Sensitive Information on Move

Sona Kaushik, Shalini Puri

Abstract— Sharing bulky data over the network securely is still a challenge for the organisations. Some data security architectures use multiple layers to provide them high security, so a Sensitive Information Security (SIS) Model is proposed in this paper which allows the sharing of very bulky data with high security over the network. The idea is to divide the documented text into its constituent parts, called data chunks, by using Natural Language Processing and then this processed and refined data is further separated into high risky and low risky data. Each data section is arranged in byte arrays of verbs, nouns, pronouns and so on. Bytes in the byte array are shuffled and encrypted realising the criticality of each array. Small chunks of shuffled arrays are encrypted and sent over the network securely. In this paper, this effort is made to present the investigations carried out using the rare combination of language processing and encryption techniques, to enhance the information security.

Index Terms - Byte Array, Category Sets, Data Security, Data Sharing, Encryption, Natural Language Processing, Term Frequency.

1 Introduction

In past years, it has always been a challenging area for the Information Technology Organisations, Defence Services and any other security domain services to share large secured data over the network. In this paper, a layered architecture model, called *Sensitive Information Security (SIS) Model* is proposed so that the large set of sensitive information is shared over the network with high security. The idea is to divide the data into small chunks and then to send these chunks of data with high security applied, over on the network.

The proposed security model processes the documented text by performing Natural Language Processing (NLP) to divide it into the chunks of data [1], the transformed form and sends them over the network which are securely collected and reassembled at the destination.

Section 2 describes the proposed layered architecture Sensitive Information Security (SIS) model. In section 3, the various layers of SIS model are discussed in detail. It discusses *the Scrambling Layer* by showing how a text document is processed to divide it into its lower level parts using the appropriate grammar and forming the syntactic tree structure [1] and then discusses the *Transformation Layer (TL)* where it is transformed into the encrypted chunks which are sent at the destination for recieval. Next, a case study which shows the NLP and Security, i.e., the use of SIS Model at Sentence Level, discussed in section 3. Section 4 discusses the conclusion and future scope of this model.

2 The Sensitive Information Security (SIS) Model

This section discusses our proposed model, i.e., *The Sensitive Information Security (SIS) Model* in detail. This model consists of two layers; *Scrambling Layer* and *Transformation Layer* to convert the text documents into data chunks and then to provide them high security to send over on the insecure network. Each layer plays its own important role in securing the data, so the proposed model works efficiently and effectively with high performance and accuracy.

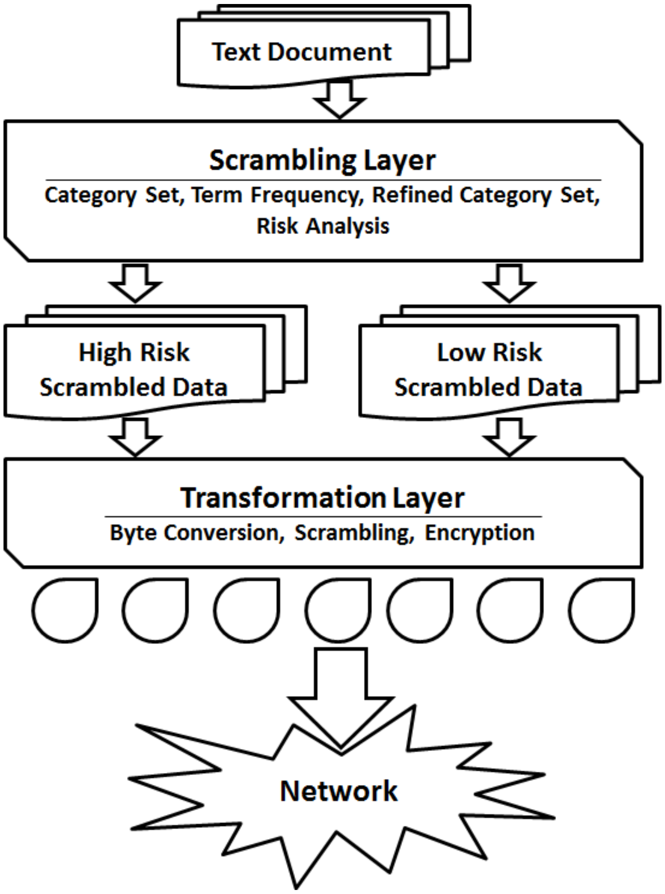
Text Document
Scrambling Layer
Category Set, Term Frequency, Refined Category Set, Risk Analysis
High Risk Scrambled Data
Low Risk Scrambled Data
Transformation Layer
Byte Conversion, Scrambling, Encryption
Network

