Volume 2, No. 7, July 2011

Journal of Global Research in Computer Science

ISSN-2229-371X

RESEARCH PAPER

Available Online at www.jgrcs.info

A UNIFIED BLOCK AND STREAM CIPHER BASED FILE ENCRYPTION

Manikandan. G[*1], Krishnan.G[2] and Dr.N.Sairam[3]

[*1,2]Department of Information and Communication Technology, SASTRA University-613401, Tamil Nadu, India
manikandan@it.sastra.edu
krishnangk2020@gmail.com,
[3]Department of Computer Science, SASTRA University-613401, Tamil Nadu, India
sairam@cse.sastra.edu

Abstract: On considering the current scenario, Most of the existing systems which offer security to a network or web or to a data are vulnerable to attacks and they are breached at some point of time by effective cryptanalysis, irrespective of its complex algorithmic design. In general, today's crypto world is restricted to a practice of following any one single encryption scheme and that too for a single iteration on a single file basis. This is evident in the 99% of the encryption-decryption cases. So, A need for "practically strong and infeasible to get attacked" technique becomes vital. In this paper, we propose a Software tool which involves Cryptographic enciphering and deciphering using two algorithms of different dimensions and also they are well assisted with File Splitting and Merging mechanisms. We used modified Blowfish algorithm and RC4 algorithm for Encryption and Decryption of data. Our results clearly justifies that our tool serves as a better solution both in terms of performance as well as security.

Keywords: Cryptography, File Slicing, Modified Blowfish Algorithm, RC4, Security

INTRODUCTION

Cryptography is the study and practice of protecting information by data encoding and transformation techniques [1].There are two types of cryptographic schemes available on the basis of key:

1. ***Symmetric key Cryptography***: This is the cryptographic scheme which uses a common key for enciphering and deciphering the message.
2. ***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 and hence only one key is needed for communication. So, the chosen cryptographic scheme involves:

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

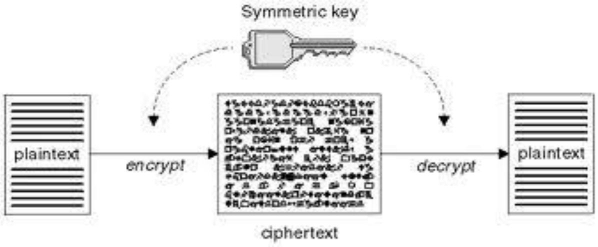


Figure 1: Symmetric Key Cryptography

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

1. *Stream Ciphers*: It is a classification under symmetric key Encryption ciphers family where plaintext is combined with a random stream of keys which resulted from a pseudorandom sequence. In a stream cipher the plaintext digits are encrypted one at a time
2. *Block Ciphers*: It is also a symmetric key cipher which works on fixed-sized bits usually referred as blocks. A block cipher encryption algorithm takes a n-bit block of plaintext as input, and produces a corresponding n-bit output block of cipher text.

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

LITERATURE REVIEW

Blowfish Algorithm

Blowfish is a symmetric block which consists of 16 rounds of iterative functional design. The block size of blowfish algorithm is 64 bits, and the size of the key may be of any length but having a maximum range till 448 bits. The power of the Blowfish algorithm relies on its sub-key generation and its complex encryption. Blowfish cipher uses 18 P-boxes and 4 Substitution boxes each of 32 bit size. Blowfish follows 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 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. The Fiestal Structure of blowfish algorithm is shown in the Fig-2

