Research Journal of Applied Sciences, Engineering And Technology 4(6): 603-607, 2012
ISSN: 2040-7467

Submitted: September 29, 2011 Accepted: November 15, 2011 Published: March 15, 2012

A New Approach for Improving Data Security using Iterative Blowfish Algorithm

[1]G. Manikandan, [1]N. Sairam and [2]M. Kamarasan
[1]School of Computing, SASTRA University, Thanjavur-613401, Tamil Nadu, India
[2]CSE Department, Annamalai University, Chidambaram-608002, Tamil Nadu, India

Abstract: In Recent times, we are facing various challenges in security related issues during data transfers. There are various security models following different enciphering techniques for the betterment of secured data transfer. Though there exist many complex cryptographic encryption algorithms, which provide high level of security, vulnerability of those algorithms increases day after a day. It is to be worth saying a point that modification of existing complex algorithms will obviously intensifies to the enhancement in security of algorithm as well as the data ,provided the modification should not be eavesdropped easily than the original algorithm. In this paper, we proposed a software tool which considerably enhances the security by following an iterative approach depending upon sender's need. Our experiments show that the use of iterative approach enhances the security provided by the algorithm when compared to the non-iterative approach.

Key words: Block cipher, blowfish, cryptography, iteration, security

INTRODUCTION

For ensuring the security, the plain text is converted to cipher text and the process is called encryption (Stallings, 1999). Although this conversion idea is old, the way of encryption should not be vulnerable to attacks. Caesar's cipher method, poly alphabetic substitution method, bit-level encryptions like substitution box; permutation box, encoding, and rotation are some of the conventional encryption methods. These methods are easy to implement but can be cracked easily with the high end technologies. The objective of this project is to develop iterative-level encrypter software that can be used to encrypt top-secret files including text, images and multimedia files in the secondary storage devices.

LITERATURE REVIEW

Most of the existing systems are vulnerable to attacks and it is breaked at some point of time by crypt analysing it. There are various cryptanalysis techniques available to break most of the encryption algorithms at one point of time. Each and every algorithm either it may be block cipher or stream cipher or any other cipher types can be easily attacked by performing various cryptanalysis techniques like linear cryptanalysis, n-gram analysis, meet in the middle attack, brute force attack, Man in the middle attack etc... It's pity to say that intruders can intrude any systems even it has a complex algorithmic design. Most of the famous algorithms of all ages are breaked easily by eavesdroppers at one stage and we are evidencing it in our day-to-day daily life. This happens because of its platform dependency and the emerging trend of open software solutions available all over the world. Despite some systems are developed to support cross platform, they do not use multi level encryption. This is because the algorithmic developers always believe in their own encryption formulas and firmly attached to the tradition of modifying or using or creating a single algorithm which is not secure after a period of time. It is quite obvious to digest the fact it is easy to cryptanalysis any algorithm within months as soon as they are adapted to practical use. Most of the existing systems support text encryption preferably than other media types. Since the intruders and eavesdroppers had shown their excelling skills towards breaking the encryption algorithms almost in all important and sensible areas like Banking, Military, Defence, Networks, a need for "practically strong and infeasible to get attacked" algorithm becomes vital. This paper suggests one such cryptographic technique which never ever gives a clue of neither the encryption pattern adopted nor the number of iterations that will carry out to obtain the high end cipher text.

Cryptography is a well known and widely used technique that manipulate information in order to crypt their existence. To be more specific, cryptography protects information by transforming it into an unreadable format (Stallings, 1999). The original text is transformed into a

Corresponding Author: G.Manikandan, Assistant Professor, School of Computing, SASTRA University, Thanjavur-613401, Tamil Nadu, India

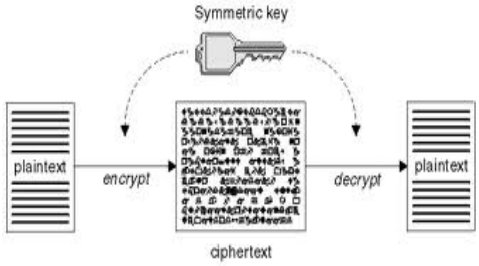

Fig. 1: Symmetric key cryptography

scramble equivalent text called cipher text and this process is called as "Encryption". This is achieved via an Encryption Algorithm. Only those who possess a secret key can decrypt the cipher text into plaintext. Simply it scrambles a message so it cannot be understood.

Cryptography deals with protecting information by encoding or transformation of data (Stallings, 1999).There are two types of cryptographic schemes available on the basis of key (Stallings, 1999):

- **Symmetric key cryptography:** This is the cryptographic scheme which uses a common key for enciphering and deciphering the message.
- **Asymmetric or public key cryptography:** This type of cryptographic scheme uses two keys for encryption and decryption called Public key and Private Keys.

