International Conference on Advances in Materials, Computing and Communication Technologies
In Association with International Journal of Scientific Research in Science and Technology
Volume 9 | Issue 1 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)

Maintaining Confidentiality with Searchable Encryption Process Using Metadata Keys

Anusha B.[1], Daphney Joann J.[2]
[1]Department of CSE , Global Institute of Engineering and Technology, Melvisharam, Tamil Nadu, India
[2]Assistant Professor, Department of CSE , Global Institute of Engineering and Technology, Melvisharam, Tamil Nadu, India

ABSTRACT

Searchable Encryption process encourages the server present in the cloud to search encoded information without decoding it. Cloud computing encourages the process of storing, accessing and retrieving the data's and programs over the internet without any computer hardware. In this paper, we discussed about how to maintain confidentiality of the data's and information which are present in the cloud. Cloud encourages encryption process in a unique manner in which the exchanges of data's are done using a single encryption key. We use an algorithm called PSE algorithm (Privacy Preserving Scheme with SHA-12) which is a symmetric block cipher algorithm that takes plain text in 128 bits and converts them to cipher text using keys 128,192 and 256 bits respectively. PSE uses a technique called SHA-12 which are being designed to keep data in a secured format and it does its function in bitwise operation. Here it is mainly used to generate Meta data keys for the files and documents which contain sensitive information's. Simulation Results Shows the Search Operation of Keywords and Trapdoor performance.

Keywords: Searchable Encryption, Cloud Security, Privacy Preserving Scheme, Secure Hash Algorithm, Attribute Based Encryption; Anonymity.

I. INTRODUCTION

The outsourced data may contain sensitive information, such as financial records of an individual or an organization, bids information submitted for a tender, Personal Health Records (PHRs) and so on, where the data can allow the cloud server or unauthorized users to access and/or infer sensitive information. To address the issue of data privacy and access control, one practical solution is to encrypt the documents before outsourcing them on cloud storage server. Let us consider the applications where multiple data owners use the public cloud storage services to upload their encrypted documents and multiple users can access. The documents stored on the cloud storage server. In such applications, applying fine-grained access control policy will enable intended security control on document access one reasonable arrangement is to encode the archives before redistributing them on cloud capacity server. Gives us a chance to think about the applications where numerous information proprietors utilize the open distributed storage administrators to transfer their encoded records and different clients can get to the reports put away on the distributed storage server. In such applications, applying fine-grained access control strategy will empower expected security control on record get to. Characteristics Based Encryption (ABE) is an intriguing cryptographic calculation that gives classification of information along with proper authorized fine-grained access control. The utilization of single-client accessible symmetric encryption conspires in such situation isn't a compelling component, as the patient requires to encode his medical reports with his mystery key and afterward share this mystery key with specialist. In this way, multi-client accessible symmetric encryption plans are best, as the necessity of inquiry over encoded information and authorizing access control arrangement, where an information proprietor for example, patient can create the common mystery key or search token from his lord mystery key and issue them to the approved clients for looking over encoded information.