Fig 1. The Sensitive Information Security (SIS) Model

The generalised view of the proposed SIS model is shown in figure 1 and shows the major use of Natural Language Processing, Byte Conversion, and Encryption and discussed in different layers [3] in detail in next sections.

- Sona Kaushik is currently pursuing M.Tech. degree program in computer science in Birla Institute of Technology, Noida, India. E-mail: sonakaushik22@gmail.com
- Shalini Puri is currently pursuing M.Tech. degree program in computer science in Birla Institute of Technology, Noida, India. E-mail: eng.shalinipuri30@gmail.com

2.1 Scrambling Layer

This section discusses that how the text documents are processed and converted into the form of data chunks and then into the High Risk Scrambled Data (HRSD) and Low Risk Scrambled Data (LRSD).

Consider a Text Document D containing the English language text only, which is a collection of various kinds of *Terms*, like nouns, pronouns, verbs, articles, adjectives, adverbs, prepositions and Determiners, like punctuation marks [1]. It is useful to point out here that there is a formal sense in which a language is defined simply as a set of strings without reference to any word being described or task to be performed. Though the great philosophical debates throughout the centuries have centred on the question of what a sentence means, but when we realize that understanding a piece of language involves mapping it into some representation appropriate to a particular situation, it becomes easy to answer this question.

This text document D is fed into the *Sentence Feeder (SF*). Each document has well-defined sentence boundaries, so each sentence is extracted by finding out their non-overlapping boundaries. So, SF extracts each sentence from D and feeds them into the *Category Set Extractor (CSE*) one by one. Each sentence in the CSE undergoes a series of steps and results into various category sets of nouns, pronouns, verbs, articles, adverbs, adjectives, prepositions, conjunctions and a set for special characters and punctuations from the document. To do so, the best method is the use of *Grammar*.

The most common way to represent grammars is as a set of production rules. Although the details of the forms they are allowed in the rules vary, the basic idea remains the same. *A sentence is composed of a noun phrase followed by a verb phrase*. The vertical bar should be read as "*or*". The ϵ denotes the rules of the grammar and compares them against the input sentence. Each rule that matches adds something to the complete structure that is being built for the sentence. So, a parse tree is made which simply records the rules and how they are matched.

The next step is to find out the syntactic structure of the sentence using the syntactic analysis. In this step, linear sequences of words are transformed into structures that show how the words relate to each other. The goal of this process is parsing of the sentence by converting the flat list of words that forms the sentence into a structure that defines the units that are represented by the flat list. So, the system is able to categorize all the independent terms and determiners and keeps them into their respective and most suitable category. Therefore, the text document is divided into its basic and low level constituent parts of various terms and determiners, each classified in its own category set where the various categories in CSE are named as CSE{C_1, C_2, C_3,…,C_m}, m is the total number of categories.