We adopted Symmetric key cryptographic scheme as shown in Fig. 1 and hence only one key is needed for communication. So, the chosen cryptographic scheme involves:

- **Plaintext:** The original message that has to be communicated to receiver.
- **Encryption:** Enciphering of data by using a key via a desired encryption algorithm at sender side.
- **Transmission**: Transfer of cipher message to receiver through a public communication channel.
- **Decryption:** Deciphering of the ciphertext thus received via the same algorithm (reverse Encryption) by using the key.

We can also classify symmetric key cryptography into two types on the basis of their operations as:

- **Stream ciphers:** It is a symmetric key cipher where stream of plaintext are mixed with a random cipher bit stream (key stream), typically by any logical operation (say exclusive-or (xor) operation). In a stream cipher the plaintext digits are encrypted one at a time
- **Block ciphers:** It is also a symmetric key cipher operating on fixed-length groups of bits, called blocks. A block cipher encryption algorithm takes an n-bit block of plaintext as input, and produces a corresponding n-bit output block of ciphertext.

We have chosen block cipher for our cryptographic operation since it is the main tool for implementing private key encryption in practice.

There are a number of ways to enhance the security of different cryptographic and steganographic schemes. One of the best ways to strengthen a cryptographic scheme is to increase the strength of its key. Manikandan *et al.* (2011) suggest a method to improve the strength of the key in case of RC4 algorithm.

Sairam *et al*. (2011) proposed an approach to improve the security by using multilevel cryptographic schemes.

Another interesting approach is to combine the two basic forms of ciphers, namely block cipher and stream cipher (Manikandan *et al*., 2011). This approach was studied further and instead of using both the type of ciphers on the plain text (Manikandan *et al*., 2011) suggested a method in which one was used with the plain text and the other with the key.

Vaithiyanathan *et al*. (2010), proposed an approach to decrease the execution time of blowfish algorithm by using a modified F-function.

PROPOSED SYSTEM

We proposed a system which is different and efficient from the existing systems as follows:

- Our System is developed in such a way that it is platform independent. Where the existing systems are limited to platform dependent design.
- It has an encryption algorithm which runs iteratively based upon the number of iterations whereas the existing systems are always focussed as encryption at single level.
- We use a numeric phrase obtained upon user's specification which determines the number of iterations that the algorithms need to be executed.
- Moreover the number of iterations that we use will always remain a secret and hence it don't even leave a single chance for the eavesdroppers to make a guess on our system and hence the security offered is up to the best of ever provided.
- This proposed system is developed in order to support not only text files but also images and media files. But still many of the existing systems are developed in order to suit basic text formats.

The algorithmic design of iterative cryptographic tool can be well understood by the following block diagram which encryption and decryption of a cryptographic algorithm at iterative level.

We use a cryptographic algorithm named blowfish algorithm for the explanation of our iterative