Attribute Based Encryption (ABE)[1] is an interesting cryptographic algorithm that provides confidentiality of data along with owner-enforced fine-grained access control. With encrypted data, one of the required functionality, searching over database, is a practical requirement of modern cloud storage data. In the area of searchable encryption, various techniques for searching over encrypted data have been introduced. Searchable encryption can help a receiver to securely and selectively retrieve the data from public cloud, which is of user's interest and which is accessible to user. For example, a doctor wants to search for all the records of his patients who have been detected with chronic kidney disease and for which doctor has been provided the access rights to patients medical records, where each report is encrypted and uploaded by the patient. The application of single-user searchable symmetric encryption schemes in such scenario is not an effective mechanism, as the patient requires to encrypt his medical reports with his secret key and then shares this secret key with doctor. Therefore, multi-user searchable symmetric encryption schemes [6-8] are most effective, as the requirement of search over encrypted data and enforcing access control policy, where a data owner such as patient can generate the shared secret key or search token from his master secret key and issue them to the authorized users (e.g., doctor in our example for searching over encrypted data. Although these schemes can work in single-sender multi-receiver setup, they cannot perform well in multi-sender multi-receiver scenario, because each data sender has to communicate with each of the data receiver in a secure manner to issue the search token or secret key, which will cost large communication overhead. As an alternative, keyword based searching over attribute based encrypted data fulfills both the objectives of searching over encrypted data and enforcing fine-grained access control policy. ABE scheme works well in multi-sender and multi-receiver scenario. It does not require a direct interaction between data owner and data receiver. Many schemes on keyword-based searching over attribute based encrypted data have been proposed in [9-12] for providing single keyword search capability over single encryption based attribute data. Wang et al,s scheme [9] provides single keyword based searching along with partial decryption task delegated to the cloud service provider. The scheme in [10] provides the verified search results in addition to searching over encrypted data. The scheme in [11] provides a solution that addresses the issue of data sharing and keyword update in addition to keyword-based seaech operation over encrypted data. The scheme in [13] provides the disjunctive multi keyword search facility. Dong et al [14] presented a scheme that provides an efficient keyword-based searching operation over ABE via an online-offline approach. The scheme in [15] includes proxy re-encryption (PRE) and a secret sharing scheme (SSS) into ABKS. However, in all these schemes the access control policies are in clear form. To preserve the data confidentiality and/or the purpose of the service enquired for protecting user privacy is an important requirement [2-5]. There exists many attribute based keyword-searchable encryption schemes [9-15] in literature. However, these schemes do not address the important issue of receiver anonymity. Koo et al [16] proposed an author-based search over encrypted data anonymously. However their scheme is found insecure in [17]. There are a few schemes [18,19], which provide keyword based searching over attribute based encrypted data with hidden access policy. Shi et al [18], have presented a scheme, authorized searchable public-key encryption (ASPKE) in which a data owner decides the access policy for his encrypted data and keeps it hidden inside cipher text. The AS-PKE scheme uses the access structure in form of LSSS. If there are n attribute fields in the system, then only one value per attribute can be placed in the scheme to preserve the receiver anonymity. This is less effective when a cipher text has multiple recipients with different attribute values for one attribute. Wang et al [19] proposed a scheme that supports only one value per attribute in the cipher text access policy, where the access structure in formed with AND gate on multivalued attributes,

II. CLOUD COMPUTING

According to the National Institute of Standards and Technology (NIST, Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, application, and services) that can be rapidly provisioned and released with minimal management effort

III. MAIN CHARACTERISTICS

The document lists five main characteristics of Cloud Computing model:

- On-demand, self-service. Cloud users add resources at any time without system administration intervention.
- Broad network access. Cloud users access resources via network
- Resource pooling. Cloud providers heavily utilize virtualization and resource pooling to exploit an economy of scale
- Rapid elasticity. Cloud users have the capability to scale resources to satisfy a fluctuating demand
- Measured services. Resources are monitored to allow cost optimization and performance analysis

A. MODELS

Infrastructure as Service: This model allows the user controlling a full stack of software from thr hardware to the applications. It gives the highest flexibility, but requires more expertise from the user. An example of Iaas provider is Amazon Web Services

Platform as a service: In this model users obtain a platform on top of which they build software. Providers take care about low-level details such as availability, security patches, scalability of resources. Some examples of PaaS providers include Heroku1 or Google App Engine

Software as a Service: This model includes all the software provided to the users as a service. This broad category ranges from customer relationship management to chat services.

Researchers argues that to differentiate service modules is misleading. Since all of the three modules refer to computing provided as a service, the authors prefer to use the more generic term utility computing. Utility Computing is characterized by properties such as flexibility, portability and ease of use. Authors describe two main actors involved in cloud computing: cloud users and cloud providers. Cloud users are those enterprises which rely on the cloud computing for their business.

Cloud providers are companies that provide cloud resources. A remarkable category of cloud providers is Infrastructure as a Service providers, which are the companies that own physical data centers and provide computational resources as a service.

As in cloud computing there are two main actors involved, there are two sides of cost optimization: cost optimization performed by providers and cost optimization performed by users

Cost optimization performed by cloud providers mainly focuses on minimizing the cost to maintain a physical data centre. The cost minimization is typically achieved by reducing electricity consumption.

