A Hybrid Cryptosystem of Image and Text Files Using Blowfish and Diffie-Hellman Techniques

Tapan Kumar Hazra
Department of Information Technology
Institute of Engineering & Management, Salt Lake, Kolkata, INDIA
tapankumar.hazra@iemcal.com

Anisha Mahato
Department of Information Technology
Institute of Engineering & Management, Salt Lake, Kolkata, INDIA
anisha.mahato11@gmail.com

Arghyadeep Mandal
Department of Information Technology
Institute of Engineering & Management, Salt Lake, Kolkata, INDIA
arghyadeep11@gmail.com

Ajoy Kumar Chakraborty
Department of Information Technology
Institute of Engineering & Management, Salt Lake, Kolkata, INDIA
akc_000@rediffmail.com

***Abstract*— In this paper we have proposed a new algorithm of encrypting and decrypting images and text files. The proposed method is implemented by combining the concepts of Diffie Hellman algorithm and Blowfish algorithm. In this new technique at first a computer user will encrypt a file using a secret key generated by blowfish algorithm. Then using Diffie-Hellman protocol a shared private key will be generated for two computer users who are trying to communicate over an insecure channel. Now if the second user wants to decrypt the file encrypted by the first user he/she has to use the shared key for permission. Once the permission is granted, the file can be decrypted using blowfish algorithm.**

Keywords*— *Hybrid cryptosystem, Diffie Hellman protocol, blowfish algorithm, image encryption, symmetric key, asymmetric key

I. INTRODUCTION

With the growth in technology, the web attacks are also increasing in a significant rate. Nowadays every possible machine can be hacked to get access of confidential data. This problem is relatable to organizations like schools, private organizations, government offices, financial institutions like banks etc. Thus the need for security is always a major concern. Security is required in any field which needs data exchange (especially sensitive data like transactions, signals etc). The data, if intercepted by any intruder, can cause irreparable damages. There are many cryptographic methods which are being developed in order to make the data secure from various attacks and we must encrypt the data in a logical way before it is transmitted to ensure its integrity.

Whenever the term encryption comes into consideration we particularly discuss two types of algorithms- symmetric and asymmetric cryptosystem. In case of symmetric algorithms a single shared key is used to encrypt and decrypt data whereas in asymmetric algorithm two keys are generated (public key and private key) in the process. Here in this paper we have combined the advantages of these two different sets of algorithms to construct a new one which would help in encryption and decryption of images and text files. We have used two protocols in this process: Diffie Hellman [1] and Blowfish [2]. These two are very well known and standard algorithms which have been used in practical world before.

There have been many modifications in these algorithms in order to increase security. As we have seen in [5], there was an improvement in plaintext attacks in Diffie Hellman. We have modified these algorithms taking the advantages of both of them.

The proposed algorithm can be applied to text files (.txt) as well as image **files (**.jpg, .png etc.). It is a block cipher algorithm and when applied to a block of text characters, the corresponding cipher text became unreadable because it contains strange characters, but when applied to block of image data corresponding to pixel intensities, the cipher image is not opened because of violation in image file format. The image encryption technique specified in [7], produces a cipher image that can be opened but information content is lost.

In this paper **D**iffie **He**llman and blowfish have been defined in section1.1. In section 2 we discuss the encryption and decryption process of the proposed algorithm in a step wise format. In section 3 we provide the results and analysis of the algorithm followed by advantages and future work in section 4. Finally the conclusion is given in section 5 followed by references.

II. BACKGROUND STUDY

A. Algorithm used

Diffie Hellman Algorithm: The Diffie-Hellman protocol [6] is a method for two computer users to generate a shared private key with which they can then exchange information across an insecure channel.

Blowfish Algorithm: The blowfish algorithm [4] which is considered as one of the simplest and fastest symmetric block cipher that is used to encrypt the text/image file by dividing it into blocks of equal sizes [3] [7]. There are many block cipher algorithm which are used for this purpose like one using pseudo random permutations [8-10].