As these terms and determiners are made available in various categories of CSE, the next step is to evaluate the *Term Frequency (TF)* of each word (term or determiner) occurred in the different categories [2]. The TF is calculated for each different term *t* occurred in C_j where j is in between 1 and m. This TF is the number of occurrences of a term in its respective category C_j. Now, a counter loop is invoked for each category to count the total number of occurrences of each different word (for eg., Bob, or *is*) occurred in a category, and simultaneously each occurrence of a particular word is stored in the subcategory C_{nm} of the *Refined Category Set (RCS)* where n denotes the word number given to a different word occurred in a category. For an example, if C_1 denotes the noun category, in which the first available term is *Ram* which occurs three times in C_1, so its TF is 3 and the subcategory C_{11} stores three occurrences of Ram in RCS. This process continues and each counter value is compared with the *Threshold Value (TV)*, say 100. The TV is the maximum size of the subcategory set in RCS. It is a predefined value for the model and can be any numeric value in accordance to the requirement of the system. This comparison is made so that any word whose occurrence is very high makes its own subcategory set size large. Such sets can be decomposed into two or more sets. So, now each one is acting as an independent set. In this way, we have made the different threshold subcategories of each category in RCS using the term frequency of each different word occurred and achieve high performance ratio of the system.

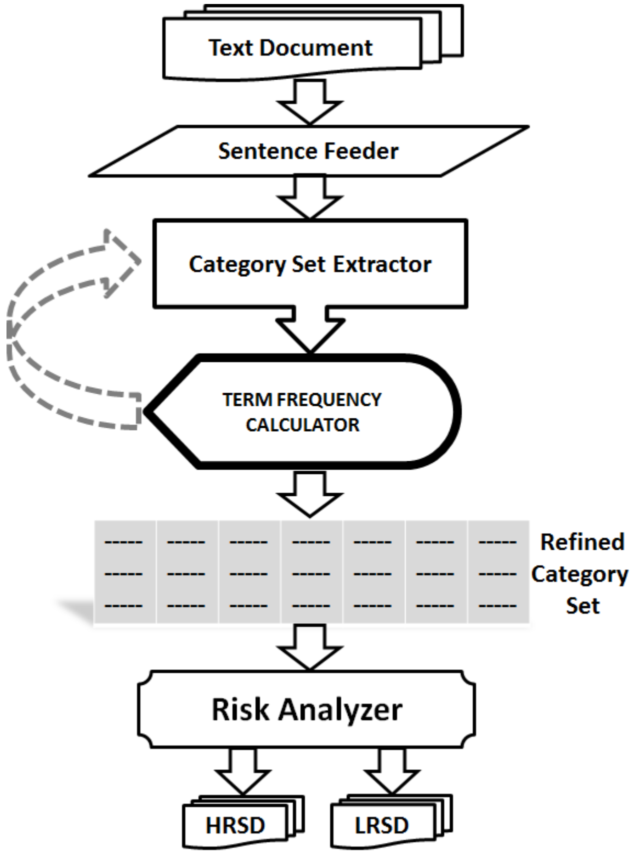

Fig 2. Detailed Scrambling Layer

As RCS is the set of final category sets, the next step is to keep each set under Risk Analysis (RA). RA is discussed in the next section where the resultant categories of RCS will be sent to Risk Analyser. This RA will divide the RCS into two categories, called *High Risk Scrambled Data (HRSD)* and *Low Risk*

Scrambled Data (LRSD) to separate the important terms of D from the less important words of D.

2.2 Transformation Layer

Transformation Layer encodes the content of each RCS into a non-readable format. This is basically encryption, but before encrypting the data, two things are considered. Firstly, the contents are not directly encrypted, but are changed to the byte array [4]. Secondly, the encryption scheme applied is symmetric or non-symmetric; is based on the criticality of the content. For high risk data, HRSD, which contains the most important information, like nouns, is encrypted using non-symmetric encryption. The rest of the data, LRSD, is using symmetric single key schemes for encryption.