A proposed approach involves dynamically halting network devices. Another study proposes architectural principles, algorithms, and resource allocation policies for energy savings. Conversely, one of the most popular techniques for the cost optimization executed by cloud users is to choose the correct balance the types of instances i.e., cloud infrastructure planning.

This paper concentrates on cost optimization performed by users. In particular, this study focuses on finding the correct balance between on-demand instances and reserved instances. The choice is made for two reasons. First, while spot instances and Lambda are specific to Amazon Web Services, on demand and reserved instances might be relevant for different IaaS providers

Therefore, a larger part of cloud users may benefit from the results of this thesis. Second, researchers and practitioners studied the effectiveness of cot optimization using reserved instances: hence, contributions in this field might be more significant.

IV. PRIVACY PRESERVING SEARCHABLE ENCRYPTION SCHEME

A privacy preserving searchable encryption scheme (PSE) with fine-grained access control. The proposed PSE scheme provides a keyword based search facility over attribute based encrypted data with hidden access policy. The scheme is applicable in a scenario where there are multiple data owners and multiple data receivers. The scheme allows each user in the system with a set of attribute values, where a trusted authority verifies the user's attributes and assigns him a secret key. One of the key features of the PSE scheme is that once the secret key obtained, the user can generate the search query himself in the form of a trapdoor using the secret key assigned to him. In PSE scheme, the trapdoor generated by the user does not reveal the keyword used for the search nor the user's attributes. Each data owner encrypts the index with the help of trusted authority. The trusted authority makes the index secure by using the master secret key elements inside the index. The inclusion of the master secret key elements in the index prevents adversary, who is capable of adaptively generated search queries for chosen keywords, not to learn the keywords from the index. After encryption, the index is uploaded along with the encrypted document on the cloud. When a user sends the trapdoor, the cloud server performs the search operation with the input of trapdoor and encrypted index. The search process is repeated for each encrypted index related to each separate document. The search operation returns true if (1) the keyword inside the trapdoor is included in the index, and (2) the access policy of encrypted index is fulfilled with user's attributes.

The scheme is designed with multi-sender and multi-receiver setup, aimed at facilitating a data owner (sender) to encrypt the index of keywords related to his document and uploads it along with the access policy and the encrypted document on cloud storage, where the access policy is decided by the data owner and kept hidden inside the cipher text. The user (receiver) sends his search query in the form of trapdoor to the cloud storage server. The cloud server uses this trapdoor to search over all encrypted indexes uploaded on the cloud storage. The documents corresponding to the indexes for which the search operation returns true are sent back to the user as the result of his query.

A. MAJOR ROLES

- **Attribute Center:** The Attribute Centre (AC) is a trusted third party of the system. AC is responsible for generating system parameters and issuing keys to users of the system.
- **Token Generator:** The Token Generator (TG) is a trusted third party of the system, which assists a data owner for generating encrypted index. TG is involved in the process of generating encrypted index. For a small system/organization the AC itself can play the role of the TG. However, in the case of a system with sufficiently large number of users, autonomous entities (e.g. TGs) should play this role.
- **Data Owner:** Data owner encrypts and stores the data on cloud storage server. The encrypted data consists of two parts: (i) the index of encrypted keywords, and (ii) the encrypted document.
- **Cloud Service Provider:** Cloud Service Provider (CSP) provides storage and computation services for the entities of the system.
- **Receiver (Data) User:** Receiver user generates and submits a trapdoor to CSP. The CSP searches over the encrypted indexes using this trapdoor. The documents corresponding to the indexes for which the search operation returns true are returned to the user. Finally, the user decrypts the resultant documents.

The data owner runs a protocol with TG to get the encrypted word tokens for creating encrypted index. The involvement of TG is for the inclusion of the master secret key parameters inside the cipher text components of encrypted index. This is done to make the scheme adaptively secure against chosen-keyword attack. We have chosen to place the TG on data owner side instead of data receiver side, which facilitates a user to generate a trapdoor from his secret key autonomously without waiting for the token from any trusted authority, which ultimately reduces the per-query interaction with the trusted third party and helps to gain efficient response time for search procedure. We note that although TG is involved in the process of encrypted index generation, it is not able to learn the keywords inside the index and the access policy enforced by the data owner for encrypted index.