B. Maintaining the Integrity and quality in Encryption

Confusion and diffusion are two important aspects that judge encryption quality. The Diffusion implies that if we alter a single character of the plaintext, then several characters of the

978-1-5386-2215-5/17/$31.00 ©2017 IEEE

cipher text will be changed accordingly. Mixing transformations is one of the ways, recommended by Shannon, to attain confusion and diffusion in the security system. Substitution -permutation networks and Feistel networks both incorporate the idea of mixing transformations. However, they both vary in certain areas of their construction details, thus making them suitable for different scenarios. In this paper, we have proposed an encryption scheme which combines the utilities of both block cipher and secret key share to achieve encryption and decryption of text and image files. The randomness of the encrypted result is increased by inherent characteristics of Blowfish algorithm.

III. Proposed Algorithm

Let user 1 is the sender and user 2 is the recipient.

Step 1: First, both the users agree upon a prime number **p** and another number **g** that has no factor in common. Note that **g** is also known as the generator and **p** is known as prime modulus.

Both are public keys. So a third person sitting in between and listening to this communication also gets to know **p** and **g**.

Step 2: Now user 1 takes a private key **x.** So, user 1 calculates a key **R1= (g^x) mod p**.

Step 3: User 1 generates a secret key and cipher and then encrypts the file using the secret key and cipher generated by blowfish algorithm.

```
[
keyGenerator = KeyGenerator.getInstance("Blowfish");
 secretKey = keyGenerator.generateKey();
cipher = Cipher.getInstance("Blowfish");
]
```

Step 4: Now User 2 wants to decrypt the file. User 2 takes a private key **y.**

So user 2 calculates a key **R2= (g^y) mod p**.

Step 5: Now user 2 wants permission for decryption from user 1. So both of them share **R1** and **R2** with each other through the insecure channel of communication. So these values become public.

Step 6: User 1 calculates final key **k1= (R2^x) mod p**. User 2 calculates **k2=(R1^y) mod p**.

Step 7: If the values of **k1** and **k2** match then only user 2 gets the permission of decryption. So user 1 sends the secret key for Blowfish to user 2. Then user 1 sends the encrypted file to user 2.

Step 8: Now user 2 decrypts the file using Blowfish algorithm.

A. Block Diagram

The block diagrams presented in Fig. 1. and Fig. 2. gives the complete explanation of the proposed system developed that consists steps for generation of the secret key, encryption of source file, sharing of the key, decryption of encrypted file received through insecure channel.

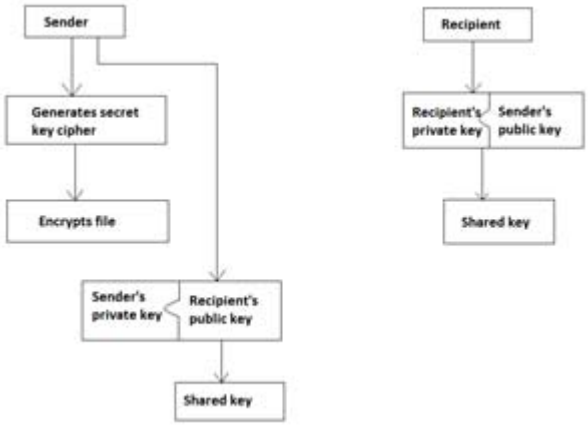

Fig. 1. Encryption process and shared key generation

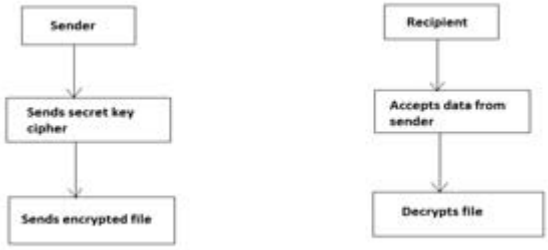

Fig. 2. When shared keys matches

B. Example of Shared Key Calculation

Let us consider, **p** and **g** be equal to 19 and 7 respectively.

User 1 takes private key x = 6 and user 2 takes private key y=9.

