A Hybrid Cryptosystem of Image and Text Files

Using Blowfish and Diffie-Hellman Techniques

Bibin Jaimon
B4E028

Guided by:

Prof. Jobin Joy

CSE Department

Govt College of Engineering Kannur

February 24, 2018

Outline

1. Introduction
2. Background Knowledge
3. Motivation
4. Proposed System
    - Blowfish Algorithm
    - Diffie Hellman Algorithm
    - Proposed Algorithm
5. Results and Analysis
6. Future Enhancements
7. Conclusion
8. References

Introduction

- Introduces a new hybrid crypto system for images and text files.
- Combines the concepts of Diffie Hellman and Blowfish algorithm.
- First encrypt a file using a secret key generated by blowfish algorithm.
- Then using Diffie-Hellman protocol users has been authenticated.

Background Knowledge I

What is Hybrid cryptosystem?

- Combines the convenience of a public-key cryptosystem with the efficiency of a symmetric-key cryptosystem.
- Encryption integrity is maintained by confusion and diffusion.

Symmetric key and Asymmetric key Cryptosystem

- Symmetric algorithms: use the same key for both encryption and decryption
- Asymmetric algorithms: use different keys for encryption and decryption.

Motivation

- In Blowfish algorithm the data can be encrypted using a key.
- Then the encrypted data and key send to the recipient without any user authentication.
- So, Introduces a hybrid crypto system combining the concept of Diffie Hellman key exchange protocol.
- DH protocol is used for authentication purpose.

Related Work I

Image Encryption and Decryption Approach using Pixel Shuffling.[1]

- Encryption and Decryption of an image by pixel shuffling.
- Find a pattern to shuffle the pixels of the image. that is considered as secret key.
- Then shuffle the image using the pattern.
- If the secret pattern is lost then the image can easily decrypted.

Related Work II

Image encryption by blockwise pixel shuffling using Fisher Yates shuffle and Psudorandom permutations

- An image encryption technique using Fisher Yates shuffling along with psudorandom permutation.
- produced an encrypt file whose size was double than that of file before encryption.
- Waste-age of memory is main problem when this approach is used.

Proposed System

- The Hybrid cryptosystem consist of cobination of two protocols.
  - Blowfish Algorithm
  - Diffie Hellman Algorithm

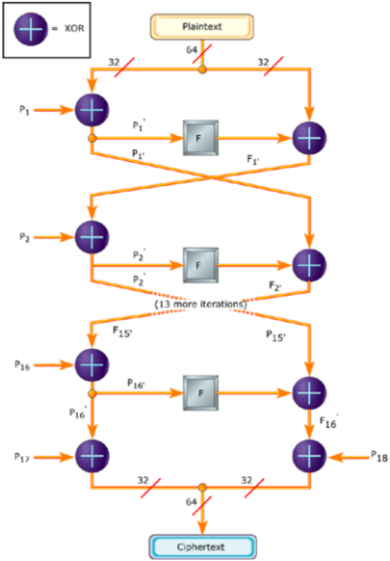
Blowfish Algorithm

- Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms.
- In Blowfish encryption, only one key is used for both encryption and decryption of data.
- algorithm is divided into two parts.
  1. Key Expansion
  2. Data Encryption

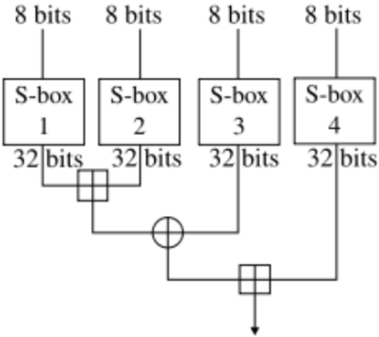
Key Expansion - Blowfish Algorithm

- Initializing the P-array and S-boxes with values derived from the hexadecimal digits of pi
- The secret key is then XORed with all the P-entries in order.
- A 64-bit all-zero block is then encrypted with the algorithm as it stands.
- The resultant cipher text replaces P1 and P2. The same cipher text is then encrypted again with the new sub keys.
- The new cipher text replaces P3 and P4. This continues, replacing the entire P-array and all the S-box entries.
- the Blowfish encryption algorithm will run 521 times to generate all the subkeys.

Data Encryption - Blowfish Algorithm



F –Function in Feistel Network



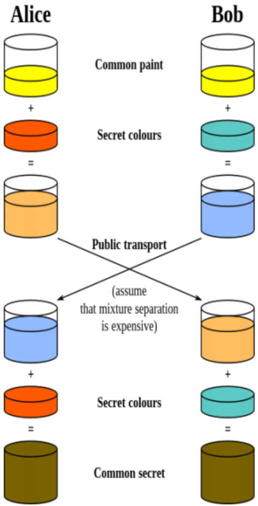
Disadvantage of Blowfish Algorithm

- It must get key to the person out of band specifically not through the unsecured transmission channel.
- Blowfish can't provide authentication and non-repudiation as two people have same key.

