Design and Development of a Novel Algorithm for Search on Encrypted File

R. Priya
Assistant Technical Officer
Department of Computer Science
Bharathiar University,Coimbatore–46

T.Devi, Ph.D
Associate Professor & Head i/c,
Department of Computer Applications Bharathiar University, Coimbatore–46

ABSTRACT

Everyone has secrets and some have more than others. Each and every one wanted to have their data and information to be more secure and confidential. There are many ways to maintain the data and information to be safe. The main objective of this paper is to be maintained the file or data to be secure. So, proposed system has developed for searching keyword in encrypted format and search with encrypted file.

General Terms

Data security, Encryption, Decryption and keyword searching algorithm

Keyword

Data security, File security, System security and Encryption

1. INTRODUCTION

This paper is concerned with the study and analysis of computer security, encryption methods in order to provide an effective security system through search on encrypted file. There are many ways to maintain the data and information to be safe. The main objective of this paper is File Security and Data Security.

2. SECURITY AND ENCRYPTION

Security is mainly used to secure the computer at 3 levels: hardware, information and service. Different security methods that is available for maintaining security in computer [1]. Computer security have many types based on working methods, tools, hardware, software etc., Main types of computer security is shown in fig 1. That is, physical security of computers and networks equipment, local area network security, network server security, switch/router/hub security, web server security and system security. In system security has two types. One is data security and other is disk security. This paper mainly focuses the data security.

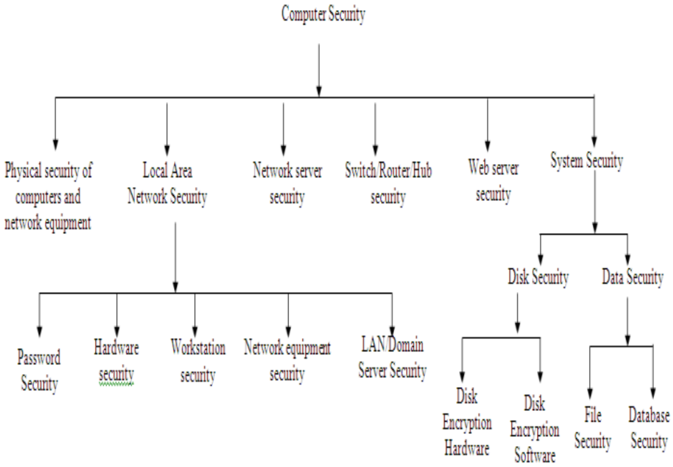


Figure 1. Types of Computer Security

Encryption is mainly used for Data protection. Data security refers to mathematical calculations and algorithmic schemes that transform plaintext into cipher text, a form that is non-readable to unauthorized parties [2].

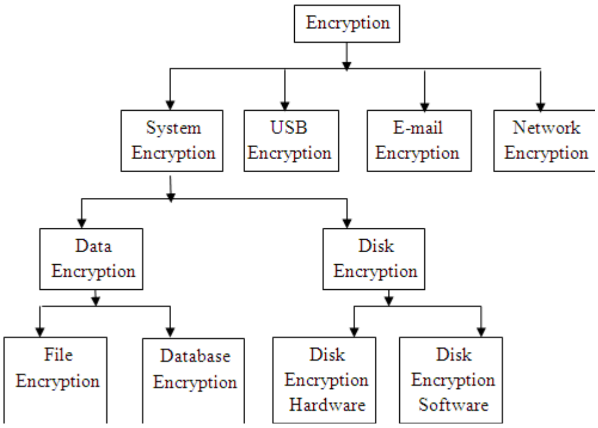


Figure 2. Types of Encryption Method

Encryption scheme have two settings. Symmetric (private key) is encryption and decryption is performed under a key shared by the sender and receiver. In Asymmetric (public key), sender has some public information and the receiver holds some corresponding secret information [3]. Encryption can be divided into many types. Mainly it can be divided into 5 types as shown in fig 2. Data encryption can be divided in two types and they are (1) File encryption and (2) Database encryption. File encryption used to protect a file with encryption mechanism. Similarly Database encryption used for database security with encryption mechanism.