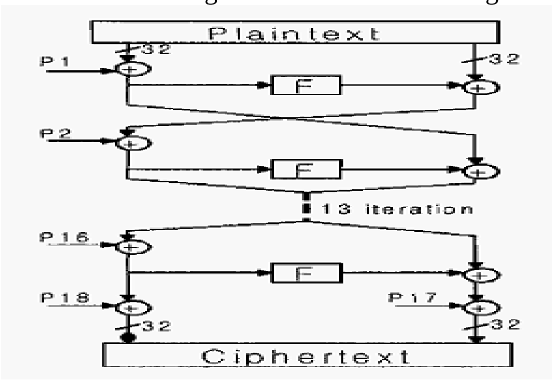


Figure.2: Fiestal Structure of Blowfish Algorithm

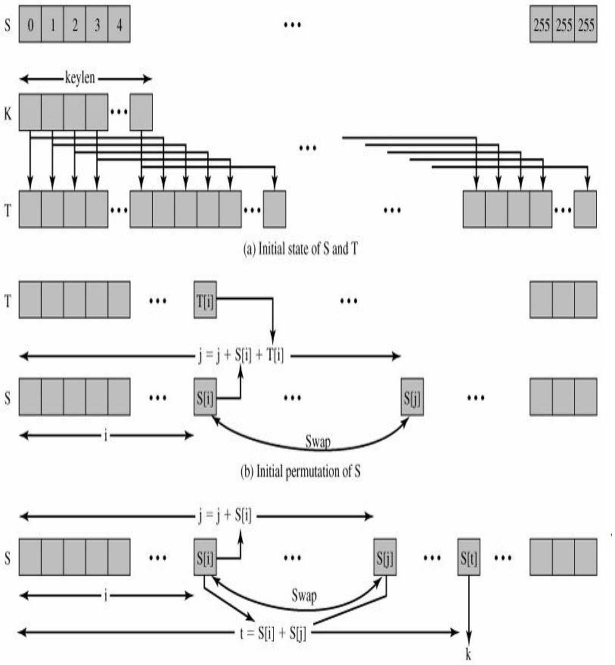
Moreover the f-function of blowfish is slightly modified for the enhancement of security and performance.

RC4 Algorithm

RC4 is most widely used stream cipher nowadays due to its simplicity and high efficiency .RC4 is a variable key size stream cipher based on a 256 byte internal state and two one byte indexes I and J. RC4 consist of two parts namely key scheduling algorithm and pseudo random generation [9].

The key-scheduling algorithm consists of following steps in order to generate a key we should start the permutation in the array "S". Key length is usually of the range 1 ≤ key length ≤ 256. Initially, the array "S" is initialized and 256 iterations are carried out and the operational procedure is same as the PRGA [10].

In the phase of Pseudo random generation algorithm (PRGA), for each iteration, the PRGA increments *i*, adds the value of S pointed to by *i* to *j*, swaps the values of S[*i*] and S[*j*], and thus each entries of S array is swapped with the another entry of the array. This should be taken place at least once for every 256 iterations.[10]



(a) Initial state of S and T

(b) Initial permutation of S

Figure 3:RC4 Algorithmic Design

PROPOSED SYSTEM

This Software tool involves Cryptographic enciphering and deciphering along with File Splitting and Merging mechanisms. In this approach a file which has secret data is sliced into two halves and then the cryptographic encryption phase is carried out. In order to achieve more security we can adopt more than one cryptographic scheme which definitely ensures nil suspicion and more security. In this paper, we differentiate the cryptographic scheme by providing two different algorithms namely a block cipher - Blowfish algorithm and a stream cipher –RC4 algorithm; provided the key should be given correctly at the time of decryption to avoid erroneous results. In particular, the usage of modified Blowfish algorithm for Encryption and Decryption of data will serves as a better solution both in terms of performance and as well as security. This enhancement in security and performance is sustainably justified in our previous work [2].

In the file joining phase, En-Ciphered files thus obtained from the different En-Ciphering techniques are merged and hence transmitted to reception side as a single file which makes the file infeasible to breach and suspicion less to get to know that varying crypto schemes are adopted. And hence the data security is maximized. At the receiver's end, We once again splits the files and decrypts it using the different algorithms and then joins all split files together to retrieve the original message.

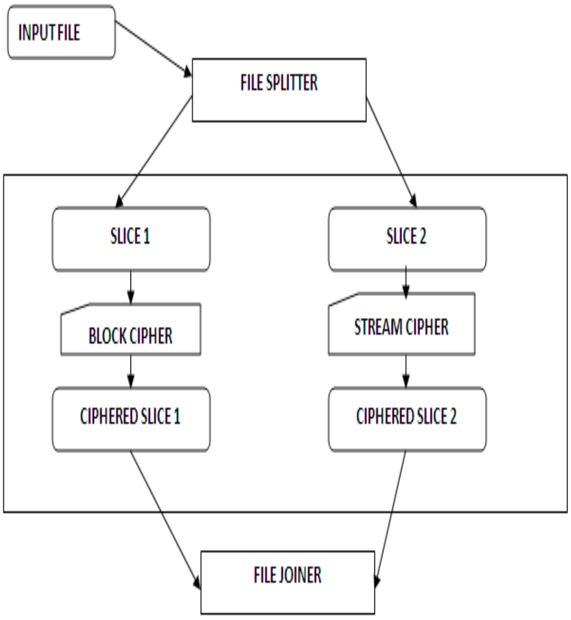


Figure 4: Proposed System Block Diagram

Modified F-Function

Function F plays an important role in the algorithm, and we decided to modify function F. Original function F is defined as follows. [3]

$F(X) = ((S_1 + S_2 \bmod 2^{32}) \text{ XOR } S_3) + S_4 \bmod 2^{32}$

Instead, we modified the F-Function by replacing 2 addition operations as XOR Operations. Thus the modified F-Function is written as,

