DATA- FILE SECURITY ENCRYPTION ALGORITHM

Krishanu Prabha Sinha

Computer Science Department, Visvesvaraya Technological University,
East West Institute of Technology
Off Magadi Road, Vishwaneedam Post
BANGALORE 560 091, Karnataka, India
krishanusinha12@gmail.com

Abstract

Algorithm and implementation of a secure username and pass word encryption and file encryption technique is described.

Keywords: secure username; and pass word encryption.

1. Introduction

Data security, protection of username and password is required not only by Governmental agencies, private companies but private individuals as well. Cyber hackers can not only steal usernames/passwords via exploits [16] but also break through firewalls and steal important files. Since one cannot stop these thefts, one can at least encrypt the data in such a way that it is almost impossible to decrypt rendering it useless for the hackers. Data Security via steganography has been suggested by a number of authors [1-15]. With this idea in mind we describe an encryption technique for both user names/passwords and files. The method can be extended to encrypted voice transmissions. Further as the security of cloud based systems is in doubt, files encrypted as outlined here, can be stored in the clouds. The main procedure is outlined in Fig. 1. The implementation of this procedure in matlab for encryption of usernames and passwords and decryption is described in Fig. 1 and Fig. 2. File encryption and decryption programs are given in Fig. 3 and Fig. 4.

2. Data Encryption

The main reason passwords are hacked is that both user names and passwords are too short (about 8-10) characters in length. What if the user name password fields are mega bytes in size ? Password hacking via the programs being currently used will become impossible. By placing randomly the username and password characters in a 2D image matrix (of megabyte size) and encrypting the location matrix we do just that. The hacker is faced with the problem of hacking a megabyte size matrix which is a mammoth problem.

3. Secure File Encryption

The steps involved in File encryption are outlined below:

1) First read the target image into a 2D matrix. (The image may be inany format. This is most easily done in MATLAB.

2) The user name and password need not be placed contiguously. Invoke the random number and generate the coordinates where the user name and password need to be placed. Replace the pixels at these coordinates by the username and password letters (or numbers). Also store these coordinates in a file (say the location file)

3) After embedding the user name and password encrypt the image file m times (where m is a random number) using RSA

4) Encrypt also the location file n times. Convert n into hexadecimal and send this to the user. This is his key.

To De-crypt:

1) Get the hexadecimal key from user. Convert the hexadecimal to decimal (n). Decrypt the location file n times

2) Read the location file and note the value of m (the number of encryptions). Decrypt the image file m times. Now the image file can be read.

Read the coordinates of the user name and password and retrieve the entire user name and password.

```
function piccrypt
%load picture
A = imread('E:\joy\img1.jpg');
prompt = 'Enter user name';
p = input(prompt,'s')
prompt = 'Enter pass word';
q = input(prompt,'s')
uspw = strcat(p,q);

for i = 1:size(p)
      %Pick a random site
      collide = 'N';
      while (collide == 'N')
            i2 = round(rand()*size(A,2));
            i1 = round(rand()*size(A,1));
            %look for collision

            for j = 1:size(B2)
               if ((B2(j)==i1) && (B2(j+1)== i2))
                  collide = 'Y';
               end
            end
      end
      %Store the coordinates
      if (collide == 'N')
      B2(i)= i1;
      B2(i+1) = i2;
      end
      %Write into the picture
      A(i1,i2) = p(i);
end

for i = 1:size(q)
      %Pick a random site
      collide = 'N';
      while (collide == 'N')
            i2 = round(rand()*size(A,2));
            i1 = round(rand()*size(A,1));
            %look for collision

            for j = 1:size(B2)
               if ((B2(j)==i1) && (B2(j+1)== i2))
                  collide = 'Y';
               end
            end
      end
      %Store the coordinates
      if (collide == 'N')
```

```
        B2(i)= i1;
        B2(i+1) = i2;
        end
        %Write into the picture
        A(i1,i2) = pq(i);
end
%Encrypt Location File
m = [1 5 3;2 11 8;4 24 21];
s = mat2str(B1);

nnumb = reshape(double(s),3,5);
ncode = mod(m*(nnumb-32),95)+32;
scode = reshape(char(ncode),1,15);
fileID = fopen('Loc1.txt','w');
fprintf(fileID,scode);
fclose(fileID);

s = mat2str(B2);

nnumb = reshape(double(s),3,5);
ncode = mod(m*(nnumb-32),95)+32;
scode = reshape(char(ncode),1,15);
fileID = fopen('Loc2.txt','w');
fprintf(fileID,scode);
fclose(fileID);
end
```

Fig. 1 Encryption Program

```
%load picture
A = imread('E:\joy\img1.jpg');
%read location file

scode = importdata('Loc1.txt');

%decrypt string
m = [1 5 3;2 11 8;4 24 21];
ncode = reshape(double(scode),3,5);
nnumb = mod(inv(m)*(ncode-32),95);
sorig = reshape(char(nnumb+32),1,15);
B1 = str2mat(sorig);
fclose(fileID);

scode = importdata('Loc2.txt');
);
%decrypt string
%m = [1 5 3;2 11 8;4 24 21];
ncode = reshape(double(scode),3,5);
```

```
nnumb = mod(inv(m)*(ncode-32),95);
sorig = reshape(char(nnumb+32),1,15);
B2 = str2mat(sorig);
fclose(fileID);
%Retrieve User Name and Pass Word
k = 1;
for i = 1: size(B1)-1
   i1 = B1(i);
   i2 = B1(i+1);
   us(k)=A(i1,i2);
   k = k+1;
end
k = 1;
for i = 1 : size(B2)-1
    i1 = B2(i);
   i2 = B2(i+1);
   pw(k)=A(i1,i2);
   k = k+1;
end
username = mat2str(us);
password = mat2str(pw);

 end

end
```