Diffie Hellman Algorithm

- Invented by Whitefield Diffie and Martin Hellman in 1976.
- DH is a mathematical algorithm that allows 2 parties to get shared secret key using the insecure communication channel.
- We don't commonly use DH to encrypt data. It is a strategy for secure exchange of the keys that encrypt data
- DH is usually utilized when you encrypt data on the Web utilizing either SSL (Secure Socket Layer) or TLS (Transport Layer Security)

Illustration of DH with colors



Advantages of Diffie Hellman Algorithm

- The security factors with respect to the fact that solving the discrete logarithm is very challenging.
- That the shared key (i.e. the secret) is never itself transmitted over the channel.
- DH shared key generation technique which ensure the file does not go to wrong hands.

Proposed Algorithm I

1) Both the users agree upon a prime p and another number g that has no factor in common.

2) User 1 takes a private key x and calculates a key $R1 = g^x modp$

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");

Proposed Algorithm II

4) User 2 takes a private key y and calculates a key R2= (g^y) mod p.

5) Both users share R1 and R2 with each other through the insecure channel. So these values become public.

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

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.

8) User 2 decrypts the file using Blowfish algorithm.

Block Diagram I

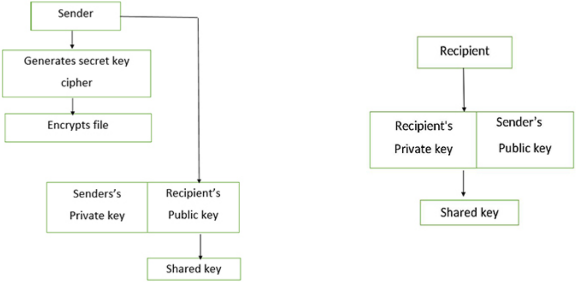


Figure: Encryption process and shared key generation

Block Diagram II

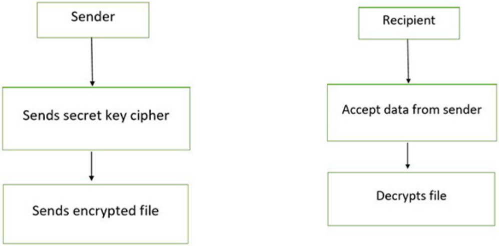
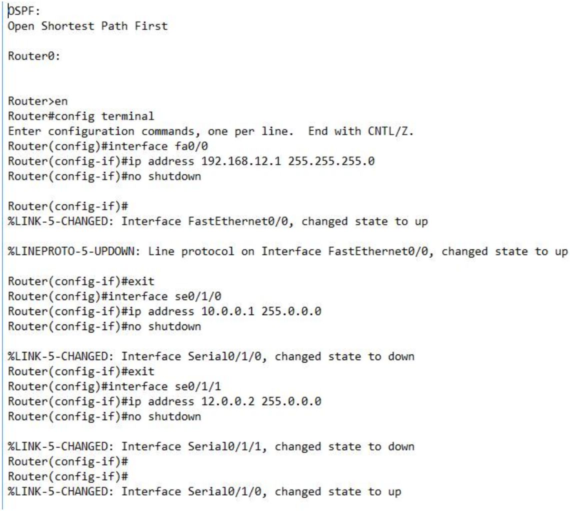


Figure: When shared keys matches

How are Attacks Intercepted

- File is sent through insecure channel.
- Attacker intercepts the file.
- The attacker won't be able to see the content as the text/image.
- The contents remain encrypted because the attacker don't have the secret key of blowfish encryption.

Results and Analysis I



```
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
```

Figure: A simple text file with extension '.txt' to be encrypted

Results and Analysis II

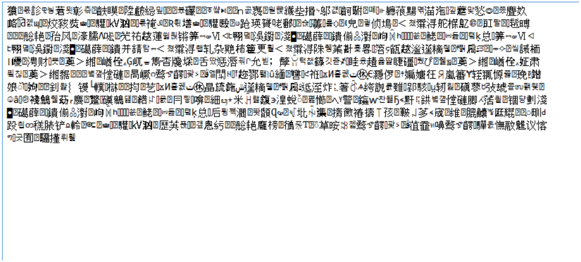


Figure: The encrypted file

- Fig.4. shows the cipher text that is obtained as a result of encryption of the text file shown in Fig. 3.
- When decrypted produces exactly same as the original file.