$F(X) = ((S1 \text{ XOR } S2 \bmod 2^{32}) + (S3 \text{ XOR } S4 \bmod 2^{32}))$

This modification leads to the simultaneous execution of two XOR operations. In the case of original F-function which executes in sequential order and it requires 32 Addition operations and 16 XOR operations. But in the case of our modified F-function it requires the same 48 gate operations (32-XOR, 16-addition) but time taken to execute these 48 operations will be reduced because of multithreading [2]. We executed 32 XOR operations in parallel order using threads and hence time taken to complete 16 gate operations will be equal to the time taken to complete 32 XOR operations since we are running it in parallel environment [4]

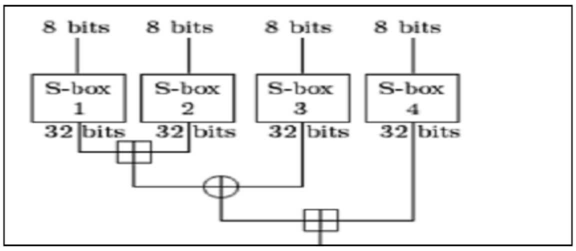


Fig 5: Existing F-Function

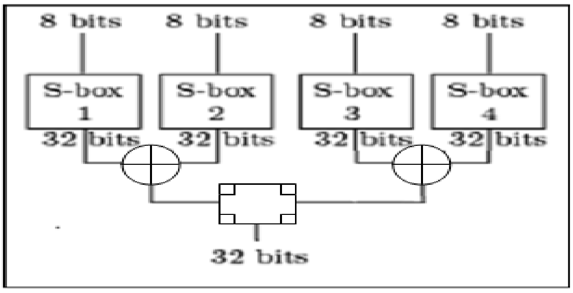
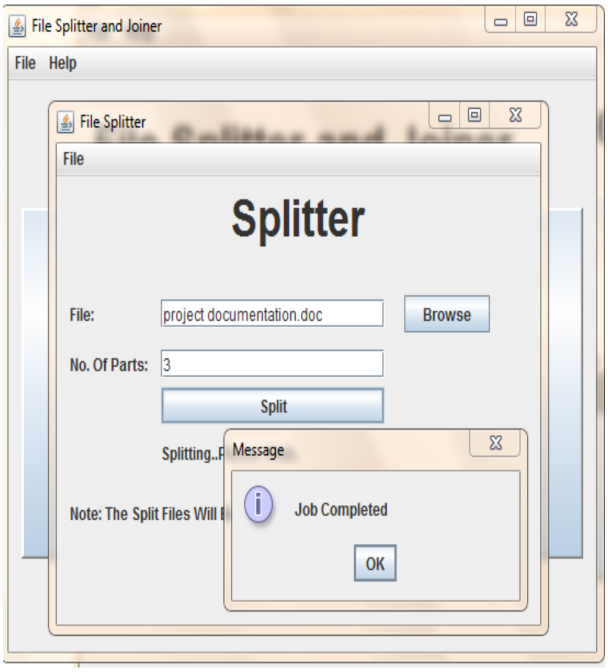


Figure 6: Modified F-Function

SIMULATION & RESULTS

For the purpose of simulating the File splitting and file merging and for modified blowfish algorithm we used Java which is well known for its platform independency and better GUI features. We developed, tested and executed using JDK 1.6 in core 2 duo processor. We adopted JCreator1.6 for IDE purposes.