Now user 1 calculates public key R1 = (g^x) mod p= (7^6) mod 19=1

User 2 also calculates public key R2= (g^y) mod p = (7^9) mod 19=1

For user 1 shared key k1= (R2^x) mod p = (1^6) mod 19=1

For user 2 shared key k2= (R2^x) mod p= (1^9) mod 19=1

So k1 = k2.

Now if an eavesdropper sends a number 3 when user 1 asks for shared key, the request for encrypted file will be denied.

C. How are Attacks Intercepted

When the file is being sent to the recipient, if somehow it gets attacked, the attacker still won't be able to see the content as the text/image will remain encrypted because the attacker don't have the secret key of blowfish encryption.

IV. Results And Analysis

When the file is being sent to the recipient, if somehow it gets attacked the attacker still won't be able to see the content as the text/image will remain encrypted and the attacker won't have the secret key of blowfish encryption.

```
OSPF:
Open Shortest Path First

Router0:


Router>en
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fa0/0
Router(config-if)#ip address 192.168.12.1 255.255.255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#interface se0/1/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/1/0, changed state to down
Router(config-if)#exit
Router(config)#interface se0/1/1
Router(config-if)#ip address 12.0.0.2 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/1/1, changed state to down
Router(config-if)#
Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up
```

Fig. 3. A simple text file with extension '.txt' to be encrypted

Fig. 4. The encrypted file

Fig.4. shows the cipher text that is obtained as a result of encryption of the text file shown in Fig. 3, which when decrypted produces exactly same as the original file, and the output is shown in Fig.5. While in the image files, the content is numeric values in the domain [0, 255]. When the proposed method is applied, the numeric content of encrypted files go out of range and violate the specification for image file format, so the display is not supported by any digital computers. The encrypted output is shown in Fig.8, which is same for both input images corresponding to color image in Fig. 6 and grey scale image in Fig. 7. But the objective of our proposed algorithm remains unharmed as the encrypted file is not readable but the file gets properly decrypted and images are correctly shown in Fig.9 and Fig.10 which are exact replica of the original image files that are encrypted.

```
OSPF:
Open Shortest Path First

Router0:


Router>en
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fa0/0
Router(config-if)#ip address 192.168.12.1 255.255.255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#interface se0/1/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/1/0, changed state to down
Router(config-if)#exit
Router(config)#interface se0/1/1
Router(config-if)#ip address 12.0.0.2 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/1/1, changed state to down
Router(config-if)#
Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up
```

Fig. 5. The decrypted file

Application of the same algorithm on image file with extension .jpeg are shown in Fig. 6 - Fig. 10.

Fig. 6. The color image file to be encrypted

Fig. 7. The gray scale image file to be encrypted

Fig. 8. The unsupported image file error message when any encrypted file is displayed

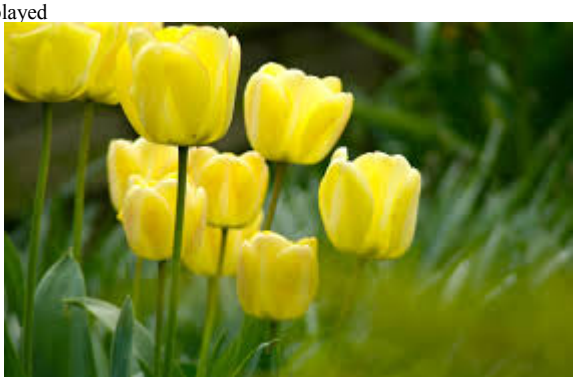

Fig. 9. Decrypted file that if exactly identical to original file shown in Fig. 6.

Fig. 10. Decrypted file that if exactly identical to original file shown in Fig. 7.

TABLE I ANALYSIS OF SPACE USAGE

File Type	Original file size	Encrypted file size	Decrypted file size
Text file(.txt) shown in Fig. 3	1.33 KB	Text file(.txt)	1.33 KB
Color image file(.jpeg) shown in Fig. 6	1.03 MB (10,87,794 bytes)	1.03 MB (10,87,800 bytes)	1.03 MB (10,87,794 bytes)
Gray scale image file(lena.jpeg) shown in Fig. 7	65.8 KB (67,438 bytes)	65.8 KB (67,440 bytes)	65.8 KB (67,438 bytes)

Table 1 gives the amount of memory needed to store the different files on which the algorithm is performed. We see that the encrypted file size is not of much different from the original file size. It could be a major concern if the encrypted file size would increase to a large amount. Thus here wastage of space is also taken into consideration.

