ISSN: 0975-766X
CODEN: IJPTFI
Available Online through
Research Article
www.ijptonline.com

IMAGE ENCRYPTION AND DECRYPTION USING BLOWFISH ALGORITHM WITH RANDOMNUMBER GENERATOR

Anjaneyulu GSGN*[1], Pawan Kumar Kurmi[2], Rahul Jain[2]
[1]Professor, Applied Algebra Division, School of Advanced Sciences, VIT University, Vellore-14, Tamilanadu, India.
[2]MCA, School of Information Technology Engraining, VIT University, Vellore-14, Tamilnadu, India.
Email: anjaneyulu.gsgn@vit.ac.in

Received on 09-12-2014 *Accepted on 31-12-2014*

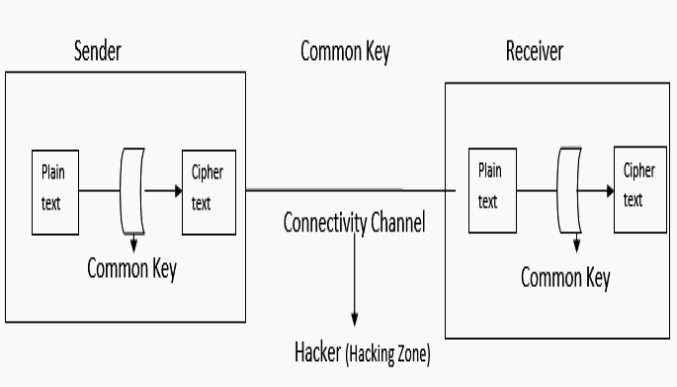
Abstract:

In this algorithm we use the concept of random function for encrypt and decrypt the image using blowfish Algorithm. At present the need of information security has become a necessity. Random number generator (RNG) is widely used in cryptographic system as the cryptographic keys generator. These keys are the most important component in thesystem since the security of the cryptographic system relies entirely on its quality. This algorithm will be used as a variable key size up to 448 bits Latest mitigation techniques proposed at register-transfer level for dependable cryptosystems deal with time redundancy in an active on-line error-detection scheme. Round-based block ciphers are very likely to be hardened with these techniques. With this approach, attackers must face two trouble:dealing alongside on-line error detection and flouting the obligation locale in the round sequence. PRNG will produce disparate repetition sequences for the rounds of the cryptosystem, making extremely tough to associate output data alongside inoculated faults.

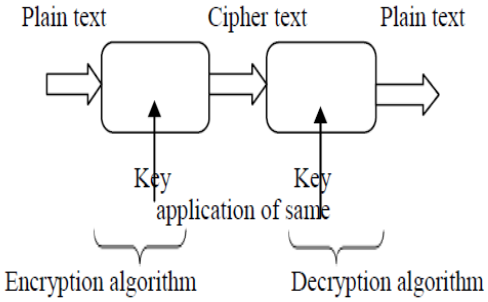
1. Introduction:

Cryptography plays a very vital role in keeping the message safe as the data is in transit. It ensures that the message being sent at one end remains confidential and should be received only by the intended receiver at the other end. Cryptography converts the original message in to non-readable format and sends the message over an insecure channel. The people who are unauthorized to read the message try to break the non-readable message but it is hard to do it so. The authorized person has the capability to convert the non-readable message to readable one. Cryptographic algorithm is classified into two categories: (I) Symmetric Key Cryptography where one key is used for both encryption and

decryption. (ii) Asymmetric Key Cryptography where two different keys are used one for encryption and other for decryption [7]. Symmetric key cryptography is divided into two types on the basis of their operations [8]: (I) Stream Cipher: A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. (ii) Block Cipher: A block cipher is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length [6]. Blowfish is a symmetric block cipher designed by Brute Schneider in 1993. Blowfish is a replacement of DES or IDEA [14]. Blowfish algorithm is a symmetric block cipher with a 64-bit block size and variable key length from 42 bits to 448 bits.



A network is a series of individual elements transmitting and receiving various data. Whenever sensitive or confidential information is transmitted, there is a possibility of an unauthorized third party ”eavesdropping” on a transmission and learning contents of the sensitive message. This possibility is unacceptable in many scenarios. Cryptography is the process of translating a message into a form which is unreadable to everyone except the intenderecipient. This is typically done with use of keys. A cryptographic key is roughly equivalent to the concept of a physical which can unlock the correct lock. In cryptography, keys are used to encrypt the message into a format which would appear as unreadable random information to an unauthorized third party.



2: Symmetric encryption/decryption process for blowfish algorithm

2. Goals of Cryptography:

Cryptography provides a number of security goals to ensure the privacy of data, non-alteration of data and so on. Due to the great security advantages of cryptography it is widely used today.

• Confidentiality Information in computer is transmitted and has to be accessed only by the authorized party.

• Authentication The information received by any system has to check the identity of the sender that whether the information is arriving from an authorized person or a false identity.

• Integrity Only the authorized party is allowed to modify the transmitted information. No one in between the sender and receiver are allowed to alter the given message.