The scope of the PSE scheme is to generate the index of encrypted keywords for a document and to perform privacy preserving search over the encrypted index. It does not include the encryption and decryption of a document. We assume that an existing efficient AABE scheme [2–5]can be used for the encryption and decryption.

B. SCHEMES

- **Setup(Pl):** The Setup(Pl) is run by the AC. It takes as input parameter a security parameter l and outputs the master private key MK, TG's secret key TSK and public key PK.
- **KeyGen(MK, L):** The KeyGen(MK, L) is a randomized algorithm and it is run by the AC. The algorithm takes as input the master key MK along with a set of attributes L of a user. It outputs secret key SKs for that user. The key SK is used to generate a trapdoor for performing search operation.
- **Encrypt Index(PK, W, T, TSK):** This is a protocol run between data owner and TG, where W is the set of keywords associated with document M. The data owner starts the computation to generate the encryption for each keyword w included in keyword set W. The data owner gets an

encrypted token for each w from TG to perform the encryption of keyword. To generate encrypted word tokens, TG uses his secret key TSK. Owner outputs a set of encrypted words CTW, known as encrypted index for document M.

- **Trapdoor(PK, SK, w):** Receiver user invokes this randomized algorithm to make a trapdoor for retrieving the documents from CSP whose associated index contains an encrypted entry for the word w and for which he possesses sufficient access rights. The algorithm outputs a trapdoor tw generated for w.
- **Search (tw, CTW):** This is a deterministic algorithm run by the CSP. The algorithm takes as input the trapdoor tw sent by the user and encrypted index CTW. The algorithm returns true if the word in tw matches with any of the keyword included in CTW and user's key satisfies the access policy of CTW.

C. SYSTEM MODEL

The Attribute Center (AC) is a trusted third party of the system. AC is responsible for generating system parameters and issuing keys to users of the system. The Token Generator (TG) is a trusted third party of the system which assists a data owner for generating encrypted index. TG is involved in the process of generating encrypted index. For a small system / organization the AC itself can play the role of the TG. However in the case of a system with sufficiently large number of users, autonomous entities (e.g., TG) should play this role. Data owner encrypts and stores the data on the cloud storage server. The encrypted data consist of two parts: (i) the index of encrypted keywords and (ii) the encrypted document. Cloud Service Provider (CSP) provides storage and computation services for the entities of the system. Receiver user generates and submits a trapdoor to CSP. The CSP searches over the encrypted indexes for which the search operation returns true are returned to the user. Finally, the user decrypts the resultant documents.

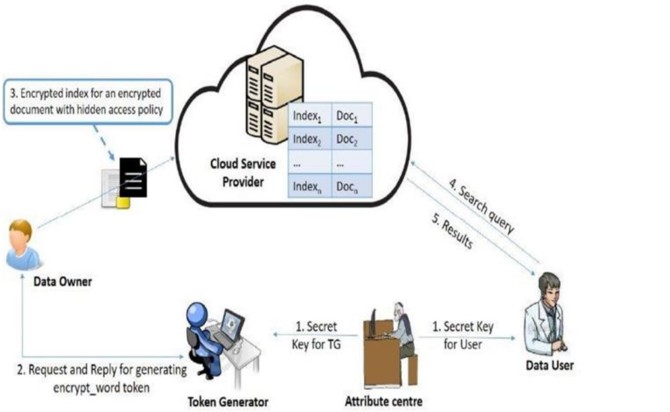
3. Encrypted index for an encrypted document with hidden access policy
Cloud Service Provider
Index_1 Doc_1
Index_2 Doc_2
... ...
Index_n Doc_n
4. Search query
5. Results
Data Owner
1. Secret Key for TG
1. Secret Key for User
Data User
2. Request and Reply for generating encrypt_word token
Token Generator
Attribute centre