A. *Advantages of present Model*

The proposed algorithm claims to be better than the existing algorithms since it takes the advantage of generating a variable length key using the Blowfish algorithm and at the same time work on its disadvantages. Blowfish can't provide authentication and non-repudiation. This drawback is taken care by Diffie Hellman shared key generation technique which ensures the file doesn't go to wrong hands while transferring it over public network. The file is decrypted only if the key matches. Even if the file goes in wrong hands somehow, it will remain encrypted and not readable.

Another advantage is that the system works for multiple formats of images (.jpg, .png, .tiff etc.) and text files.

B. *Future Scope*

Our aim always would be to modify this algorithm to increase the level of security. To increase the security aspect, future enhancement can try to prevent replay attacks. Because if someone is repeatedly trying to access the encrypted file with wrong keys, it might very well be possible that the user is trying permutation and combination to get the correct secret base. So we can include timestamp for this reason. If multiple timestamps are being received from a single source it will be easy to comprehend that request is probably coming from an attacker.

V. CONCLUSION

With the advent of internet, security of data that is being transferred online has become very difficult to ensure. Diffie Hellman and Blowfish methods when individually applied, faces a lot of security threats like man in the middle attack, data authentication etc. But as we know the cryptosystems used in today's world is based on these two basic algorithms, so we cannot compromise on these. Thus, the proposed system attempts to ensure that the data is read by only intended user by providing a two level security system and overcoming most of the shortcomings faced by existing algorithms.

REFERENCES

[1] Ritu Tripathi, Sanjay Agrawal, "Comparative Study of Symmetric and Asymmetric Cryptography Techniques," International Journal of Advance Foundation and Research in Computer (IJAFRC), Volume 1, Issue 6, June 2014. ISSN 2348 – 4853.

[2] Malek Jakob Kakish, "Security Improvements To The DIFFIE-HELLMAN Scheme," International journal of Engineering and Technology July 2011892, pp.68–73.

[3] Manku, KV Saikumar, and K. Vasanth. "Blowfish encryption algorithm for information security." *ARPN Journal of Engineering and Applied Sciences* 10.10 (2015): 4717-4719.

[4] NehaKhatri–Valmik, Ms, and V. K. Kshirsagar. "Blowfish Algorithm." *IOSR Journal of Computer Engineering* 16.2 (1994).

[5] Sehgal, Parth, Nikita Agarwal, Sreejita Dutta, and PM Durai Raj Vincent. "Modification of Diffie-Hellman Algorithm to Provide More

Secure Key Exchange." *International Journal of Engineering & Technology*: 0975-4024.

[6] Forouzan, A. Behrouz. *Data communications & networking (sie)*. Tata McGraw-Hill Education, 2006.

[7] T. K. Hazra and S. Bhattacharyya, "Image encryption by blockwise pixel shuffling using Modified Fisher Yates shuffle and pseudorandom permutations," *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, 2016, pp. 1-6. doi: 10.1109/IEMCON.2016.7746312.

[8] T. K. Hazra, R. Ghosh, S. Kumar, S. Dutta and A. K. Chakraborty, "File encryption using Fisher-Yates Shuffle," 2015 International Conference and Workshop on Computing and Communication (IEMCON), Vancouver, BC, 2015, pp. 1-7.doi: 10.1109/IEMCON.2015.7344521

[9] Sreejit Roy Chowdhury, Tapan Kumar Hazra, Ajoy Kumar Chakroborty, " Image Encryption using pseudo random permutations," American Journal of Advanced Computing, 1.1 (2014), doi: http://dx.doi.org/10.15864/ajac.v1i1.2

[10] Li, Chengqing. "Cracking a hierarchical chaotic image encryption algorithm based on permutation." *Signal Processing* 118 (2016): 203-210.