INTERNATIONAL JOURNAL
OF SOFT COMPUTING

SINCE 2000
MEDWELL PUBLICATIONS

A Secured Symmetric Key Encryption Technique using Images as Secret Keys

U. Sravan Goud, P.H.M.S. Raviteja, T.T.S.S.R. Phani, K. Anjesh and David Raju Kuppala
KL University, Vaddeswaram, Guntur, India

Key words: Symmetric key encryption, image, cipher text, ASCII, image

Corresponding Author:
U. Sravan Goud
KL University, Vaddeswaram, Guntur, India

Page No.: 64-67
Volume: 16, Issue 4, 2021
ISSN: 1816-9503
International Journal of Soft Computing
Copy Right: Medwell Publications

Abstract: Symmetric key cryptography is the most common cryptographic scheme which uses same key at both sender and receiver side. The plus point of symmetric key cryptography is its less computational cost when compared to public-key cryptography. In this study, a new symmetric key cryptography scheme is proposed. Rather than using typical binary secret keys this scheme uses images as secret key. Images are comprised of pixels, this work is going to exploit that property of the image. Every character in the message to be encrypted is converted into its corresponding ASCII values and image is scanned till a match is found. After a match is found the pixel co-ordinates are taken and encrypted and are written to a file. This file is transmitted to the receiver who decrypts the co-ordinates and compare with image to get the message.

INTRODUCTION

With the development of network and communication technology and the increasing number of users and many kinds of web services increased the complexity of computer networks and applications. At the same time, network security breaches occurred much more frequently and network information security is confronting an extreme crisis. Without proper security measures data might get compromised to attackers. Some attacks might be passive and others are active, even the traditional defense mechanisms are unable to meet the network requirements.

Cryptography[1] is a technique in which the data is morphed or modified or in technical term “encrypted” into and unrecognizable format known as cipher text whereas the actual real data is known as plain text. Only authorized users having the secret key can decipher the cipher text and get the plain text. Cryptography is a technique that provides security to user’s data and privacy for user and makes sure that the data is not compromised. A lot of cryptographic algorithms are available but all the encryption algorithms can be broadly classified into two types:

- Symmetric key encryption algorithms
- Asymmetric key encryption algorithms

In this study, we are using “Diffie- Hellman[2]” key exchange algorithm as a part of encryption process. Key is the most important factor in these algorithms. Encryptions with weak key are more prone to compromising of data. Strong keys means longer keys are difficult to break. Some widely used encryption schemes are: DES[3], 3DES[4] Blowfish[5], AES[6].

Diffie-Hellman key exchange: Diffie-Helman is a method for creating a mutual secret key between two individuals in a manner that the key can’t be seen by watching the communication. That is an essential distinction: You’re not sharing data amid the key trade, you’re making a key together.

The way it works is sensibly basic. A great deal of the math is the same as we find out in the public key cryptography in that a trapdoor capacity is utilized. And keeping in mind that the discrete logarithm issue is generally used (the x^y mod p business), the general procedure can be changed to utilize elliptic curve cryptography[7] also. The procedure works like this:

- Transmitter generates two prime numbers say p and g and informs the receiver about them
- Transmitter now generates a secret number say (a) but doesn't tell anyone. Instead he computes g^a mod p and sends the result of the computation to the receiver (Let's call it A)
- Receiver generates a secret number (b) and computes g^b mod p and sends the result to Transmitter (Let's call it B)
- Transmitter upon receiving B computes secret key using following function B^a mod p
- Receiver upon receiving A computes secret key using following function A^b mod p

IMPLEMENTATION

The primary preferred standpoint of symmetric key cryptography is its less computational cost when compared to its counterpart public key cryptography. In this work, a new symmetric key technique is proposed. Instead of using typical binary keys that all the standard algorithms use, proposed technique uses image as secret key. Image should be known on both the transmitter and receiver sides.

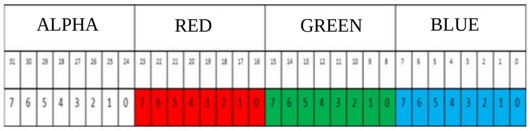
The smallest unit of an image is called a pixel, the image is made up of these pixels arranged in a order. Each pixel is made up of four components:

- Alpha
- Red
- Green
- Blue

Alpha determine the transparency of an image, whereas Red, Green, Blue show the color of the pixel. Generally, these four components are denoted as Alpha-A, Red-R, Green-G, Blue-B.

