Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 10, October 2015, pg.12 – 19

RESEARCH ARTICLE

SECURED SUMMARIZATION OF PATIENTS REPORT BASED ON CLASSIFICATION ALGORITHM ON CLOUD

Ms. K.Devika Rani Dhivya
[*]M.Sc.,M.Phil.,MBA.,
Assistant professor,
BCA&M.Sc(SS) Department,
Sri Krishna Arts and Science college,
Coimbatore,
Devika58@gmail.com

S.Deebika, [**]
IV M.Sc.SS,
BCA&M.Sc(SS) Department,
Sri Krishna Arts and Science college,
Coimbatore,
deebika95@gmail.com

Abstract—*Nowadays secured data sharing is a challenging process in cloud computing. This paper is to implement the secured data sharing in the online health communities. These online health communities continue to offer huge variety of medical information useful for medical practioners, system administrator and patients alike. In this paper the doctors and patients gets interact with each other. The patients searches for doctors and upload their medical details. Doctor views the details of the patient and suggests the drugs for the patient. The patients can also express their views and side effects on drugs used by them for the particular diseases. The conversations among the doctor and the patient are secured by encrypting the information using RC4 algorithm. The patients are classified based on their age, disease, cause of disease (climate, genetic, etc) using classification algorithm aprioiri algorithm. The patients were classified for the purpose of future analysis of different patients.*

Keywords—*Medical Practitioners, RC4 Algorithm, Aprioiri Algorithm*

I. INTRODUCTION

Data sharing is an important functionality in cloud storage. And also securely, efficiently, and flexibly share data with others in cloud storage. However, while enjoying the convenience of sharing data via cloud storage, users are also increasingly concerned about inadvertent data leaks in the cloud. Such data leaks, caused by a malicious adversary or a misbehaving cloud operator, can usually lead to serious breaches of personal privacy or business secrets. To address users' concerns over potential data leaks in cloud storage, a common approach is for the data owner to encrypt all the data before uploading them to the cloud, such that later the encrypted data may be retrieved and decrypted by those who have the decryption keys. Such cloud storage is often called the cryptographic cloud storage.

However, the encryption of data makes it challenging for users to search and then selectively retrieve only the data containing given keywords. A common solution is to employ a searchable encryption (SE)[5] scheme in which the data owner is required to encrypt potential keywords and upload them to the cloud together with encrypted data, such that, for retrieving data matching a keyword, the user will send the corresponding keyword trapdoor to the cloud for performing search over the encrypted data.

Although combining a searchable encryption scheme with cryptographic cloud storage can achieve the basic security requirements of a cloud storage, implementing such a system for large scale applications involving millions of users and billions of files may still be hindered by practical issues involving the efficient management of encryption keys, which, to the best of our knowledge, are largely ignored in the literature. First of all, the need for selectively sharing encrypted data with different users usually demands different encryption keys to be used for different files.

This paper describes new public-key cryptosystems which produce constant-size cipher texts such that efficient delegation of decryption rights for any set of cipher texts are possible. The novelty is that one can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. In other words, the secret key holder can release a constant-size aggregate key for flexible choices of cipher text set in cloud storage, but the other encrypted files outside the set remain confidential. This compact aggregate key can be conveniently sent to others or be stored in a smart card with very limited secure storage.

Summarization [2] is defined as taking information from the source, extracting content from it, and presenting the most useful content to the user in a condensed form and in a manner suitable to the user's application needs. Summarization is very important in different NLP applications like Information Retrieval, Quality Analysis, Text Comprehension etc. Commonly there are two types of summaries. First one is Extract in which contents from text i.e. words and sentences are reused. Second one is Abstract which includes regeneration of extracted contents[3].

The Online health communities continue to offer huge variety of medical information useful for medical practitioners. The patients express their views, including their experiences and side-effects on drugs used by them. This is to perform Summarization[2] of user posts per drug, and come out with useful conclusions for medical fraternity as well as patient community at a glance.