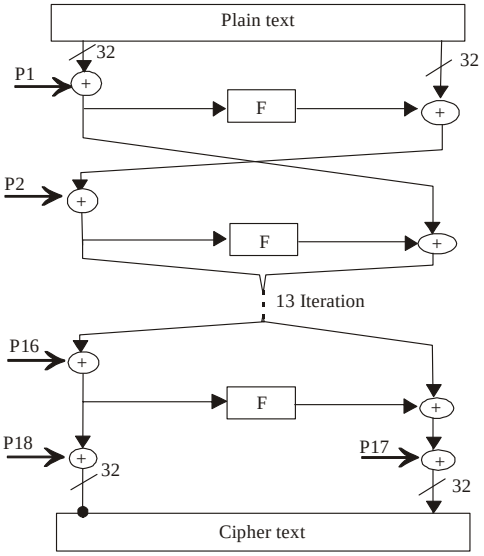

Fig. 2: Fiestal structure of blowfish cipher

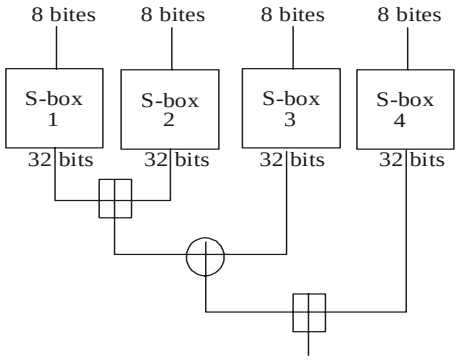

Fig. 3: Structure of F-function

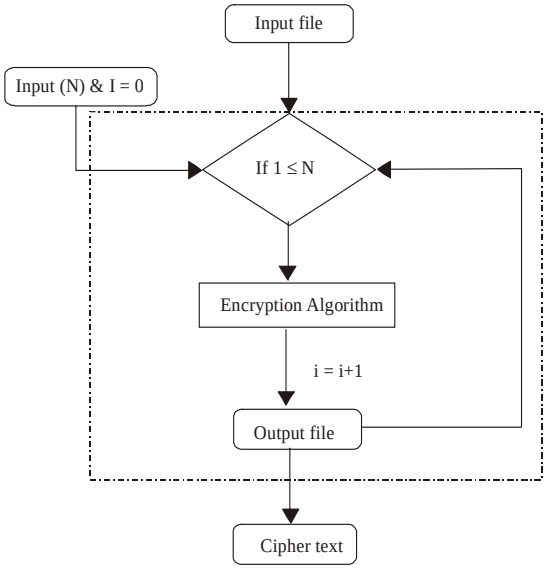

Fig. 4: Iterative encryption

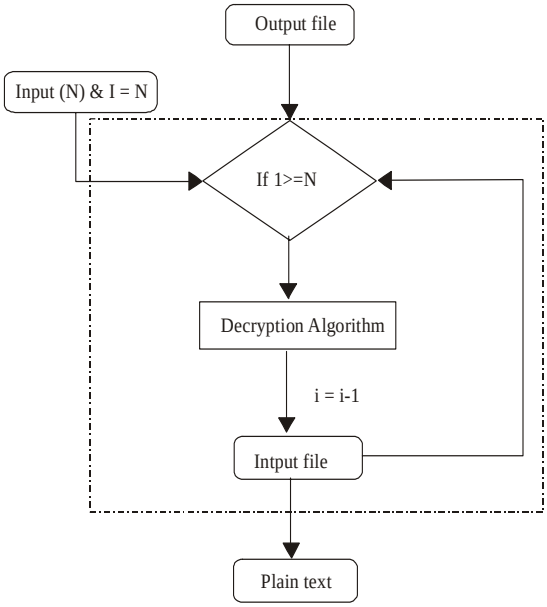

Fig. 5: Iterative decryption

cryptographic approach. So let us review the algorithm and its style of encryption in brief in the forth coming paragraphs.

BLOWFISH ALGORITHM

Blowfish, a symmetric block cipher that uses a modified Feistel network structure, which has 16 rounds for encryption and decryption. The block size is 64 bits, and the key size is up to 448 bits. The strength of the Blowfish algorithm relies on its sub-key generation and its encryption. Blowfish is a block cipher which uses a variable-length key. It is well fitted for applications in which the key size does not change often. It is significantly faster than most encryption algorithms when implemented on 32-bit microprocessors with large data caches. (Schneier, 1994).

Blowfish cipher uses 18 each of 32-bit sun arrays commonly known as P-boxes and four Substitution boxes each of 32 bit size and having 256 entries each. It uses a Feistel cipher which is a general method of transforming a function into another function by using the concept of permutation. The working of blowfish cipher can be illustrated as follows.

It splits the 64 bit block into two equal blocks having 32 bit size each. Left block is XORed with first sub array P1 and thus obtained result is fed in to a function called F-function. Inside the F-function substitution operations are carried out which in turn converts 32 bit blocks in to an another 32 bit blocks. Thus resulted 32bit entries are XORed with the Right half and the result obtained is swapped as the left half for the next round. So, After the successful completion of each round Right half becomes

Table 1: Plain text and cipher texts in iterative encryption process

NO	IV Iteration	Iteration III	II Iteration	I Iteration	Plain text
1	¹ÿîjŠüvï´ß<Ì±”	dŠ!¼ ú+ù”Q	ˆ×G`ƒtNÄ¸	±†;3K,†	Morning
2	?ìV4ú¸?:4Pîÿ.Õti	eé<Xf›‘r 8V7&`sd	šmw½.$B!`Úí¢	Å,ê·¨»"Üi?bP[	Afternoon
3	¸¾ëxC¡¯tiOÉó	Q¤Ð«GãXÍ_°®š	ie-ç%NÄ	5"=[Ö‰	Evening

Table 2: Plain text and cipher texts in iterative decryption process

NO	IV Iteration	Iteration III	II Iteration	I Iteration	Plain text
1	¹ÿîjŠüvï´ß<Ì	±”¸dŠ!¼ ú+ù”Qˆ	×G`ƒtNÄ±	†;3K,†	Morning
2	ìV4ú¸ :4Pîÿ.Õti	eé<Xf›‘r 8V7&`sd	šmw½.$B!`Úí¢NÄ	Å,ê·¨»"Üi ?Bp[	Afternoon
3	¸¾ëxC¡¯ ti	OéóQ¤Ð«GãXÍ_°®š	ie-ç%NÄ	5"=[Ö‰	Evening

the new left half or vice versa and Fiestal structure is followed up to 16 rounds. The resultant left and right halves are not swapped but XORed with the seventeenth and eighteenth P-arrays (Vaudenay, 1996). The Fiestal Structure of blowfish algorithm is shown in the Fig. 2.

