation by FRBG

- IndexSearchBF $_{BF}$ (IB_{BF}, T, key_pair, H): With the index built by using Bloom filter, trapdoor T, key_pair generated by FRBG and hash functions H the algorithm IndexSearch$_{BF}$ outputs the file containing the desired keyword KW.

  Finally index is built for searching files (F$_i$) by using the sequence of random numbers generated by FRBG algorithm (Figure 4).

Hash functions used in Bloom filter are publicly known thereby revealing the contents of files. The proposed scheme allows hash to be not applied directly on keywords KW but indexes each keyword KW by its encrypted version $E_{Pk}(KW_i)$. Hence, Bloom filter is built with hash values $H_f(E_{Pk}(KW_i))$, $f = 1, 2, \cdots, n$. The positions of bits in BF which are positioned to 1 equivalent to KW are similar for each and every file that contain KW. Due to vulnerability of frequency-based attacks, file identifier (ID) is used. $H_f(E_{key\_pair}(id(F_i), T(KW_i)))$, $f = 1, 2, \cdots, n$ is calculated as the hash function for Bloom filter. Suppose if the same keyword occurs in several different files, only the file with priority is retrieved. An attacker can find the specified file only if the trapdoor T is offered. The algorithm for index building using Bloom filter (IB_{BF}) is in Algorithm 2.

Algorithm 2 Index Building using Bloom Filter (IB_{BF})

```
Input: Set of Files ($F_i$), public key $P_k$, keywords KW, $H(h_1, h_2, \cdots, h_n)$;
Output: $IB_{BF}$ /* Index built for files using bloom filter */;
if ($BF_{Fi} \neq empty$) then
    for all $KW_i \in F_i$ do
        Generate trapdoor $T_{ij}(P_k, KW_i) = E_{Pk}(KW_i)$;
        Generate string-matching $S_i = E_{key\_pair}(id(F_i), T(KW_i))$;
        for $f = 1$ to $n$ do do
            Calculate pos_bit ($Pb_f$) = $H_f(S_i)$;
            Set $IB_{BF}[Pb_f] = 1$;
        end for
    end for
    Return $IB_{BF}$;
end if
End
```

Each file in file collection F_i is assigned an unique identifier ID. The index constructed maps keywords to ID of file that actually contains the KW, thereby reducing search complexity. This index is then encrypted and stored on cloud. Search on inverted index with KW returns all the id of files containing the KW without looking into the original file collection F.

3.4 Index Search Algorithm IS_{BF}

Search with multiple keywords is done in the following manner. Suppose the user needs to search for keyword KW, trapdoor $(KW_i) = E_{Pk}(KW_i)$ is sent to cloud server CS. CS runs the $IndexSearch_{BF}$ algorithm on each file F in the set and returns the specified ones (See Algorithm 3.

Algorithm 3 Index Search IS_{BF}

```
Input: Index built with Bloom filter $IB_{BF}$, Trapdoor $T(KW_i)$, key_pair, $H(h_1, h_2, \cdots, h_n)$;
Output: Specified file F or $\phi$;
Compute $y = E_{key\_pair}(id(F), m(KW_i))$
for $f = 1$ to $n$ do do
    if $IB_{BF}[H_f(y)] \neq 1$ then then
        Return $\phi$;
    end if
end for
Return $F$;
End
```

