

A SECURED SYMMETRIC KEY ENCRYPTION TECHNIQUE USING IMAGES AS SECRET KEYS

Sravan Goud Utkam¹, Raviteja Parasa², Anjesh Kondapaneni³,
 Phani Raj Kumar Tulabandula⁴, David Raju Kuppala⁵, Amudhavel J⁶
^{1,2,3,4,5,6}Department of CSE, KL University, Andhra Pradesh, India
^{1,2,3,4,5,6}sra1@gmail.com

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 paper 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.

Keywords: symmetric key encryption, image, cipher text.

1. 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:

1. Symmetric key encryption algorithms.
2. Asymmetric key encryption algorithms.

In this paper 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].

1.1 Diffie-Hellman Key Exchange:

Diffie-Hellman 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 $x^y \bmod p$ business), the general procedure can be changed to utilize elliptic curve cryptography[7] also.

The procedure works like this:

1. 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 $g^a \bmod p$ and sends the result of the computation to the receiver. (Let's call it **A**).
- Receiver generates a secret number (b) and computes $g^b \bmod p$ and sends the result to Transmitter. (Let's call it **B**).
- Transmitter upon receiving **B** computes secret key using following function $B^a \bmod p$.
- Receiver upon receiving **A** computes secret key using following function $A^b \bmod p$.

2. 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.

- 1) Alpha.
- 2) Red.
- 3) Green.
- 4) 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 to 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 (refer with: Figure 1) 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.

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

Figure 1. 8 Bit representation of ARGB

2.1 2D Images

Images have height and width and are generally denoted as pixels. Below Figure shows details like Dimensions, type and size (refer with: Figure 2).

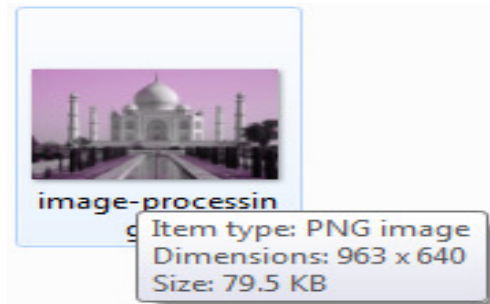


Figure 2. 2D Image

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

2.2 Representing 2D Image Pixels

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

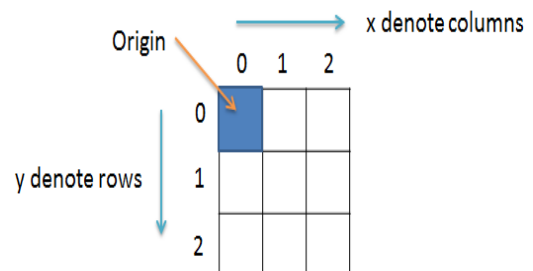


Figure 3. Rows and Columns of image

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 paper 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.

2.3 Algorithm:

At sender side:

1. Consider the plain text and take each character and find its corresponding ASCII value.
2. Store all these values into an array.
3. Now start scanning the image till a match is found between the height of pixel in image and ASCII value.
4. 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.
5. Repeat steps 4 and 5 till all the values in array are completed.

At receiver side:

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

2.4 Example:

At sender side:

1. 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**'.
2. The image that we are using is scanned till we get a coordinate with height as 104 like (98,104).
3. For above coordinate we consider the A, R, G, B values which might be A-255, R-20, G-10, B-50.
4. Now these values are encrypted using key generated through diffie-hellman algorithm. Let's say the shared key is 8.
5. Encryption is nothing but performing XOR operation between the shared key and all the A, R, G, B values.
6. In this case the encrypted values are A- 247, R- 28, G- 2, B- 58 (refer with: Figure 4) . These values are now written to a file.
7. This file is transmitted to receiver.

At receiver side:

1. The file containing encrypted values A- 247, R- 28, G-2, B-58 is received.
2. 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.
3. The values after decrypting are A- 255, R- 20, G- 10, B- 50.
4. 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.
5. This height is converted to character format which gives '**h**' (refer with: Figure 5).

In such a way the plain text can be encrypted and decrypted at sender and receiver side respectively.

3. Results

```
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 is 1459 ms
```

Figure 4. Encryption

```
Plain Text is:
how
```

Figure 5. Decryption

4. Conclusion

In this paper 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.

Acknowledgement

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

References

- [1] B. A. Forouzan, "Cryptography and Network Security", Special Indian Edition, The McGraw-Hill companies, New Delhi, 2007.
- [2] Akhil Kaushik & Satvika, "Extended Diffie-Hellman Algorithm for key exchange and management", International Journal of Advances in Engineering Sciences Vol.3 (3), July, 2013 e-ISSN: 2231-0347 Print-ISSN: 2231-2013.
- [3] Nimmi Gupta, "Implementation of Optimized DES Encryption Algorithm upto 4 Round on Spartan 3", International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2 , Issue 1
- [4] Gurmeet Singh & Supriya, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security", International Journal of Computer Applications (0975 – 8887) Volume 67– No.19, April 2013
- [5] Ms NehaKhatri & Prof. V. K Kshirsagar, "Blowfish Algorithm", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727Volume 16, Issue 2, Ver. X (Mar-Apr. 2014), PP 80-83 .
- [6] K. Soumya & G. Shyam Kishore, "Design and Implementation of Rijndael Encryption Algorithm Based on FPGA", IJCSMC, Vol. 2, Issue. 9, September 2013, pg.120 – 127.
- [7] Himja Agrawal & Prof.P.R.Badadapure, "A Survey Paper On Elliptic Curve Cryptography", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 03 Issue: 04 | Apr-2016 www.irjet.net p-ISSN: 2395-0072.