The transformation operations that actually happen inside an F-function are XOR Operation, ADD Operation and few table look up operations. These operations are carried out between four S-Boxes and as a result of all manipulations finally 32 bit entries are transformed into another 32 bit entry. F-Function of a Blowfish algorithm can be depicted pictorially in Fig. 3 (Kishnamurthy *et al*., 2006).

WORKING OF ITERATIVE MODEL

The encryption and the decryption process using our iterative approach has been shown in Fig. 4 and 5, respectively.

The steps which are involved in the completion of iterative encryption process are as follows:

1 At first, Input file which will be the plaintext is obtained from the user along with a number N
2 Initialize integer i = 0
3 For the condition, i ≤ N, execute steps 4 and 5 else GOTO 6
4 Encryption takes place and the input file is encrypted using Encryption algorithm
5 Increment i and GOTO 3
6 The final resultant text will be considered as the high end cipher text and it will be transported to receiver through any communication medium.

The steps which are involved in the completion of iterative decryption process are as follows:

1 At first, Input file which will be the ciphertext is obtained from the sender along with a number N
2 Initialize integer i = N
3 For the condition, i > = N, execute steps 4 and 5 else GOTO 6
4 Decryption takes place and the input file is decrypted using Decryption algorithm
5 Decrement i and GOTO 3
6 The final resultant text will be actual text to be shared between sender and receiver.

Table 1 and 2 shows the plain text and the corresponding cipher text obtained through iterative approach. The encryption and decryption are coded, tested and verified successfully using Java and the results are summarized for the various standard plain text inputs.

ADVANTAGES OF ITERATIVE APPROACH

- It is n times more secure than the standard Encryption algorithm where n is the number of iterations.
- Even intruders decrypts it successfully the cipher text. which results will be in scrambled form and hence they will end up with none other than confusion and frustration.
- The number of rounds of encryption and decryption is in the hands of user and hence one cannot predict the number very easily.
- There won't be any increase or decrease in the size of text and hence it does not give chance for any suspicion.
- The architecture is standard and simple and hence it suits any encryption algorithm.
- The Security level can be increased bit more by encrypting each and every level with different key provided it should be given back correctly in reverse order for the decryption process.

FUTURE ENHANCEMENTS

This system can be enhanced by developing a standard formula for generating the number N which determines the number of iterations that is to be carried out. Though the system is designed for storage level but the modules can be used in web services also. By adding a new button with a server and client sockets, the system can also be improved to work as secure LAN File messenger. Security can also be enhanced by following encryption, decryption, encryption in the enciphering phase and similarly decryption, encryption, decryption while deciphering it.

CONCLUSION

The iterative model proposed in this paper has a major advantage over the regular blowfish algorithm. The overall security of the algorithm is enhanced drastically. From our experiments we concluded that the algorithm gives us different cipher text for the same plain text, in a given iteration. For our experimental purpose we have used only one key throughout the process. The strength of this algorithm can be increased by using a different key in each iteration.

REFERENCES

Kishnamurthy, G.N., V. Ramaswamy and G.H.. Leela, 2006. Performance Enhancement of Blowfish algorithm by modifying its function. Proceedings of International Conference on Computers, Information, System Sciences and Engineering, University of Bridgeport, Bridgeport, CT, USA, pp: 240-244.

Manikandan, G., R. Manikandan and G. SundarGanesh, 2011a. A New Approach for generating strong key in RC4 algorithm. J. Theor. Appl. Inf. Technol., 24(2): 113-119.

Manikandan, G., G. Krishnan and N. Dr. Sairam, 2011b. A unified block and stream cipher based file encryption. J. Global Res. Comp. Sci., 2(7): 53-57.

Manikandan, G., R. Manikandan, P. Rajendiran, G. Krishnan and G. SundarGanesh, 2011c. An integrated block and stream cipher approach for key enhancement. J. Theor. Appl. Inf. Technol., 28(2): 83-87.

Sairam, N., G. Manikandan and G. Krishnan, 2011. A novel approach for data security enhancement using multi level encryption scheme. Inter. J. Comp. Sci. Inf. Technol., 2(1): 469-473.

Schneier, B., 1994, Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993), Springer-Verlag, pp: 191-204.

Stallings, W., 1999. Cryptography and Network Security: Principles and Practices. 2nd Edn., Prentice Hall, Country.

Vaithiyanathan, V., G. Manikandan and G. Krishnan, 2010. A novel approach to the performance and security enhancement using blowfish algorithm. Inter. J. Adv. Res. Comp. Sci., 1(4): 451-454.

Vaudenay, S., 1996. On the Weak Keys in Blowfish Fast Software Encryption, Third International Workshop Proceedings, Springer-Verlag, pp: 27-32.