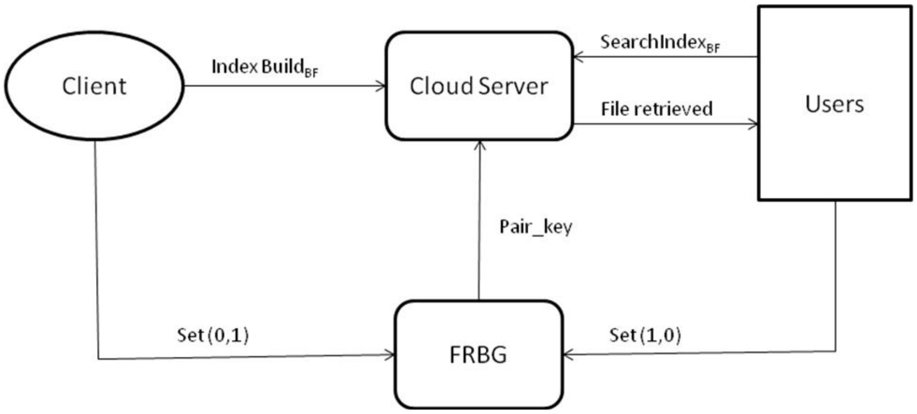

Figure 4: Secure search scheme with bloom filter and FRBG

4 Security Analysis

The security analysis of the proposed model is discussed.

Theorem 1. *Given the collection of files F_1, F_2, F_3, F_n, if $F_1 \in F_n$, then perform index building using Bloom Filter and also retrieve false random bits with FRBG. For the proof of this theorem, we have n number of files with index building $IndexBuild_{BF}(F_i, P_k, KW, H) = TrapdoorT_{ij}(P_k, KW_i) = E_{Pk}(KW_i) + F : \{0,1\}^{Fi} * \{0,1\}^{Fj-n} \rightarrow \{0,1\}^n$. Since the hash value position is greater than 0 (i.e) 1, if the theorem generates false random bits and the cryptographic algorithm over secured access using Bloom filter is preserved. To measure the time taken for index building IB_{BF} and index search IS_{BF} employing Bloom filter is $O(n)$ which is much less compared to brute-force approach. As shown in Figure 5, X-axis represents time taken for search in milliseconds and Y-axis represents file size in KB.*

Claim 1: Use of Bloom filter imposes reduced false positives.
The presence or absence of a file in file collection is straightforwardly verified by usage of Bloom filter. False positives may occur as a result of querying for a file which may not be in the collection. Since every related bit in BF is positioned to 1 for hash indices, there are less chances of false positives thereby reducing them totally.

```
$BF \in \pm 1|0\{$
    $If\ pos\_1:\ values$
        $Generate\ bits\ FRBG$
    $Else$
        $Return\ pos\_0\}$
```

Claim 2: The proposed scheme prevents remote attacks.
An adversary can obtain the information stored remotely on servers through remote attacks. The proposed scheme employs FRBG for key matching process and bloom filter for indexing and searching process. $F : \{0,1\}^{Fi} * \{0,1\}^{Fj-n} \rightarrow \{0,1\}^n$ provides security through matching the perfect keyword and usage allowed only for authenticated users. These schemes are proven to be robust and provide two level security for data storage and retrieval on cloud. Thus an adversary is prevented by obtaining access on the server information and remote attacks are averted.

Claim 3: The proposed scheme is secure with the usage of Bloom filter and false random bit generation function.
Two major variants employed in search over encrypted data are Bloom filter and false random bit generation function. Pair_key generated by FRBG secures the model by authenticating users ($T_{ij}(P_k, KW_i) = E_{Pk}(KW_i)$) and the usage of Bloom filter enhances the search process to be performed in minimum time $O(n)$.

Claim 4: Search over encrypted file is faster than any other schemes with reduced search time.
Bloom filters speed up search over encrypted data in cloud as availability of necessary data in the right time is one of the most important feature in cloud computing. Assume there are 'n' number of files to be stored on remote server $F_1, F_2, \cdots, F_n$ containing 'm' keywords. Search time by applying brute force method is $O(n * m)$. By using 'n' number of bloom filters, the search time is reduced as $O(n)$. To check whether the respective bit in BF is set to 1, it takes only constant time thereby reducing the time for search.