Fig. 2 Decryption

```
function filecrypt
%load picture
A = imread('E:\joy\img1.jpg');
f = importdata('E:\A1.txt');


for i = 1:size(f)
      %Pick a random site
      collide = 'N';
      while (collide == 'N')
            i2 = round(rand()*size(A,2));
            i1 = round(rand()*size(A,1));
            %look for collision

            for j = 1:size(B2)
               if ((B2(j)==i1) && (B2(j+1)== i2))
                  collide = 'Y';
               end
            end
      end
      %Store the coordinates
```

```
    if (collide == 'N')
    B2(i)= i1;
    B2(i+1) = i2;
    end
    %Write into the picture
    A(i1,i2) = f(i);
end

%Encrypt Location File
m = [1 5 3;2 11 8;4 24 21];

s = mat2str(B2);

nnumb = reshape(double(s),3,5);
ncode = mod(m*(nnumb-32),95)+32;
scode = reshape(char(ncode),1,15);
fileID = fopen('fLoc.txt','w');
fprintf(fileID,scode);
fclose(fileID);
end
```

Fig. 3 File Encryption

```
function filedecrypt
%load picture
A = imread('E:\joy\img1.jpg');
%read location file

 scode = importdata('fLoc.txt');

%decrypt string
m = [1 5 3;2 11 8;4 24 21];
ncode = reshape(double(scode),3,5);
nnumb = mod(inv(m)*(ncode-32),95);
sorig = reshape(char(nnumb+32),1,15);
B1 = str2mat(sorig);
fclose(fileID);

scode = importdata('Loc2.txt');
);
%decrypt string
%m = [1 5 3;2 11 8;4 24 21];
ncode = reshape(double(scode),3,5);
nnumb = mod(inv(m)*(ncode-32),95);
sorig = reshape(char(nnumb+32),1,15);
B2 = str2mat(sorig);
fclose(fileID);
%Retrieve User Name and Pass Word
k = 1;
```

```
for i = 1: size(B1)-1
   i1 = B1(i);
   i2 = B1(i+1);
   f(k)=A(i1,i2);
   k = k+1;
end

fileID = fopen('Loc2.txt','w');
fprintf(fileID,f);
fclose(fileID);

end
```

4. Conclusion

We have described a technique via which vital information (username/password, important files) are randomly stored character by character in a 2D image matrix. The user keeps a highly encrypted location file. The decrypted location file is used to decrypt the image file and recover his original data. One can also split the data into multiple image files, or use audio/video files to further enhance the security.

References

[1] F. A. P. Petitcolas, , R. J. Anderson, and M. G. Kuhn. "Information Hiding-A Survey." Proceedings of the IEEE,special issue on protection of multimedia content, 87(7):1062-1078, July 1999.

[2] R. J. Anderson, and F. Petitcolas. "On the Limits of Steganography."University of Cambridge, Computer Laboratory: Cambridge, UK.September 1997. Published in IEEE Journal on Special Areas in Communications, v 16 no 4: 463-473. (May 98). http://www.cl.cam.ac.uk/~fapp2/papers/jsac98-limsteg/.

[3] H. Wang, & S.Wang, “Cyber warfare: Steganography vs.Steganalysis”, Communications of the ACM, 47:10, October 2004.

[4] R. Chandramouli, , M. Kharrazi, & N. Memon, “Image steganography and steganalysis: Concepts and Practice”, Proceedings of the 2nd International Workshop on Digital Watermarking, October 2003.

[5] L.M. Marvel, Boncelet Jr., C.G. & C. Retter, “Spread Spectrum Steganography”, IEEE Transactions on image processing, 8:08, 1999

[6] N. Johnson and S. Jajodia, “Steganalysis of images created using current steganography software,” Lecture Notes in Computer Science, Vol. 1525, pp. 273-289, 1998.

[7] M. Swanson, M. Kobayashi, and A. Tewfik, “Multimedia data embedding and watermarking technologies,” Proceedings of the IEEE, Vol. 86, No. 6, pp. 1064-1087, June 1998.

[8] R. Wolfgang, C. Podilchuk and E. Delp, “Perceptual watermarks for images and video,” to appear in the Proceedings of the IEEE, May, 1999. (A copy of this paper is availavle at: http://www.ece.purdue.edu/~ace).

[9] N.F. Johnson, & S. Jajodia, “Exploring Steganography: Seeing the Unseen”, Computer Journal, February 1998

[10] W. Bender, D. Gruhl, N. Morimoto, & A. Lu, “Techniques for data hiding”, IBM Systems Journal, Vol 35, 1996

[11] I. Cox and M. Miller, “A review of watermarking and the importance of perceptual modeling,” Proceedings of the SPIE/IST&T Conference on Human Vision and ElectronicImaging II , SPIE Vol. 3016, San Jose, CA, pp. 92-99, February1997.

[12] Sutaone, M.S., Khandare, M.V, “Image based steganography using LSB insertion technique”, IEEE WMMN, pp. 146-151, January 2008.

[13] M. Dobsicek, “Extended steganographic system”, 8th International Student Conference on Electrical Engineering, FEE CTU 2004, Poster 04.

[14] Nameer N. EL-Emam, “Hiding a large amount of data with high security using steganography algorithm”,Journal of Computer Science, Page(s): 223 – 232, April 2007.

[15] G. Sahoo, R. K. Tiwari, “Designing an Embedded Algorithm for Data Hiding using Steganographic Technique by File ybridization”, IJCSNS, Vol. 8, No. 1, pp. 228-233, January 2008

[16] see http://www.kalitutorials.net/2013/08/kali-linux.html