FIG 1 CLOUD COMPUTING WITH PRIVACY PRESERVING SCHEME

Cloud service provides users with a certain degree of information leakage control in which no single cloud provider is privacy to all the user's data. It can be observed that there is a trust boundary between the metadata and storage servers. We assume that clients and metadata servers, which are situated inside the trust boundary, are trustable by users while remote servers outside the boundary are untrustworthy. For example, the metadata can be stored in private database servers while storage servers the total size of the data stored on a cloud to the size of entire data of the user, while the system specific weight is modelled as prior knowledge of a CSP, i.e., the set of data nodes which have been stored on it. Thus, the amount of prior knowledge of a CSP increases with the number of data nodes stored on it.

To optimize the information leakage, we presented the storesim, an information leakage aware storage in the multi cloud. StoreSim achieves this goal by using novel algorithms, BFSMinHash and SPClustering, which place the data with minimal information leakage (based on similarity) on the same cloud. Protocols only require each client only storing the newest version and use signature based approach to detect updates. Specifically, every local file in the client is partitioned into small chunks and these chunks are hashed with finger printing algorithms such as SHA-1 , MD5. In this way, a file's contents can be uniquely identified by this list of hashes and we call these hashes as signatures.

The metadata server will detect these modified chunks by comparing current signatures with the signatures of last version and only returns the signatures of these changed chunks to the client. Finally, the client will only upload these chunks with changes signatures to the storage server. A file 's contents can be uniquely identified by this list of hashes and we call these hashes as signatures. To synchronize the update to the cloud, the client will firstly send the signatures of current file to the metadata server. The client will only upload these chunks with changed signatures to the storage server.

V. ALGORITHM-PRIVACY PRESERVING SCHEME WITH SHA-12 FUNCTION

```
Step 1 :    Set up Pl,I
                Pl,I  is the data owners ID
Step 2 :    Key Generation (MK,L)
                MK<- Master Key
                L<- attributes
                SK<- Secret Key
                SK=(Di ,Mi, SKi)
                Where Di  <-Data
                Mi<- Message Space
                SKi<-Secret Key of Datas
Step 3 :    Perform Hash Values
        Generate Vectors of hash values Vi values from Mi using a secure hash function h
            3)a)H=(KGen,,F)
            3)b)δ<-KGen(λ)
        KGen  is an alg that generates and outputs a hash key δЄK on the input of a security parameter λ
            3)c) H<-F(δ,m)   F is an algorithm that outputs a hash value  HЄG on the input of a hash key δЄK and message mЄM
            3)d)The hash value of jth combination of Mi can be represented as
            δ<-KGen(λ)
            H<-F(δ,Ci,j)
            Hi,j =H(KGen,F)
            Ci,j  is the jth  combination of Mi
            Hi,j is the generated hash value of  Ci,j
            3)e)Hash values are generated for all combinations of secret attributes using the hash function Hi,j
            The vector  Vi is denoted as  Vi={ Hi,j| 1≤j≤ 2K-1}
Step 4 :    Encryption
            CTw= E(PK,W,T,TSK)
            PK<- Public Key
            W<-Set of Keywords
            T<- Access Policy
            TSK<- Token Generator's Secret Key
Step 5 :    Trapdoor
            tw<-Trapdoor(PK,SK,W)
Step 6 :    Search(tw, CTw)
            if  tw≥ CTw   (If the word in tw matches with any of the keyword in CTw  it returns true)
            return true
            else if  tw!=CTw
            return false  (No matching Keywords found)
```

The Privacy Preserving Scheme with SHA-12 with access policy enforced by the data owner and hidden inside the cipher text. The scheme is designed with multi-sender and multi-receiver setup, aimed at facilitating a data owner(sender) to encrypt the index of keywords related to his document and uploads it along with the access policy and the encrypted document on cloud storage, where the access policy is decided by the data owner and kept hidden inside the cipher text. The user(receiver) sends his search query in the form of trapdoor to the cloud storage server. The cloud server uses this trapdoor to search over all encrypted indexes uploaded on the cloud storage. The documents corresponding to the indexes for which the search operation returns true are sent back to the user as the result of his query.