3. SEARCH AND SEARCH ON ENCRYPTED FILE

Data Extraction is the operation of extracting data from a source file [4]. Designing and creating the extraction process is often one of the most time-consuming tasks process. The source files might be very complex and poorly documented and thus determining extracted data can be difficult.

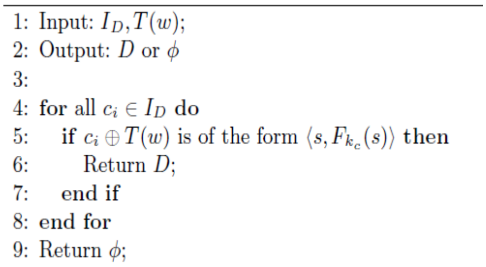
Data extraction is the act or process of retrieving data out of (usually unstructured or poorly structured) data sources for further data processing or data storage (data migration). Data extraction method is called Information retrieval [5]. The process of searching specific information among a large number of information items is known as information retrieval (IR).

CRYPTOGRAPHIC MECHANISM

Cryptography can be achieved through various mechanisms. The goal is more specific in that the invention provides a privacy cryptographic mechanism and implementation that guarantees that an interloper must perform an average amount of work to recover the secret that is much greater than the amount of work required to create the secret between the two parties [6]. Some of the cryptographic mechanisms are given below:

Secret Key Algorithms

Secret key algorithms otherwise known as ‘Symmetric’ key encryption, which uses the same key for both, encrypt and decrypt a message. Some of the secret key algorithms are given below. Fig 3 is shows that algorithm for search keywords.



```
Input: $I_D, T(w)$;
Output: $D$ or $\phi$

for all $c_i \in I_D$ do
    if $c_i \oplus T(w)$ is of the form $\langle s, F_{k_c}(s) \rangle$ then
        Return $D$;
    end if
end for
Return $\phi$;
```

Figure 3. Algorithms for Search Keywords

4. PROPOSED ALGORITHM FOR SEARCHING ENCRYPTED KEYWORD FROM ENCRYPTED FILE

Any method can use for creating encrypted file by authorized persons. In this proposed system is created encrypted file with +4 methods. i.e., alphabets and numbers are the contents of the file, each and every alphabet and numbers have to change with next to four letters. Fig 4 shows the proposed algorithm for searching encrypted keyword from encrypted file.

For example,
a →e, b→f, c→g ,…….
1→5, 2→6, 3→7……

```
Existence : searching keyword in file
    1. Read : Encrypted file
    2. Decrypt the needed file
    3. Read : Keyword
    4. Search, keyword from file
    5.
Proposed Algorithm
Searching the existence of given keyword from file
    1. Searching keyword
    2. File
    3. [Encrypt the file]
       Repeat
       (1) Find : ch; the character from file
       (2) if ch equal to whitespace, then
           use same white space
           else if ch is alphabet, then,
           encrypt = ch+4 alphabets
           [end of if]
           until [end of character in file]
    4. [Encrypt the keyword]
       for all character in keyword, do
       if character = alphabet , then,
       set Encrypt Keyword := character +4 alphabets
       else if character = Number, then
       set encrypt word= number +4
       [end of if]
       [end o for all ]
    5. [searching a keyword]
       for all word in encrypted file, do
       if (word equal to encrypt keyword, then,
       count := count+1
       write: “Occurrence of keyword”, count
       else
       write: “keyword not exists”
       [end of if]
       [end o for all ]
```

Figure 4. Proposed algorithms for encrypted keyword searching from encrypted file

In existence method can fetch records or information from the file with keywords ([7] and [8]) . If give keyword for searching from information and get back information’s from file [9]. But in this method unauthorized persons can fetch the information from file and also can corrupt the original file [10]. So it is not a security process. But there is no method for keeping security for data retrieval [11]. Overcome this problem, proposed system had been developed with encrypted file and encrypted keyword. Many methods are there for encryption process. In this method, original file stored in encrypted format. Searching keyword is also encrypted. Which encryption method is used for the encryption file, the same method is used for encrypt the keyword also. So Authorized persons only known which method is used for encryption. Unauthorized persons cannot find the encryption methods for original file and keyword. So it is a purely security process.