• Non Repudiation Ensures neither the sender, nor the receiver of message can deny the transmission.

• Access Control only the authorized parties are able to access the given information.

Blowfish was designed to meet following design goals:

Speed:

It is meant to be significantly faster than DES on 32-bit microprocessors with relatively large caches. This type of architecture is readily available today to everyone.

Compactness:

It is designed to run in a relatively small memory space, less than 5K.

Simplicity:

Only simple operations are used, including addition, exclusive- or, and table lookups.

Flexibility of key size:

The size of key can vary up to 448 bits (in 32 bit increments).

3. Description of Algorithm

The Blowfish algorithm is conceptually simple, but is actual implementation and use is complex. Blowfish has fixed 64 bit block size. The key length of Blowfish is anywhere from 32 bits to 448 bits. The cipher is 16 round Feistel network which utilizes a structure that makes encryption and decryption very similar through use of following elements: P-boxes (permutation boxes these perform bit shuffling), S-boxes (substitution boxes – similar to non-linear function) and

Xoring to achieve Linear mixing [8].Blowfish is a Feistel network block cipher with a 64 bit block size and a variable key size up to 448 bits long. The Blowfish algorithm is unencumbered by patents and is free to use for any one.

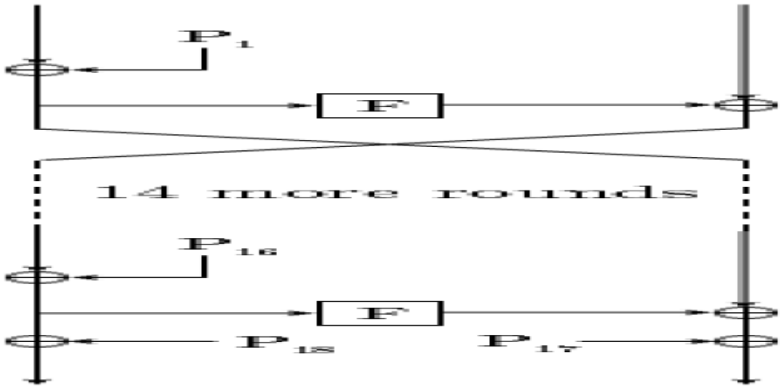


Fig. 3: Blowfish Algorithm

The F function is the feistel function of Blowfish, the contents of which are shown below.

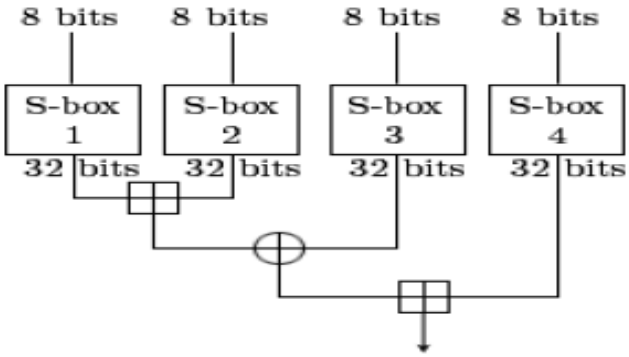


Fig. 4: The Feistel Function of Blowfish

3.1 Blowfish consists of three parts:

A. Encryption algorithm

B. Key-expansion

C. Decryption algorithm

A: Encryption Algorithm: During the key expansion stage, the input key is converted into several sub key arrays total 4168 bytes. There are the P-arrays, which has eighteen 32-bit boxes, and the S-boxes, which are four 32-bit arrays with 256 entries each. All of these boxes are initialized with a fixed string, the hexadecimal digits of pi [10]. After the string initialization, the first 32 bits of the key are XOR with P1 (the first 32-bit box in the P-array). The second 32 bits of the key are XOR with P2, and so on, until all 448, or fewer, key bits have been XOR. Cycle through the key bits is completed by returning to the beginning of the key, until the entire P-array has been XOR with the key [12].

B. Key-Expansion: In a simple cipher, one might exclusive-OR the key with the plaintext. Such a step is easily reversed by another exclusive-OR of the same key with the cipher text. In the case of the Blowfish, there are a number of rounds, needing the key, so the actual key size is 64 bytes [9] [12].

C. Decryption Algorithm: Decryption is the same as encryption, except the P-arrays are used in reverse. Hence, Blowfish encrypts by splitting half the block (32 bits) into 8-bit chunks (quarters) and inputting this into the S-box. The result from S-boxes then are added and XOR. Decryption is quite simple and accomplished by merely inverting the P17 and P18 cipher blocks and using P entries in reverse. The S-boxes and P-boxes are initialized with values from hex digits of pi. The variable length user-input key is then XOR with P-entries. Then a block of zeros is encrypted, and this result is used for P1 and P2 entries. The cipher text resulting from the encryption of a zero block is then encrypted again and use for P3 and P4. This process continues until every P-box entry and S-box entry has been replaced, resulting in 521 successive key generations. This involves about 4KB of data processing. This relatively complex key schedule makes Blowfish an effective and durable cryptographic algorithm [10].Blowfish is among the fastest block ciphers available and yet remains cryptographically secure.