VI. RESULTS AND DISCUSSIONS

A user can place a search query with multiple keywords, but it also creates a drawback because the user has to reveal the information about which keyword fields the queried keyword belongs to. Other limitation of Shi et al's scheme is that at most one value per attribute can be placed in the cipher text access policy, while in the PSE scheme multiple values for each attribute can be placed inside the cipher text access policy. In Shi et al's scheme, if a user wants to issue τ different search queries, then he has to communicate τ times with the trusted third party, which adds substantial communication overhead on user side. Whereas, in the PSE scheme, once the user obtains the secret key from the trusted third party, he can generate the search token independently without interacting with the trusted third party.

We have implemented the PSE scheme using pairing based cryptography (pbc) library .Bilinear pairings are constructed on the curve y2 = x3 + x over the field Fq for prime q=3 mod 4, where the order of the groups G0 and G1 is a prime of size 160 bits and the length of q is 512 bits. We have evaluated the scheme with varying number of attributes (n) = 3, 5 and 7 and their varying sizes of value-sets (m). The results shown in the graphs are taken from the experiments where we have taken the fixed size of value-set for each attribute as 5. Therefore, the total number of attribute values (m* n) are 15, 25, 35. The dataset contains the diabetes patient's medical records. Each record is a separate document containing the information of patients who are suffering from diabetes. Each report is identified with four keyword values which represents (1) the date when the test was conducted, (2) the time at which the blood sample is taken for test, (3) the type of report(glucose level or insulin dose), and (4) the outcome of test. Assuming that each report belongs to a different patient and each patient has his own privacy policy; the index is encrypted with each report with a different access policy. A Doctor working in hospital-A is able to search for all the reports conducted on, say 5th Feb., 2018, and for which the doctor has access rights (i.e. the reports whose access policy includes the attributes of Doctor and Hospital-A). The search procedure does not reveal the doctor's credentials and the keyword (in this case 5th Feb.2018). In real life, the health reports also include the patients 'name in medical report and a doctor might want to search for the reports of a certain patient. In this case, the patient name will be one of the keywords which is included in the encrypted index of that report. The Setup, KeyGen, Trapdoor algorithms, and Encrypt Index protocol are implemented on a Linux system with Intel core-i5 processor running at 2.30 GHz with 8 GB RAM. The Setup and KeyGen algorithms are executed by the AC.