5. DESIGN OF ALGORITHM

The overall system design is the search on encrypted files using encryption on given keyword has been designed using top down approach and developed using bottom up approach.

Searching Encrypted File with Encrypted Keyword (SEFE (K)) method architecture of the prototype is as shown in fig 5.

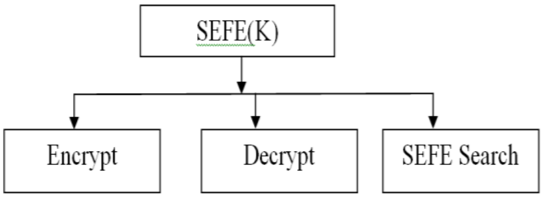


Figure 5. Architecture of SEFE Method

The overall software prototype is subdivided in to three types of modules namely,

1. **Encrypt Module :** Given an input text file, this module applies encryption on the contents of the file and output an encrypted file.
2. **Decrypt Module :** Given an input encrypted file, this module applies decryption on the contents of the file and output a text file.
3. **Search Encrypted Files:** Given a keyword and the encrypted data file as input this program initially encrypts the keyword using the same method by which the encryption has been carried out. Then, the encrypted keyword text is searched in the encrypted file and the records where the match occurs are retrieved and shown to the user. If there is no match, "None records found" is reported to the user.

6. DEVELOPMENT OF SOFTWARE PROTOTYPE

Has mentioned earlier this module carries out the searching function of the prototype. The step by step process is this module is explained in fig 6. In the searching function, initially user taken for searching keyword with encrypted file. The keyword should be encrypted in the same encryption method of encrypted file. After encryption process of the keyword, the encrypted keyword searches the matching records or contents in the encrypted file. If matching contents are there, the output will be displayed. Otherwise, "There is no matching" is reported to the user. In this module can calculate the time also. Total time of process also displayed. Searching time is depends of size of the files. If size of file is increasing, the searching time also increase.

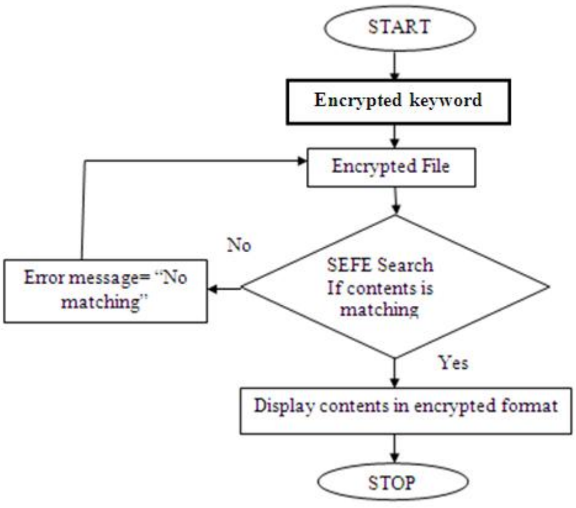

START
Encrypted keyword
Encrypted File
No
Error message= "No matching"
SEFE Search If contents is matching
Yes
Display contents in encrypted format
STOP

Figure 6. Step by Step Process of SEFE Search

6.1 Encryption Module

Given an input text file, this module applies encryption on the contents of the file and output an encrypted file. i.e, if text file is given, the file convert to encrypted file with this encryption module and then output file come out from the process in encrypted format. Sample input data is shown in fig 7.