Each of these A, R, G, B takes a value from the range 0-255 where 0 and 255 are inclusive. 0 means component is absent and 255 means the component is fully present. Since integers till 256 can be represented in 8 bit format so each of A, R, G, B also can be represented in 8 bits and since there are four components the number of bits that are required to store all the four components is 32 bit or 4 bytes (Fig. 1).

2D images: Images have height and width and are generally denoted as pixels. Figure 2 shows details like dimensions, type and size.



ALPHA								RED								GREEN								BLUE							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Fig. 1: 8 bit representation of ARGB

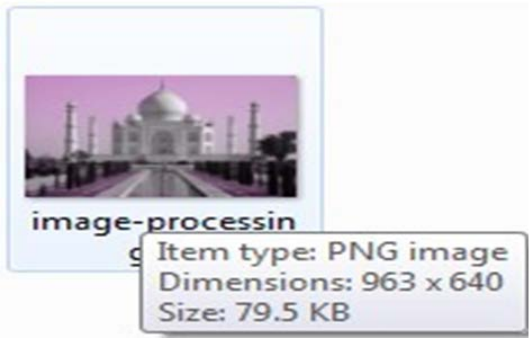


image-processin
g

Item type: PNG image
Dimensions: 963 x 640
Size: 79.5 KB

Fig. 2: 2D image

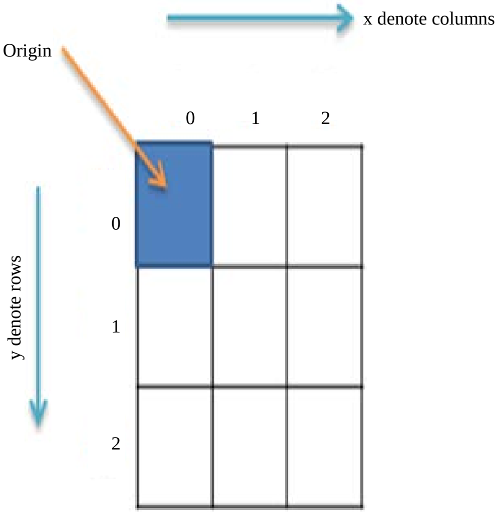


Fig. 3: Rows and columns of image

In general dimensions of an image are represented in width x height format. In above picture, we can see dimensions as 963×640 means width is 963 and height is 640.

Representing 2D image pixels: For any 2D image pixels are arranged in rows and columns. Origin (Starting point of an image) is at the coordinate (0, 0) (Fig. 3).

We can denote a pixel coordinate (x, y) as $P_{x,y}$ (A, R, G, B) where x, y are actual coordinates of pixel and A, R, G, B are Alpha, Red, Green and Blue components of pixel.

In this study, we make use of these A, R, G, B values of the pixel coordinates. We take the plain text and consider each character of the plain text and take the ASCII value of that character and now the image is scanned till we get a height that matches the ASCII value, at this coordinate the A, R, G, B values are taken and encrypted using key generated from diffie-hellman algorithm and written to a file. This file is transmitted to the receiver who decrypts the A, R, G, B values using key and scan the image till the match is found for these A, R, G, B values and when a match is found the height of that coordinate is taken and converting it to character format we get the plain text.

Algorithm

At sender side:

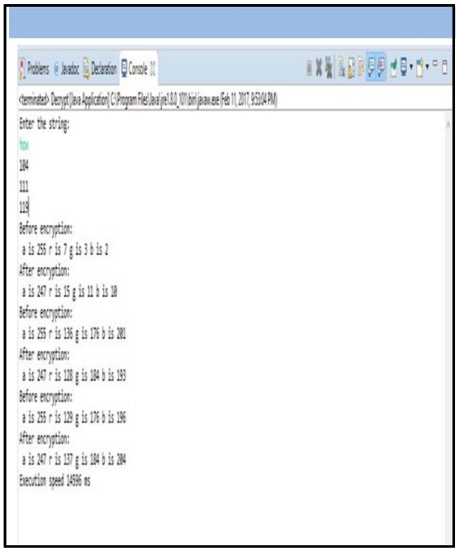
- Consider the plain text and take each character and find its corresponding ASCII value
- Store all these values into an array
- Now start scanning the image till a match is found between the height of pixel in image and ASCII value
- Now take the A,R,G,B values of that pixel and encrypt them using key generated from diffie-hellman key exchange algorithm and write it into a file
- Repeat steps 4 and 5 till all the values in array are competed

At receiver side:

- Receive the file which contains all the encrypted values
- Take each value and decrypt them using key generated from Diffie-Hellman key exchange algorithm
- Now start scanning the image till a match is found between the ARGB values of pixel and ARGB values in the file
- Take the height of that pixel and convert i
- t into character
- Repeat 3 and 4 steps till all values in file are decrypted

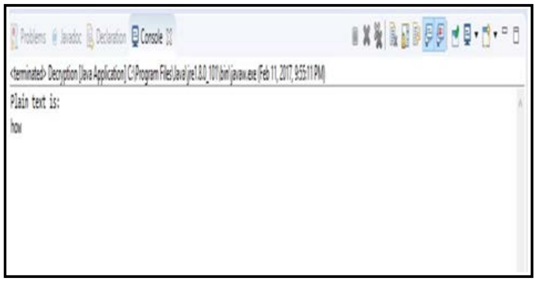
Example: At sender side: for simplicity let us consider the text to be sent is h, now the text is read and the ASCII value of h is taken, that is ‘104’. The image that we are using is scanned till we get a coordinate with height as 104 like (98,104).

For above coordinate, we consider the A, R, G, B values which might be A-255, R-20, G-10, B-50. Now these values are encrypted using key generated through Diffie-Hellman algorithm. Let’s say the shared key is 8. Encryption is nothing but performing XOR operation between the shared key and all the A, R, G, B values.



```
<terminated> Decrypt [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Feb 11, 2017, 9:53:04 PM)
Enter the string:
how
104
111
119
Before encryption:
 a is 255 r is 7 g is 3 b is 2
After encryption:
 a is 247 r is 15 g is 11 b is 10
Before encryption:
 a is 255 r is 136 g is 176 b is 201
After encryption:
 a is 247 r is 128 g is 184 b is 193
Before encryption:
 a is 255 r is 129 g is 176 b is 196
After encryption:
 a is 247 r is 137 g is 184 b is 204
Execution speed 14596 ms
```

Fig. 4: Encryption



```
<terminated> Decryption [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Feb 11, 2017, 9:55:11 PM)
Plain text is:
how
```

Fig. 5: Decryption

In this case, the encrypted values are A- 247, R- 28, G-2, B-58. These values are now written to a file. This file is transmitted to receiver.

At receiver side: The file containing encrypted values A-247, R- 28, G-2, B-58 is received. These encrypted values are decrypted using shared key generated by Diffie-Hellman algorithm by performing XOR operation between shared key and A, R, G, B values.

The values after decrypting are A-255, R-20, G-10, B-50. The image is used as key and scanned till we find a coordinate that has same A, R, G, B as above which is (90,104) and for this coordinate we consider the height which is 104. This height is converted to character format which gives ‘h’. In such a way, the plain text can be encrypted and decrypted at sender and receiver side, respectively (Fig. 4 and 5).

CONCLUSION

In this study, we have discussed a new technique using image as secret key which is very different from the conventional encryption techniques like DES, 3-DES, AES, Blowfish, etc. This technique can provide more security, since, we are using image as secret key which of very large size compared to binary keys, so, it is hard to crack.

ACKNOWLEDGMENT

This research is supported by the Department of Science and Technology, India through the fund sanctioned for improvement of Science and Technology infrastructure, at Department of CSE, K.L University by order number SR/FST/ESI-332/2013.

REFERENCES

01. Forouzan, B., 2007. Cryptography and Network Security. McGraw-Hill, New Delhi, India,.
02. Kaushik, A., 2013. Extended Diffie-Hellman algorithm for key exchange and management. Int. J. Adv. Eng. Sci., 3: 67-70.
03. Gupta, N., 2012. Implementation of optimized des encryption algorithm upto 4 round on spartan 3. Int. J. Comput. Technol. Electron. Eng. (IJCTEE.), 2: 82-86.
04. Singh, G., 2013. A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. Int. J. Comput. Appl., 67: 33-38.
05. NehaKhatri-Valmik, M. and V.K. Kshirsagar, 1994. Blowfish algorithm. IOSR. J. Comput. Eng., 16: 80-83.
06. Soumya, K. and G.S. Kishore, 2013. Design and Implementation of Rijndael encryption algorithm based on FPGA. Int. J. Comput. Sci. Mob. Comput., 2: 120-127.
07. Agrawal, H. and P.R. Badadapure, 2016. A survey paper on elliptic curve cryptography. Int. Res. J. Eng. Technol. (IRJET.), Vol. 3, No. 4.