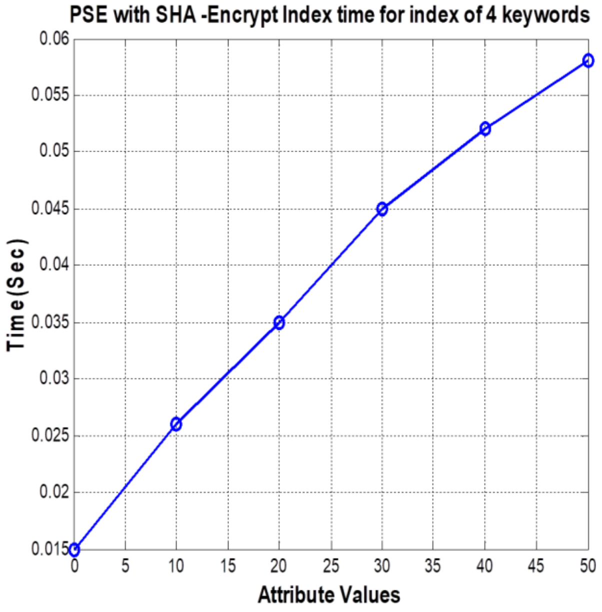

Fig 2 PSE with SHA-Encrypt Index time for index of 4 Keywords

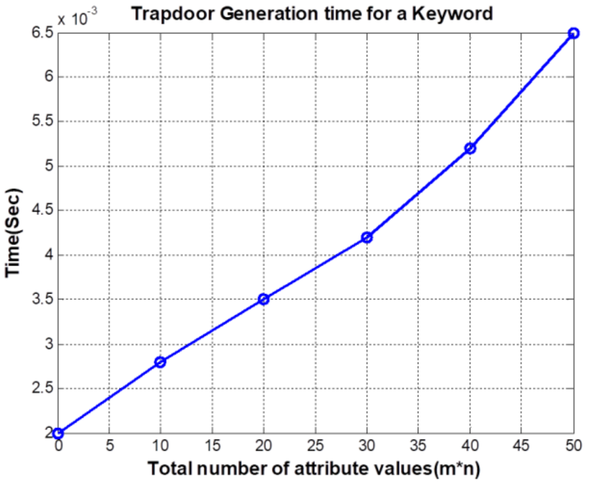

Fig 3 Trapdoor Generation time for a Keyword

The time complexity of Encrypt Index protocol and Trapdoor algorithms are shown in Figures (2) and (3). It is apparent from the results that the performance of the scheme operations linearly depends on the total number of attribute values. The Search algorithm is tested on a Google cloud computing instance of n1 series with 16 virtual CPUs. Each virtual CPU is implemented as a single hardware hyper-thread on a 2.6 GHz Intel Xeon E5. To provide the inputs to the search algorithm, the results obtained from Encrypt Index and Trapdoor algorithms are uploaded on the Google cloud instance.

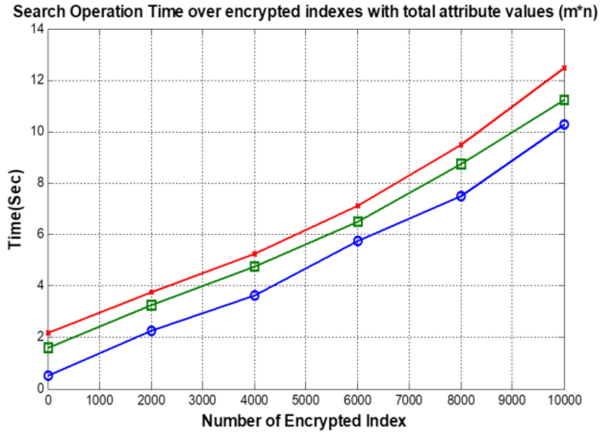

Fig 4 Search Operation Time over encrypted indexes with total attribute values (m*n)

Each index is for an individual report and it contains four keywords such as date of report, time of report, type of report and outcome of report. In Figure (4), we have shown the time to search over 500 to 10000 encrypted indexes with different number of total attribute values, where each index contains 4 keywords. The search operation time complexity is O(m *n), where m * n is the total number of attribute values in the system. Therefore, we have shown the results with different values of m * n.

VII.CONCLUSION

We proposed Privacy Preserving Scheme which uses SHA-12 technique which helps in assigning bit size of the meta data key being generated for the purpose of providing confidentiality . The proposed PSE scheme allows an authorized user to retrieve a subset of documents, over encrypted documents stored on CSP, pertaining to his chosen keyword and satisfying his access rights. The PSE scheme preserves the confidentiality of data and privacy of user's access rights. The search functionality of the PSE scheme is proven adaptively secure against chosen-keyword attack under SHA algorithm worked in a way to reduce the computational cost and reduce the complexity. In future we find a way easier to implement the algorithm with software available and improving the processing time much better and finally we will work to maintain the key generation size in fixed length.

VIII. REFERENCES

[1]. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attributebased encryption for fine-grained access control of encrypted data. In Proceedings of the ACM Conference on Computer and Communications Security, pp. 89–98, 2006.

[2]. T. Nishide, K. Yoneyama, and K. Ohta. Attribute-based encryption with partially hidden encryptor-specified access structures. In Proceedings of Applied Cryptography and Network Security, LNCS 5037, Springer, pp. 111–129,2008.

[3]. J. Li, K. Ren, B. Zhu, and Z. Wan. Privacy-aware attributebased encryption with user accountability. In Proceedings of Information Security, LNCS 5735, Springer, pp. 347–362, 2009.

[4]. Y. Zhang, X. Chen, J. Li, D. S. Wong, and H. Li.Anonymous attribute-based encryption supporting efficient decryption test. In Proceedings of the ACM SIGSAC Symposium on Information, Computer and Communications Security, pp. 511–516, 2013.