101	priya	75
102	vidya	80
103	kalai	60
104	S.Priya	50
105	Shanmugapriya	79

Figure 7. Sample input data

The crux of the module is the mathematical function which is used for encryption. For example, "+4" is used to encrypt the file. The sample output is shown in fig 8.

```
545$$$tvm}e$$$$$$$$$$;9546$$$zmh}e$$$$$$$$$$<454
7$$$oepem$$$$$$$$$$:4548$$$W2Tvm}e$$$$$$$$94549
$$$Wlerqyketvm}e$$;=
```

Figure 8. Sample output data

The algorithm is used for this module is given in fig 4. In this algorithm, the given text file is taken for encryption, initially the character of the contents files taken one by one and convert into encrypt format as above. Ie., if c is the character, fI = c + 4 is the encryption function for the input given in fig 7.

In this SEFE method, the thesis used the methods is only add with 4 characters of the text character. But based on this method use can modify the addition numbers as their wishes. The sample source code of the encryption module is given in fig 9. In this source code directly get any text file name with scanf method and put the character one by one and add with 4 for getting encrypted file. If there is no any file like this name, "**The file %s can't be open**" is reported to the user. In this command %s denotes text file.

```
Printf("Enter the filename to Encrypt:");
scanf("%s",name);
fp=fopen(name,"r+");
if(fp==NULL)
        {
        printf("The file %s can't be open",name);
        getch();
        exit();
         }
fp1=fopen(temp,"w+");
if(fp1==NULL)
       {
        printf("The file Temp can't be open");
        getch();
        exit();
        }
c=fgetc(fp);
while(c!=EOF)
       {
        j=c;
        j=j+4;
        cc=j;
        fputc(j,fp1);
        c=fgetc(fp);
        }
fclose(fp);
fclose(fp1);
printf("The file is Encrypted:");
```

Figure 9. Sample source code for encryption module

6.2 Decryption Module

Decryption module is just reverse of the encryption module. In the decryption function, each and every character is taken for decryption and then minus with 4 to each character. For example e turns to a, f written to b, g written to c and so on.

For example tvm}e → priya.

Given an input Decrypted file, this module applies for decryption on the contents of the file and output a text file. i.e, if decrypted file is given, the file convert to decrypted file with this decryption module and then output file come out from the process in original text format. Sample input data is shown in fig 10.

```
545$$$tvm}e$$$$$$$$$$;9546$$$zmh}e$$$$$$$$$$<4
547$$$oepem$$$$$$$$$$:4548$$$W2Tvm}e$$$$$$$$9
4549$$$Wlerqyketvm}e$$;=
```

Figure 10. Sample input data for decryption module

The crux of the module is the mathematical function which is used for decryption. For example, "-4" is used to decrypt the file. The sample output is shown in fig 11.

```
101  priya            75
102  vidya             80
103  kalai             60
104  S.Priya           50
105  Shanmugapriya  79
```

Figure 11. Sample output data for decryption module

Based on Fig 4, user can decrypt any encrypt file into original text format. Fig 12 shows the source code for getting directly any encrypted file name and put the character one by one and minus with 4 for getting original text file.

```
    Printf("Enter the Filename to Decrypt:");
    scanf("%s",name);
    fp=fopen(name,"r+");
    fp1=fopen(temp,"w+");
    c=fgetc(fp);
    while(c!=EOF)
    {                   j=c;
                        j=j-4;
                        cc=j;
                        fputc(cc,fp1);
                        c=fgetc(fp);          }
                        fclose(fp);
                        fclose(fp1);
                        remove(name);
                        rename(temp,name);
                        printf("The file is decrypted:");
                        getch();
  }  }
```