5 Implementation Results

The proposed model along with multiple keyword search scheme is implemented in Openstack for effective results. When compared with traditional techniques, the proposed framework worked faster with less time complexity (Figure 5). X-axis represents the time taken to retrieve

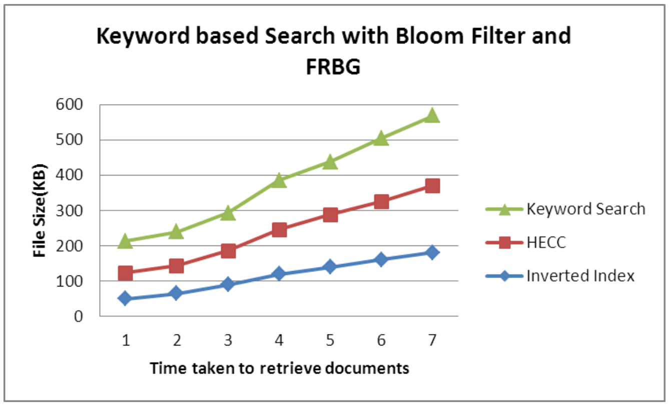

Figure 5: Comparison of proposed scheme with inverted index

documents from cloud server and Y-axis represents the file size uploaded on cloud server. With the increase in file size, the search time also reduces for our proposed scheme and vice-versa. Compared with HECC, inverted index incurs additional time for retrieving files. Using HECC algorithm, Bloom filter and false random bit generator the time taken for searching the document is less compared to existing works. When file size increases, Bloom filter has 0% false positive rate. When the data size increases, space occupied by inverted index much more compared to Bloom filter. Applications supporting data sets of larger size can prefer Bloom filter instead of inverted index allowing minimum amount of false positive rate.

Efficiency of retrieved files:

Let N_{kw} be the number of documents that contain keywords KW and N_Q be number of documents returned as a result of a query Q. The efficacy of retrieved files is given by the formula

$$E = N_{kw}/N_Q.$$

Such overhead of files retrieved presents the amount of unused files and the percentage is not high in case of the proposed model.

6 Conclusion

The proposed model achieves data security and privacy. The utilization of HECC for encryption/decryption purpose, false random bit generator for pair_key generation and Bloom filter for index building and searching files in cloud enables the proposed model to be efficient. Faster encryption and decryption time is achieved by HECC and lesser search time for retrieving encrypted files from cloud is attained by FRBG and Bloom filter add advantages to the data security model. Minimal key size and the hardness of discrete logarithmic problem of HECC prevent the adversaries from attacking the model. Two-layer protection with techniques such as FRBG and Bloom filter avoids unauthorized users from retrieval of data from cloud servers. The future work is focused on providing security to data-at-rest and also ranking the search results based on relevance.

References

[1] M. Azraoui, K. Elkhiyaoui, M. Onen and R. Molva, “Publicly verifiable conjunctive keyword search in outsourced databases,” in *IEEE conference on Communications and Network Security (CNS’15)*, pp. 619–627, 2015.

[2] F. Bao, R. Deng, X. Ding, and Y. Yang, “Private query on encrypted data in multi-user settings,” in *Information Security Practice and Experience*, LNCS 4991, pp. 71–85, Springer, 2008.

[3] M. Bellare, A. Boldyreva, and A. ONeill, “Deterministic and efficiently searchable encryption,” in *Annual International Cryptology Conference*, pp. 535–552, Springer, 2007.

[4] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in *Advances in Cryptology (EUROCRYPT’04)*, LNCS 3027, pp. 506–522, Springer, 2004.

[5] D. Boneh and B. Waters, “Conjunctive, subset, and range queries on encrypted data,” in *Theory of Cryptography Conference*, pp. 535–554, 2007.

[6] Y. C. Chang and M. Mitzenmacher, “Privacy preserving keyword searches on remote encrypted data,” in *International Conference on Applied Cryptography and Network Security*, pp. 442–455, 2005.

[7] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.

[8] R. Ganesan, M. Gobi, and K. Vivekanandan, “A novel digital envelope approach for a secure e-commerce channel,” *International Journal of Network Security*, vol. 11, no. 3, pp. 121–127, 2010.

[9] W. R. Ghanem, M. Shokir, and M. Dessoky, “Defense Against Selfish PUEA in Cognitive Radio Networks Based on Hash Message Authentication Code,” *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 12–21, 2016.

[10] E. J. Goh, *Secure Indexes*, Cryptology ePrint Archive, Report 2003/216, 2003. (`http://eprint.iacr.org/`)

[11] T. Gulom, “The encryption algorithm AES-PES16-1 and AES-RFWKPES16-1 based on network PES16-1 and RFWKPES16-1,” *International Journal of Electronics and Information Engineering*, vol. 3, no. 2, pp. 53–66, 2015.