[5]. P. Chaudhari, M. L. Das, and A. Mathuria. On Anonymous Attribute Based Encryption. In Proceedings of the International Conference on Information Systems Security,LNCS 9478, Springer, pp. 378–392, 2015.

[6]. R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, Searchable symmetric encryption: improved definitions and efficient constructions. Journal of Computer Security. 19(5), pp. 895–934, 2011.

[7]. D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M. C. Rou, M. Steiner, Highly-scalable searchable symmetric encryption with support for boolean queries. In Advances in Cryptology-CRYPTO, Springer, pp. 353-373, 2013

[8]. S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, M. Steiner. Outsourced symmetric private information retrieval. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, ACM pp. 875-888, 2013.

[9]. C. Wang, W. Li, Y. Li, and X. Xu. A ciphertext-policy attribute-based encryption scheme supporting keyword search function. In Proceedings of Cyberspace Safety and Security, LNCS 8300, Springer, pp. 377–386, 2013.

[10]. Q. Zheng, X. Shouhuai, and G. Ateniese. VABKS: Verifiable attribute-based keyword search over outsourced encrypted data. In Proceedings of IEEE Conference on Computer Communications (INFOCOM), pp. 522–530, 2014.

[11]. P. Liu, W. Jianfneg, M. Hua, and N. Haixin. Efficient verifiable public key encryption with keyword search based on KP-ABE. In Proceedings of International Conference on Broadband and Wireless Computing, Communication and Applications, pp. 584–589, 2014.

[12]. K. Liang andW. Susilo. Searchable attribute-based mechanism with efficient data sharing for secure cloud storage. In IEEE Transactions on Information Forensics and Security 10(9):1981–1992,2015.

[13]. J. Li and L. Zhang. Attribute-based keyword search and data access control in cloud. In Proceedings of International Conference on Computational Intelligence and Security, pp. 382–386, 2014.

[14]. Q. Dong, Z. Guan, and Z. Chen. Attribute-based keyword search efficiency enhancement via an online/offline approach. In Proceedings of IEEE International Conference on Parallel and Distributed Systems, pp. 298–305, 2015.

[15]. B. Hu, Q. Liu, X. Liu, T. Peng, G. Wang, and J. Wu. DABKS: Dynamic attribute-based keyword search in cloud computing. In Proceedings of IEEE International Conference on Communications, pp. 1–6, 2017.

[16]. D. Koo, J. Hur, and H. Yoon. Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage. In Computers & Electrical Engineering, Elsevier, pp.34–46, 2013.

[17]. P. Chaudhari and M. L. Das. On the Security of a Searchable Anonymous Attribute Based Encryption. In International Conference on Mathematics and Computing, Communications in Computer and Information Sciencebook series (CCIS), vol. 655, Springer, pp. 16-25,2017

[18]. J. Shi, J. Lai, Y. Li, R. H. Deng, and J. Weng. Authorized keyword search on encrypted data. In Proceedings of Computer Security, LNCS 8712, Springer, pp. 419-435, 2014.

[19]. H. Wang, X. Dong, and Z. Cao. Multi-value-Independent Ciphertext-Policy Attribute Based Encryption with Fast Keyword Search. In IEEE Transactions on Services Computing,99, 2017.

[20]. P. Chaudhari and M. L. Das. A2BSE: Anonymous Attribute Based Searchable Encryption. In Proceedings of ISEA Asia Security and Privacy Conference, IEEE, 2017.

[21]. D. Cash, P. Grubbs, J. Perry, T. Ristenpart. Leakageabuse attacks against searchable encryption. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, pp. 668–679, 2015.

[22]. Y. Zhang, J. Katz, and C. Papamanthou. All Your Queries Are Belong to Us: The Power of File-Injection Attacks on Searchable Encryption. In Proceedings of USENIX Security Symposium, pp. 707–720, 2016.

[23]. M. Lichman. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml [Last Accessed 23 January 2017]

[24]. The Pairing-Based Cryptography Library.https://crypto.stanford.edu/pbc/ [Last Accessed 20 February 2017]