Fig 1.1. Cloud data sharing

II. WORKFLOW

Fig: 2. Working Process of Summarization of Patients Report

III. PROPOSED METHODOLOGY

In this paper, a decryption key as more powerful in the sense that it allows decryption of multiple cipher texts. The introduction of a public-key encryption which we call key-aggregate cryptosystem (KAC)[5]. In KAC, users encrypt a message not only under a public-key, but also under an identifier of cipher text called class. That means the cipher texts are further categorized into different classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key have can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of cipher text classes. The public system parameter has size linear in the number of cipher text classes, but only a small part of it is needed each time and it can be fetched on demand from large (but non-confidential) cloud storage. Previous results may achieve a similar property featuring a constant-size decryption key, but the classes need to conform to some pre-defined hierarchical relationship. This paper is uses the RC4 algorithm for performing the encryption and decryption process. Hence it is more flexible in the sense that this

constraint is eliminated, that is, no special relation is required between the classes. This paper is based on the RC-4 algorithm to overcome the disadvantages of the existing system. RC4 is a symmetric key cipher and bite-oriented algorithm that encrypts PC and laptop files and disks as well as protects confidential data messages sent to and from secure websites. Output bytes require eight to 16 operations per byte. It is a stream cipher.

I. RC4 Algorithm

RC4 was designed by Ron Rivest of RSA Security in 1987. While it is officially termed "Rivest Cipher 4", the RC acronym is alternatively understood to stand for "Ron's Code".

The name *RC4* is trademarked, so RC4 is often referred to as *ARCFOUR* or *ARC4* to avoid trademark problems. RSA Security has never officially released the algorithm; RC4 has become part of some commonly used encryption protocols and standards, including WEP and WPA for wireless cards and TLS. The main factors in RC4's success over such a wide range of applications are its speed and simplicity: Efficient implementations in both software and hardware are very easy to develop.

RC4 generates a pseudorandom stream of bits (a keystream). As with any stream cipher, these can be used for encryption by combining it with the plaintext using bit-wise exclusive-or; decryption is performed the same way (since exclusive-or with given data is an involution). (This is similar to the Vernam cipher except that generated *pseudorandom bits*, rather than a prepared stream, are used.) To generate the keystream, the cipher makes use of a secret internal state which consists of two parts:

1. A permutation of all 256 possible bytes (denoted "S" below).
2. Two 8-bit index-pointers (denoted "i" and "j").

The permutation is initialized with a variable length key, typically between 40 and 256 bits, using the *key-scheduling* algorithm (KSA). Once this has been completed, the stream of bits is generated using the *pseudo-random generation algorithm* (PRGA).The schematic representation of rc4 algorithm is as follows:

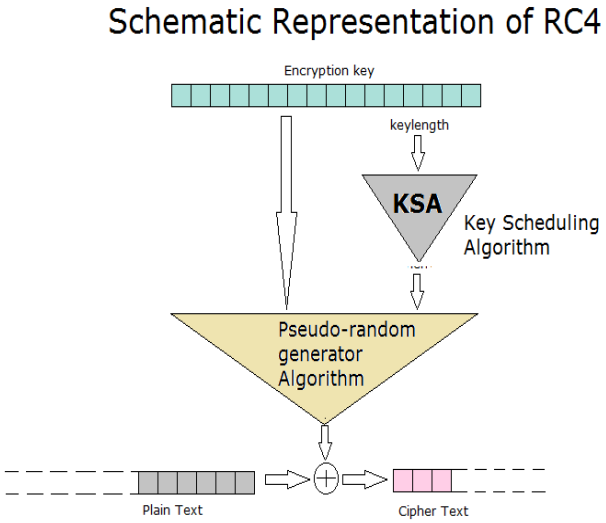

Fig: 3. Schematic Representation of RC4 Algorithm

Key-scheduling algorithm (KSA)

The key-scheduling algorithm is used to initialize the permutation in the array "S". "Key length" is defined as the number of bytes in the key and can be in the range $1 \leq$ keylength ≤ 256, typically between 5 and 16, corresponding to a key length of 40 – 128 bits. First, the array "S" is initialized to the identity permutation. S is then processed for 256 iterations in a similar way to the main PRGA, but also mixes in bytes of the key at the same time.

for i **from** 0 **to** 255

S[i] := i

endfor

j := 0
for i **from** 0 **to** 255
j := (j + S[i] + key[i modkeylength]) mod 256
swap values of S[i] and S[j]
endfor

Pseudo-random generation algorithm (PRGA)