[12] M. S. Hwang, C. C. Chang, K. F. Hwang, “An ElGamal-like cryptosystem for enciphering large

messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445–446, 2002.

[13] M. S. Hwang and C. Yu Liu, "Authenticated encryption schemes: current status and key issues," *International Journal of Network Security*, vol. 1, no. 2, pp. 61–73, 2005.

[14] K. Hu and W. Zhang, "Efficient verification of data encryption on cloud servers," in *2014 Tewlfth Annual Conference on Privacy, Security and Trust (PST'14)*, pp. 314–321, 2014.

[15] F. G. Jeng, S. Y. Lin, B. J. Wang, C. H. Wang, T. H. Chen, "On the security of privacy-preserving keyword searching for cloud storage services," *International Journal of Network Security*, vol. 18, no. 3, 2016, pp. 597–600, 2016.

[16] T. Jung, X. Y. Li, Z. Wan and M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 190–199, 2015.

[17] A. V. N. Krishna, A. H. Narayana, K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.

[18] C. C. Lee, M. S. Hwang, Li H. Li, "A new key authentication scheme based on discrete logarithms," *Applied Mathematics and Computation*, vol. 139, no. 2, pp. 343–349, 2003.

[19] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. Shen, "Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage," *IEEE Transactions On Emerging Topics in Computing*, vol. 3, no. 1, pp. 127–138, 2015.

[20] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling efficient fuzzy keyword search over encrypted data in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1467–1479, 2012.

[21] A. Mersaid, T. Gulom, "The encryption algorithm AES-RFWKIDEA32-1 based on network RFWKIDEA32-1," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 1–11, 2016.

[22] R. Ostrovsky, *Software Protection and Simulations on Oblivious Rams*, Ph.D. Dissertation, Massachusetts Institute of Technology, 1992.

[23] K. R. Santosh, C. Narasimham, and P. Shetty, "Cryptanalysis of multi-prime RSA with two decryption exponents," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 40–44, 2016.

[24] Z. Shen, J. Shu and W. Xue, "Keyword search with access control over encrypted data in cloud computing," in *International Symposium of Quality of Service (IWQoS'14)*, pp. 87–92, 2014.

[25] J. Singh, "Cloud based technique for blog search optimization," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 32–39, 2016.

[26] M. Strizhov and I. Ray, "Multi-keyword similarity search over encrypted cloud data," *IFIP International Information Security Conference*, pp. 52–65, Springer, 2014.

[27] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 44–55, 2000.

[28] Y. Wang, W. Bao, Y. Zhao, X. Hu, and Z. Qin, "An ElGamal encryption with fuzzy keyword search on cloud environment," *International Journal of Network Security*, vol. 18, no. 3, pp. 481–486, 2016.

[29] B. Waters, D. Balfanz, G. Durfee, and D. Smetters, "Building an encrypted and searchable audit log," in *Proceedings of 11th Annual Network and Distributed System*, 2004.

[30] H. Xu, S. Guo and K. Chen, "Building confidential and efficient query services in the cloud with rasp data perturbation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 1–18, 2014.

[31] Q. Xu, H. Shen, Y. Sang, H. Tian, "Privacy-preserving ranked fuzzy keyword search over encrypted cloud data," in *International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'13)*, pp. 239–245, 2013.

[32] J. Yu, P. Lu, Y. Zhu, G. Xue and M. Li, "Towards secure multi-keyword top-k retrieval over encrypted cloud data," *IEEE Transactions On Dependable And Secure Computing*, vol. 10, no. 4, pp. 239–250, 2013.

Biography

Devi Thiyagarajan is pursuing Ph.D in computer science from VIT University. Her areas of interest include cloud computing and cloud data security.

R. Ganesan is working as Associate Professor in VIT University. His area of expertise includes wireless sensor networks, cryptography and network security and cloud security. He has published several papers in international journals and conferences.