Figure 12. Sample source code for decryption module

6.3 Searching Encrypted file with Encrypted Keyword (SEFE) Search

In the searching function, initially user taken for searching keyword with encrypted file. The keyword should be encrypted in the same encryption method of encrypted file. After encryption process of the keyword, the encrypted keyword searches the matching records or contents in the encrypted file. If matching contents are there, the output will be displayed in encrypted format. Basic source code of the SEFE module is displayed in Fig 13.

```
    Clock_t start, end;
    clrscr();
    start = clock();
    printf("The time start: %f\n", start);

    printf("\n\tEnter the file name: ");
    scanf("%s",&f);
    f1=fopen(f,"r");
    f2=fopen("time.txt","at");
  c=fgetc(f1);
 while(c!=EOF)
      {
            printf("%c",c);
             c=fgetc(f1);
     }
     n=I;
    printf("\n\n\tEnter the keyword : ");
    scanf("%s",&b);
    len=strlen(b);
 rewind(f1);
 i=0;
 strcpy(strtemp,"");
 tempi=0;
 while(!feof(f1))
 {
 c=fgetc(f1);
 if(c-4=='\n')
 {
 strtemp[tempi++]='\0';
 reccnt=reccnt+1;
 if((stricmp((strstr((strlwr(strtemp)),(strlwr(b)))),""))!=0)
 {
 printf("%s\n",strtemp);
 recmatch++;
 }
```

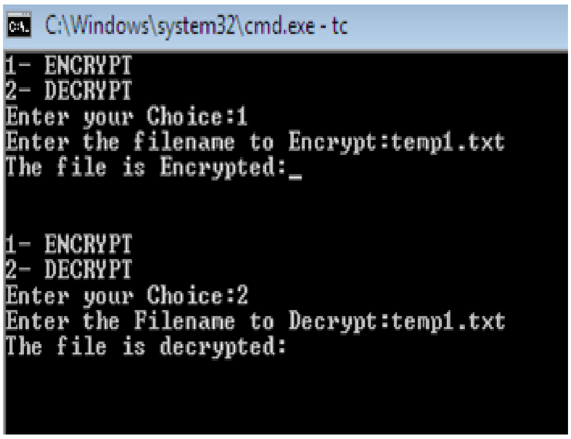
Figure 13. Sample source code of SEFE Search

7. DISCUSSIONS

This thesis proposed to unique method. It is a new method for fetching data from text file. Administrator only knows which methods are used for encrypted file and encrypted keyword. Both are maintained with same encryption methods. So unauthorized persons cannot find out the correct encryption method and they cannot get any information from this file. It is a very security method for fetch information from file.

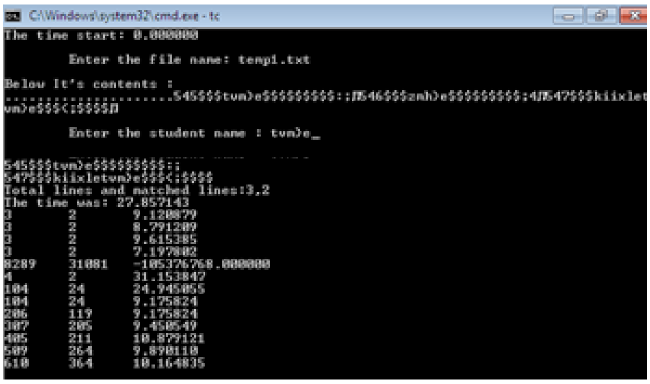
8. TESTING AND RESULTS

The encryption module, decryption module and search modules showed in fig 14 and fig 15. In integration testing, fig 8 is based on encryption modules for integration testing. In fig 9 explains that integration testing for SEFE search.



```
C:\Windows\system32\cmd.exe - tc
1- ENCRYPT
2- DECRYPT
Enter your Choice:1
Enter the filename to Encrypt:temp1.txt
The file is Encrypted:_

1- ENCRYPT
2- DECRYPT
Enter your Choice:2
Enter the Filename to Decrypt:temp1.txt
The file is decrypted:
```