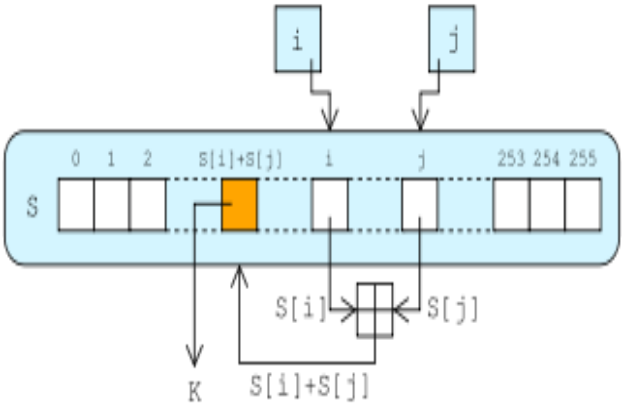

Fig: 4. Lookup Stage of PRGA

The lookup stage of RC4. The output byte is selected by looking up the values of S(i) and S(j), adding them together modulo 256, and then using the sum as an index into S; S(S(i) + S(j)) is used as a byte of the key stream, K. For as many iterations as are needed, the PRGA modifies the state and outputs a byte of the keystream.

i := 0
j := 0
whileGeneratingOutput:
i := (i + 1) mod 256
j := (j + S[i]) mod 256
swap values of S[i] and S[j]
K := S[(S[i] + S[j]) mod 256]
output K
endwhile

The classification is implemented using the apriori algorithm. Apriori is designed to operate on databases containing transactions. The algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets.

II. Apriori Algorithm

The Apriori algorithm was proposed by Agarwal and Srikant in 1994. Apriori is designed to operate on databases containing transactions. Each transaction is seen as a set of items (an *itemset*). Given a threshold C, the Apriori algorithm identifies the item sets which are subsets of at least C transactions in the database.

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as *candidate generation*), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length $k-1$. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The pseudo code for the algorithm is given below for a transaction database T, and a support threshold of ϵ. Usual set theoretic notation is employed, though note that T is a multiset. C_k is the candidate set for level k. At each step, the algorithm is assumed to generate the candidate sets from the large item sets of the preceding level, heeding the downward closure lemma. Accesses a field of the data structure that represents candidate set c, which is initially assumed to be zero. Many details are omitted below, usually the most important part of the implementation is the data structure used for storing the candidate sets, and counting their frequencies.

$$
\begin{array}{l}
\text{Apriori}(T, \epsilon) \\
\quad L_1 \leftarrow \{\text{large } 1-\text{itemsets}\} \\
\quad k \leftarrow 2 \\
\quad \textbf{while } L_{k-1} \neq \emptyset \\
\quad\quad C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a\} - \{c \mid \{s \mid s \subseteq c \wedge |s| = k-1\} \nsubseteq L_{k-1}\} \\
\quad\quad \textbf{for } \text{transactions } t \in T \\
\quad\quad\quad C_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\} \\
\quad\quad\quad \textbf{for } \text{candidates } c \in C_t \\
\quad\quad\quad\quad count[c] \leftarrow count[c] + 1 \\
\quad\quad L_k \leftarrow \{c \mid c \in C_k \wedge \ count[c] \geq \epsilon\} \\
\quad\quad k \leftarrow k + 1 \\
\quad \textbf{return } \bigcup_k L_k
\end{array}
$$

The steps involved in apriori algorithm is represented in diagrammatic form as follows:

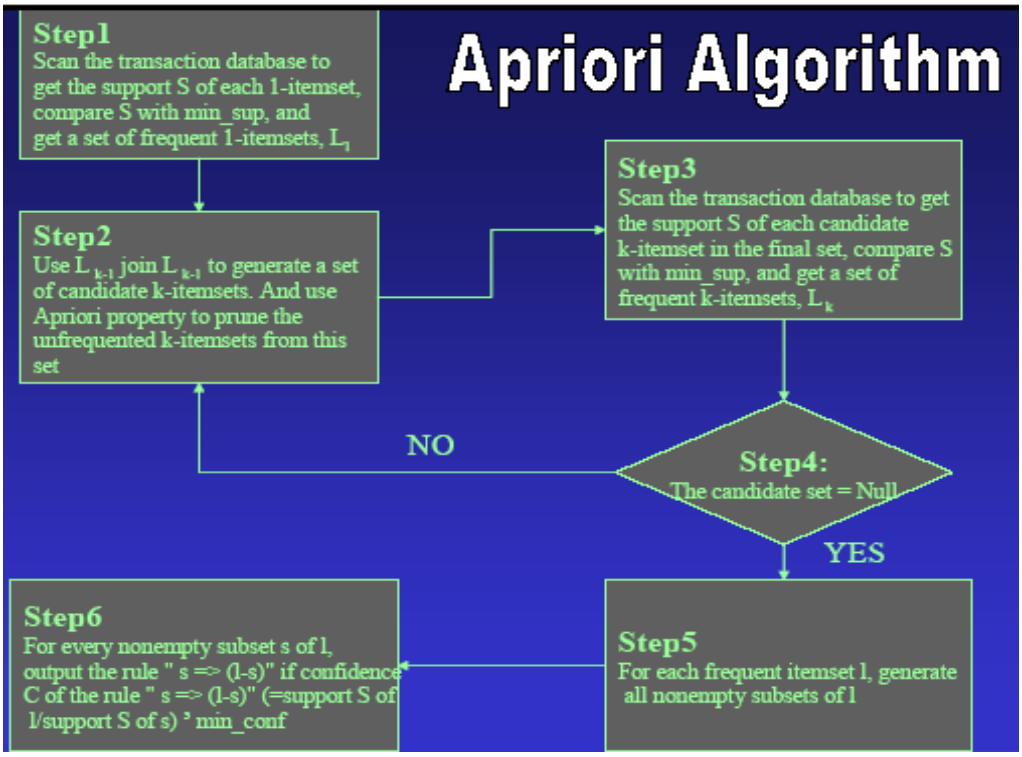

Fig: 5. Diagrammatic Representation of Apriori Algorithm

The Fig 2.3 shows the diagrammatic representation of Apriori algorithm. The apriori algorithm is used for the classification. The fig 2.3 shows the steps involved in classifying the datas based on the user request. This algorithm compares the searchable data with the datas in the database and then performs classification.In this paper the RC4 algorithm used for encryption and decryption purpose. The details which are uploaded by the user are encrypted and the public key has been sent to the doctor. The doctor views the details uploaded by the user by giving the public key and private key, and the apriori algorithm is used for classification[4] purpose. The new users can view the symptoms of the disease, drugs used for it and the views expressed by the previous users and also the patients are classified based on their age, sex, disease and the causes of the disease using the apriori algorithm.

IV. CONCLUSION

Considering the sensible drawback of privacy conserving information sharing system supported public cloud storage which needs a knowledge owner to distribute an outsized range of keys to users to change them to access the construct of key-aggregate searchable secret writing (KASE)[5] and construct a concrete KASE theme. Analyzing user posts from health communities for data discovery is a stimulating space in analysis. This work can facilitate patients to seek out association among totally different medicine, diseases and symptoms. It will facilitate doctors to seek out side-effects of various medicine in order that they will inflict higher medicine to alternative patients with similar illness. Pharmaceutical corporations are going to be conjointly benefited as we have a tendency to area unit classifying users of specific drug into totally different categories like traditional, depressed and happy. It will be indirect input to corporations to come to a decision that drug is in style, whether or not to supply alternate drug to the current etc. This projected work can equally profit all 3 parties–medical fraternity, patient community and pharmaceutical corporations.

REFERENCES

[1] JayashreeR,Srikanta Murthy K,Basavaraj .S.Anami, "Categorized Text Document Summarization in the Kannada Language by Sentence Ranking", 12th International Conference on Intelligent Systems Design and Applications (ISDA), pp 776-781, 2012.

[2] AlokRanjan Pal, DigantaSaha, "An Approach to Automatic Text Summarization using WordNet", IEEE International Advance Computing Conference (IACC), 2014.

[3] JesminNahar, Tasadduq Imam, Kevin S. Tickle, Yi-Ping Phoebe Chen, "Association rule mining to detect factors which contribute to heart disease in males and females", J. Nahar et al. / Expert Systems with Applications 40 (2013) 1086–1093, Elsevier, 2012.

[4] Lakshmi K.S, G. Santhosh Kumar, "Association Rule Extraction from Medical Transcripts of Diabetic Patients", IEEE, 2014.

[5] Baojiang Cui, Zheli Liu_ and Lingyu Wang "Key-Aggregate Searchable Encryption (KASE)for Group Data Sharing via Cloud Storage", IEEE,2014.