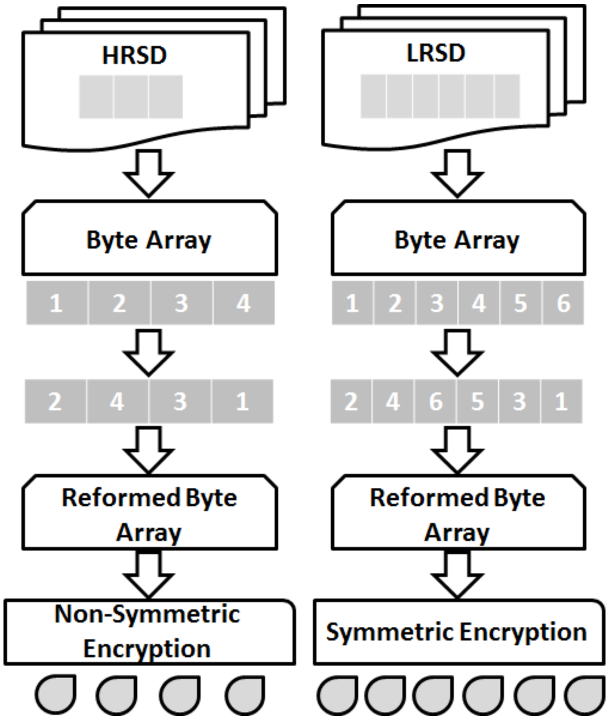

Fig 3. Detailed Transformation Layer

Before encrypting the byte arrays, they are tousled from its original structure. The *Byte Array (BA)* is the sequence concatenation of category sets. This sequence is messed by dividing the byte array length into equal even divisions. If the length of array is odd, then minus one of its length for division and later add that last bit to the last division of array. Arrangement of all such sections is done in a predefined order. One possible arrangement can be arranging all the even sections in the beginning in ascending order, and then arrange the odds in descending order. Some other more complex scheme for disarrangement of the byte array can also be applied. The resultant is the *Reformed Byte Arrays (RBA)*; set of high risk data and other set for low risk data. Now, if the array belongs to the category of high risk data, then a non-symmetric encryption scheme is applied. In the model, RSA scheme is used which uses a public and private key mechanism for encryption. The key exchange is not in the scope of this paper.

For low risk byte arrays, AES encryption scheme is used which involves single key to encode and decode of data. The identification of high and low risk chunk is made by flagging each chunk.

The entire transformation results in small RCS encrypted chunks. These chunks are now ready for flow on the network to be received at the other end.

2.3 Chunks at Destination

At the receiver's end, chunks are identified as high risk or low risk, and so accordingly are decrypted. Then all the bytes are re-assembled to their original structure. Now the structures of received chunks have taken the shape of Refined Category Sets. These RCS are then fed to the mapper that uses the sentence and position number from each word of each RCS. Finally, the original structured document D is prepared and successfully received at destination.

3 Case Study: Sentence Level Analysis

As per the proposed SIS model for the sensitive information based systems, let us consider an example where we input a sentence for processing according to the model and finally results the input to the secured data chunks. Consider the following sentence-

"Prime Minister Manmohan Singh and his entourage left India for New York 21 September 2011."

Step 1: *Write the Grammar for the System.*

S→NP VP
NP→ART NP1
NP→NP1 NP2
NP→PRO|PN|NP1
NP→NP1 NUMP
VP→V
VP→V NP
NP1→ADJS N
NP1→PREP N1
NP1→N1 PN1
NP1→PN1
NP1→N1
NP2→CON NP1
NP2→ϵ
ADJS→ϵ|ADJ ADJS
N1→N|ϵ
PN1→PN
NUM1→ϵ|N NUM
NUM→ϵ|21|2011|2012
NUMP→NUM|NUM1
PN→PN|Manmohan|Singh
PREP→his|her|him| |for|ϵ
N→Prime|Minister|Entourage|September|India|New|York|ϵ