Figure 14. Integration Testing for Encryption Module and Decryption Module

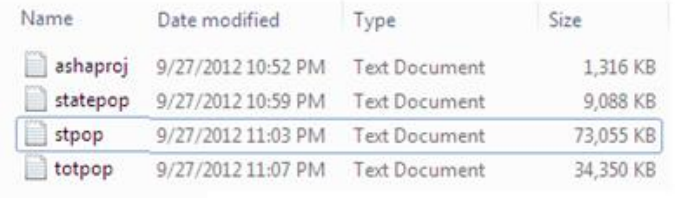


```
C:\Windows\system32\cmd.exe - tc
The time start: 0.000000

         Enter the file name: temp1.txt

Below It's contents :
........................545$$$tvm)e$$$$$$$$$::ЛБ46$$$zmh)e$$$$$$$$$;4ЛБ47$$$kiixlet
vm)e$$$<;$$$$Л

         Enter the student name : tvm)e_

545$$$tvm)e$$$$$$$$$::
547$$$kiixletvm)e$$$<;$$$$
Total lines and matched lines:3,2
The time was: 27.857143
3        2        9.120879
3        2        8.791209
3        2        9.615385
3        2        7.197802
8289     31081    -105376768.000000
4        2        31.153847
104      24       24.945055
104      24       9.175824
206      119      9.175824
307      205      9.450549
405      211      10.879121
509      264      9.890110
610      364      10.164835
```

Figure 15. Integration Testing for SEFE Search

9. TESTING WITH REAL DATA

Fig 16 shows the list of files for processing time of encryption module, decryption module and SEFE search module. The system is tested with big real data. In this figure, the Asha project in Tamilnadu details in ashaproj.txt file. State population details are in statepop.txt, state wise population and male, female population details are in stpop.txt. Similar to total population of the world is explained in totpop.txt. These are the original files. These text files are downloaded from internet. The system is used for any text file like these files.



Name	Date modified	Type	Size
ashaproj	9/27/2012 10:52 PM	Text Document	1,316 KB
statepop	9/27/2012 10:59 PM	Text Document	9,088 KB
stpop	9/27/2012 11:03 PM	Text Document	73,055 KB
totpop	9/27/2012 11:07 PM	Text Document	34,350 KB

Figure 16. Sample Text File for Encryption Module, Decryption Module and SEFE Search Module

The result of obtained through this is as given in table 1. In fig 17 shows the relationship between the four different type of text file. If the size of file changed, then automatically time of encryption module, decryption module and SEFE search also changed.

Table 1. Details of Time Processing in Encryption Module, Decryption Module and SEFE Search Module

Sl. No	File name	Size of file	Encryption process time	Decryption process time	SEFE search time
1	Ashaproj.txt	1.316 KB	10.7692	10.8791	29.65155
2	Stpop.txt	73.055 KB	50.8241	51.9230	437.3749
3	Statepop.txt	9,088 KB	13.1868	16.2087	191.3186
4	Totpop.txt	34,350 KB	28.7912	30.2747	310.4329

Encryption and decryption modules taken time was more or less equal. But SEFE search module taken more time than other modules. Approximately 75 MB file taken nearby 45 minutes. In fig 18 explains that the comparison between total records and retrieved records. Table 2 shows the details of size of file, total records, retrieved records and SEFE search time.

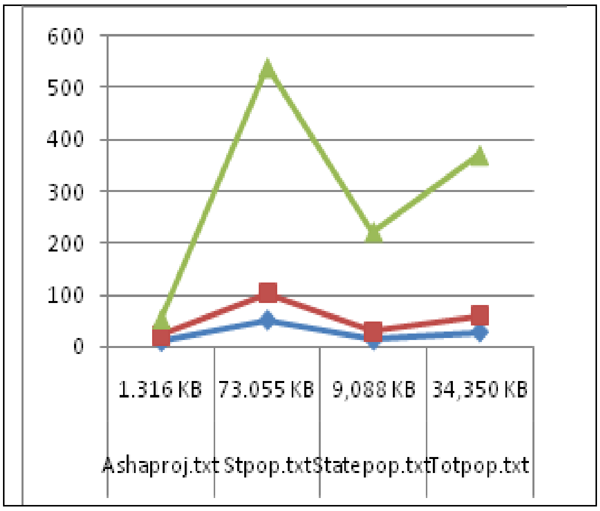


Figure 17.Comparison of Four Real Data Text File

Table 2 Details of SEFE Search Module and Data Extraction

Sl.No	File name	Size of file	Total Records	Retrieved Records	SEFE search time
1	Ashaproj.txt	1.316 KB	26,998	9,330	29.65155
2	Stpop.txt	73.055 KB	14,706	14,458	437.3749
3	Statepop.txt	9,088 KB	28,534	6,491	191.3186