4. Methodology: The main feature of the encryption/decryption program implementation is the generation of the encryption key. Other features are related to the design of the GUI, progress of encryption details, and user notification of the status of encryption.

4.1. Key Generation: A symmetric Encryption key is used for this application, which means the same key is shared for both Encryption and decryption. A copy of the generated key is saved in a file named .ekf during the Encryption process and the same key is used as the decryption key to retrieve the encrypted file. The technique of generating the key uses two methods: random number generation and combination. First, a long number with only digit values called A is generated. Then another long number with character values called B is generated. The size of B is twice the size of A. Then an insertion operation is performed such that each digit of A is inserted after two characters of B. The result of the insertion is called C. Then another only digit number called D is randomly generated. C is combined with D by placing alternately one character or digit from C after a character or a digit from D. The result of the combination is a relatively strong key. Then, an odd and even partitioning is performed on the key. The position of each character in the key

decides. It to be an even or an odd character. For example, the character at position 0 is an even one while the character at position 1 is an odd character. Similarly, position 2 is an even position while position 3 is odd one. The even part of the key is combined together and the odd part of the key is combined together. Finally, the two parts of the key are joined as an even part followed by an odd part to produce one final encryption key. Since the final key is a key that consists of all characters, another key with only ASCII values of each character is obtained. The result is a very long decimal key. Figure 1 shows the complete key generation process.

4.2. Encryption Rules: For the Encryption method, a single digit in the decimal ASCII representation will decide which Encryption method is to be applied to the single a binary block in a file. I. 0 in the key means a rotation of bit to the left is performed and the next integer to 0 in the ASCII code decides how many bits the block will be moved to the left. II. 1 in the key means a rotation of bit to the right is performed and the next integer to 1 in the ASCII code decides how many bits the block will be moved to the right. III. 2 means the block will be passed to an XOR Encryption to be performed with a binary block from the file. IV. Else, all other numbers in the key like 3,4,5,6,7,8,9 are ignored.

5. File Types:

There are no limitations of the type of files accepted for encryption in this application, which means any type of a file such as data files, audio files, video files or image files can be encrypted by the application. This is because all the files are encrypted at the binary level. There is also no limitation of the size of the file that can be encrypted using this application, which provides flexibility to the user. The encrypted file can only be opened and viewed after it has been decrypted to its original file using the symmetric encryption key.

6. Conclusion & Future Work:

A new simple tool has been created, which is targeted for use inside of a small institution such as a small university for lecturers' daily use of sending exam files and sensitive material such that the material can be encrypted and the file is sent in one e-mail while the encryption key is sent in another e-mail or via any secure communication channel. The encryption application developed and described in this paper might not be comparable to well-known encryption algorithms but its File Types File Size(Mb) Encryption Time (S) Success Rate Document 1/ 3/ 5 9/27/45 100% Image 1/ 3/ 5 10/26/44 100% Audio/ Video 1/ 3/ 5 18/28/45 100% Zipped 1/ 3/ 5 10/25/44 100% Exe. 1/ 3/ 5 12/27/48 100%

Simplicity and availability proves that tools can be developed that fit the needs of an institution without resorting to purchasing expensive software from the market. For future enhancement to this application public key encryption can be applied where two keys can be generated: one to encrypt a file using the public key and another private key to decrypt it. Also, other more advanced encryption operations can be included to enhance the security of the application so that it can be used to encrypt more sensitive administrative material in an institution.

References:

1. Wikipedia,"Encryption", http://en.wikipedia.org/wiki/Encryption, modified on 13 December 2006.
2. Freeman J., Neely R., and Megalo L. "Developing Secure Systems: Issues and Solutions". IEEE Journal of Computer and Communication, Vol. 89, PP. 36-45. 1998
3. Agnew G. B., Mullin R. C., Onyszchuk I. M., and Vqanstone S. A. "An Implementation for a Fast Public-Key Cryptosystems". Journal of Cryptology, Vol.3, No 2, PP. 63-79. 1995.
4. Beth T. and Gollmann D. "Algorithm Engineering for Public Key Algorithms". IEEE Journal on Selected Areas in Communications; Vol. 7, No 4, PP. 458-466. 1989.
5. IBM. "The Data Encryption Standard (DES) and its strength against attacks". IBM Journal of Research and Development, Vol. 38, PP. 243-250. 1994.
6. Wikipedia, "Bitwise operation", http://en.wikipedia.org/wiki/Bitwise_operation, last modified on 10 December 2006.
7. Andy Wilson, "Tips and Tricks: XOR Encryption" http://www.andyw.com/director/xor.asp, 1998.
8. Baraka H., El-Manawy H. A., and Attiya A. "An Integrated Model for Internet Security Using Prevention and Detection Techniques". IEEE Journal of Computer and Communication Vol. 99, PP. 25-33. 1998.
9. Microsoft, "Encrypting File System for Windows 2000", http://www.microsoft.com/windows2000/techinfo/howitworks/security/encrypt.asp, 1998.

Corresponding Author:
Anjaneyulu GSGN*[1],
Email: *anjaneyulu.gsgn@vit.ac.in*