V→left|gone|went
ART→the|a|an
CON→and|but

Step 2: *Draw the Syntactic Tree Structure of the given sentence.*

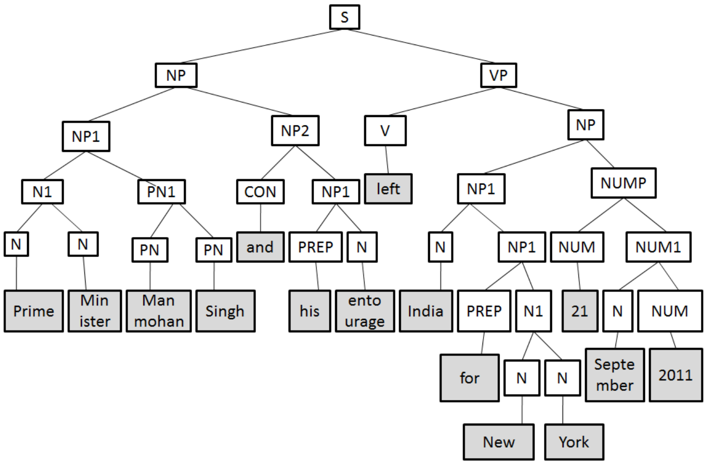

Fig 4: Syntactic Structure of the Sentence *'Prime ...2001'*.

Step 3: *Find out the Category Sets.*

There will be total seven category sets, named as Noun, Proper Noun, Number, Verb, Conjunction and Preposition, named as C_1 to C_6.

C1: Prime Minister entourage September New York India

C2: Manmohan Singh

C3: 21 2011

C4: left

C5: and

C6: his for

Step 4: *Calculate Term Frequency of each different word occurred in each category, compare them with the threshold value 10, and Form Refined Category Sets.*

As each word is occurring only one time, C11 denotes *Prime* with TF = 1 < 10. Similarly, C21 is for *Minister with TF = 1*. In the same manner, C31, C41 and so on, will be made and checked with the TV.

Step 5: *Form the Groups of HRSD and LRSD.*

In this example, HRSDs are the nouns and numerals, and all the rest categories as LRSDs. Here we divide the categories as:

HRSD: C1, C2, C3.
LRSD: C4, C5, C6.

Step 6: *Both data sets are converted to byte arrays.*

For each category, the byte conversion process is executed which converts each in separate byte array.

Step 7: *Byte arrays are tousled and given a new form which if attempted to convert to data will not give the original data sets directly.*

Each byte array for categories is shuffled as discussed in section 2.2.

Step 8: *Encrypt the arrays using non-symmetric encryption algorithm (RSA) for HRSD byte tousled arrays and symmetric encryption algorithm (AES) for LRSD byte tousled arrays.*

Step 9: *Send these chunks of arrays to the destination.*

These array chunks are sent to the destination the time differences in each send operation.

4 Conclusion

The Layered Architecture SIS Model is an effective and efficient way to send the bulky text documents with the high security provision over on the network. This proposed model shows high performance and accuracy than the other models of security. It uses the strong concepts of Natural Language Processing to find out the various different words with their associated TF and to separate the more important data from the less informative ones by adding an extra layer of security. The conversion of data chunks into byte arrays reduces the size and makes it easier to further process with another security pattern of scrambling the bytes.

Although using the standard encryption schemes are no way new to use, but using more critical algorithm with more critical data and vice versa gives new approach of suppressing the computational complexity of the model. This paper can further be extended to include the non-textual data, like images, sound, music, and video.

References

[1] Richand Knight, "Artificial Intelligence", Mc Graw Hills, Third Edition, 2010.

[2] Shady Shahata, Fakhri Karray, and Mahamed Kamel, "Enhancing Text Clustering using Concept-based Mining Model", *IEEE Proceadings of the sixth international conference on Data Mining*, 2006.

[3] Jithesh Sathyan and Manesh Sadasivan, "Multi-Layered Collaborative Approach to Address Enterprise Mobile Security Challenges", *IEEE Second Workshop on Collaborative Security Technologies (CoSec)*, 2010.

[4] Adeel Bhutta and Hassan Foroosh, "On combining encryption for multiple data streams", *IEEE INMIC Ninth International Multitopic Conference*, 2005.