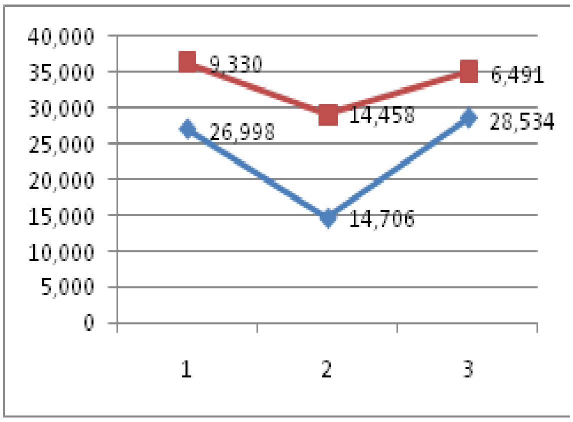


Figure 18. Comparisons between Total Records and Retrieved Records

10. CONCLUSIONS

This results work aims at developing concepts and design in a search algorithm on encrypted data to improve the security. A detailed study on Analysis on encryption methods have been carried out. A novel technique to search a given encrypted keyword an on encrypted file has been invented, an algorithm is carried out and outcome output is also in encrypted format. So authorized person only known the details of which encryption method is used for this encrypted keyword search in encrypted file. This software is designed and developed using Turbo C.

- Data encryption has many ideas. But this proposed system is new idea for keeping security of the file.
- The developed prototype has been tested data and real big data and found successfully.
- The developed prototype has been tested with unique and integration testing successfully.
- Disk encryption is a unique method and in this paper, gives some suggestions for proposed model.

11. REFERENCES

1. Deborah Russell & G.T. Gangemi, Sr, “Computer Security Basics”, 1st Edition July 1991, 0-937175-71-4, Order Number: 714, 464 pages
2. Dorothy E. Denning and Peter J.Denning, “Data Security”, Computer Science Department, Purdue University, West Lafayette, Indiana, 47907, Vol.11, No.3, September 1979
3. Prof. Christof Paar, “applied cryptography and data security”, version 2.5 | January 2005
3. Kristina Unnebrink And Ju Rgen Windeler, “Sensitivity Analysis by Worst and Best Case Assessment: Is it really sensitive?” Drug Information Journal, Vol. 33, Pp. 835–839, 1999
4. http://docs.oracle.com/cd/B10501_01/server.920/a96520/extract.htm
5. John E. Hershey, Radha Krishna R. Yarlagadda, Lawrence H. Ozarow, “PUBLIC KEY CRYPTOGRAPHIC MECHANISM”, General Electric Company, Appl. No.: 739,263, 1991
6. Maryam Hourali and Gholam Ali Montazer, “An Intelligent Information Retrieval Approach Based on Two Degrees of Uncertainty Fuzzy Ontology”, Volume 2011 (2011), Article ID 683976, 11 pages
7. Y.-C. Chang and M. Mitzenmacher, “Privacy preserving keyword searches on remote encrypted data,” in Proc. of ACNS’05, 2005
8. D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. of IEEE Symposium on Security and Privacy, 2000
9. Dawn Xiaodong Song David Wagner Adrian Perrig “Practical Techniques for Searches on Encrypted Data”, University of California, Berkeley
10. Whitepaper, “Advantages and disadvantages of EFS and effective recovery of encrypted data”, Copyright (c) 2007 ElcomSoft Co.Ltd.
11. Jin Li ; Qian Wang ; Cong Wang ; Ning Cao , et al., “Fuzzy Keyword Search over Encrypted Data in Cloud Computing” INFOCOM, 2010 Proceedings IEEE, ISBN No.978-1-4244-5836-3