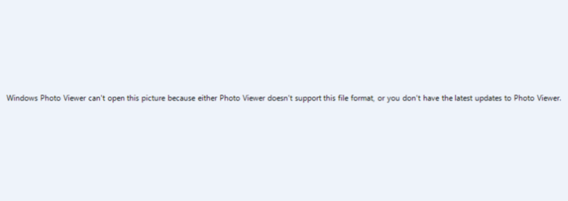
Results and Analysis III

- In the image files, the content is numeric values in the domain [0, 255].
- The numeric content of encrypted files go out of range and violate the specification for image file format.
- The display is not supported by any digital computers.
- Encrypted file is not readable but the file gets properly decrypted.

Result and Analysis IV

Figure: The color image file to be encrypted

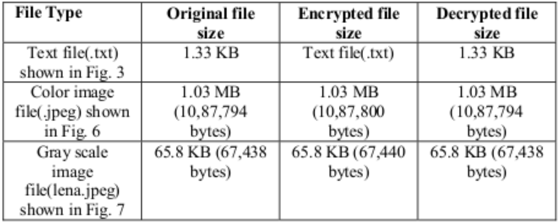
Result and Analysis IV



Windows Photo Viewer can't open this picture because either Photo Viewer doesn't support this file format, or you don't have the latest updates to Photo Viewer.

Figure: The unsupported image file error message when any encrypted file is displayed

Results and Analysis V



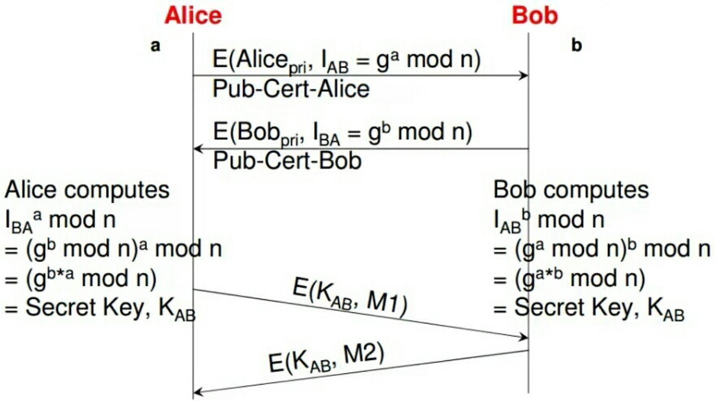
File Type	Original file size	Encrypted file size	Decrypted file size
Text file(.txt) shown in Fig. 3	1.33 KB	Text file(.txt)	1.33 KB
Color image file(.jpeg) shown in Fig. 6	1.03 MB (10,87,794 bytes)	1.03 MB (10,87,800 bytes)	1.03 MB (10,87,794 bytes)
Gray scale image file(lena.jpeg) shown in Fig. 7	65.8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

Figure: Memory Requirement

Table 1 gives the amount of memory needed to store the different files on which the algorithm is performed.

Future Enhancements I

Station To Station (STS) Protocol



Alice

Bob

a

b

$E(Alice_{pri}, I_{AB} = g^a \bmod n)$

Pub-Cert-Alice

$E(Bob_{pri}, I_{BA} = g^b \bmod n)$

Pub-Cert-Bob

Alice computes

$I_{BA}{}^a \bmod n$

$= (g^b \bmod n)^a \bmod n$

$= (g^{b*a} \bmod n)$

$=$ Secret Key, K_{AB}

Bob computes

$I_{AB}{}^b \bmod n$

$= (g^a \bmod n)^b \bmod n$

$= (g^{a*b} \bmod n)$

$=$ Secret Key, K_{AB}

$E(K_{AB}, M1)$

$E(K_{AB}, M2)$

Conclusion

- Takes the advantage of generating a variable length key using the Blowfish algorithm.
- The file is decrypted only if the key matches.
- Overcome the limitations of blowfish such as user authentication.
- Overcomes most of the shortcomings faced by existing algorithms.

References I

[1] A. K. Prusty, A. Pattanaik, and S. Mishra, "An image encryption & decryption approach based on pixel shuffling using arnold cat map & henon map," *2013 International Conference on Advanced Computing and Communication Systems*, Oct 2014.

[2] T. K. Hazra, A. Mahato, A. Mandal, and A. K. Chakraborty, "A hybrid cryptosystem of image and text files using blowfish and diffie-hellman techniques," *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*, Oct 2017.

[3] M. Mukhedkar, P. Powar, and P. Gaikwad, "Secure non real time image encryption algorithm development using cryptography & steganography, applications, and challenges," *2015 Annual IEEE India Conference (INDICON)*, March 2016.

References II

[4] S. Hassene and M. N. Eddine, “A new hybrid encryption technique permuting text and image based on hyperchaotic system,” *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, July 2016.

QUESTIONS?

Thank You