File: project documentation.doc

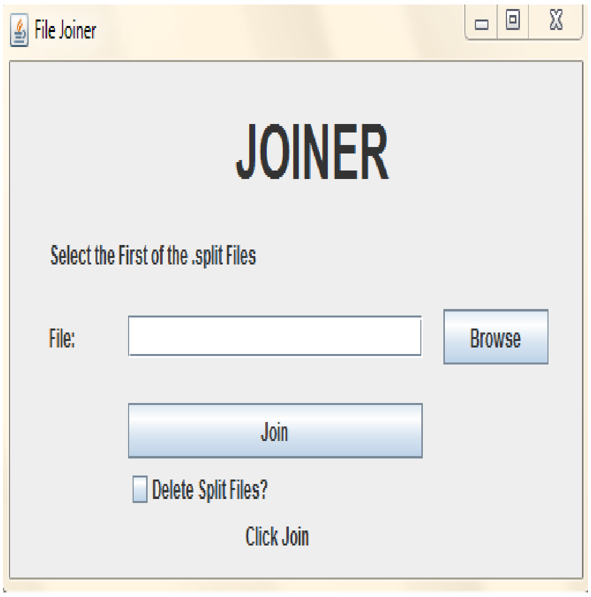
No. Of Parts: 3

Splitting..F

Note: The Split Files Will I

Job Completed

Figure 7: File Splitter GUI

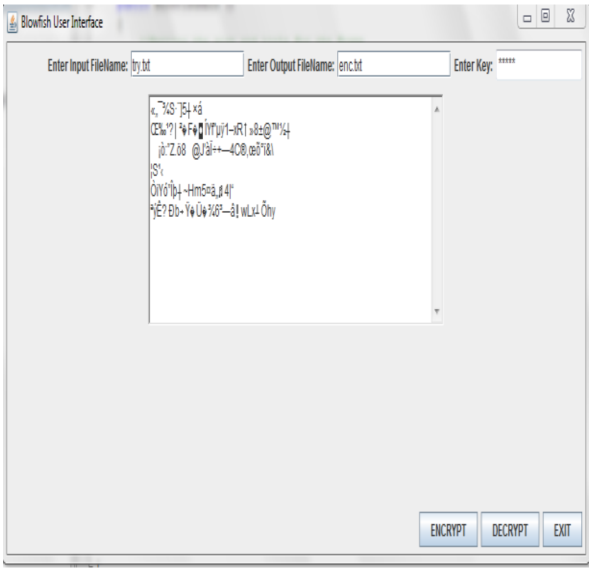


Select the First of the .split Files

File:

□ Delete Split Files?

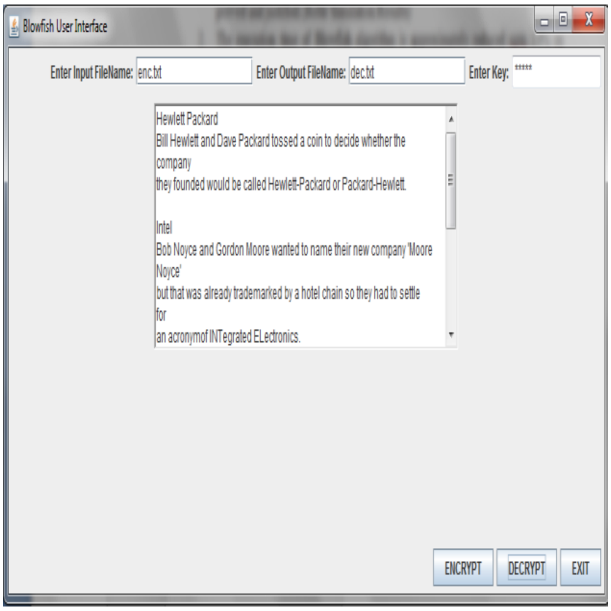
Figure 8: File Joiner GUI



Enter Input FileName: try.txt Enter Output FileName: enc.txt Enter Key: *****

```
«„¯¾S·]5┤×á
Œ‰¹?|²♦F♦▮ÍYf'µÿ1–xR↑»8±@™½┤
  ¡ò:'Z.ö8  @J'àÏ÷+—4C®,œõ˜ï&\
¦S'‹
ÒiYó'Îþ┤~Hm5¤ä„ɏ4|"
*ÿÊ? Ðb÷Ÿ♦Ü♦¾6³—â! wLx┴Õhy
```

Figure 9: Blowfish Encryption

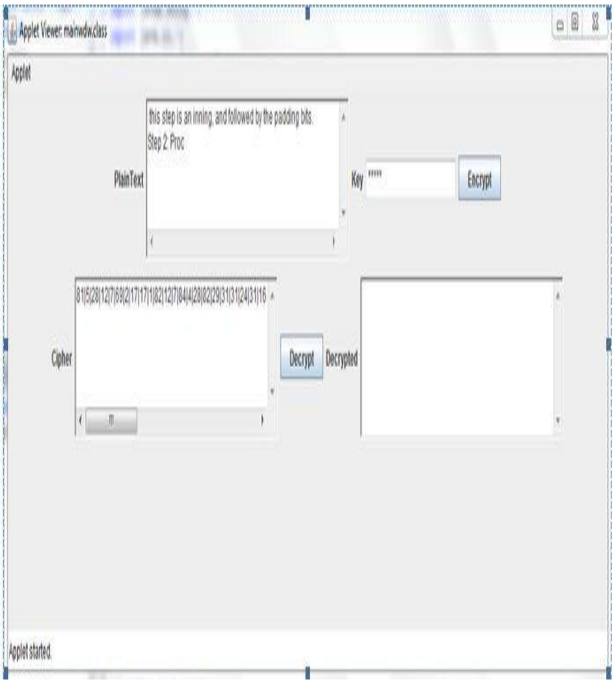


Enter Input FileName: enc.txt Enter Output FileName: dec.txt Enter Key: *****

```
Hewlett Packard
Bill Hewlett and Dave Packard tossed a coin to decide whether the
company
they founded would be called Hewlett-Packard or Packard-Hewlett.

Intel
Bob Noyce and Gordon Moore wanted to name their new company 'Moore
Noyce'
but that was already trademarked by a hotel chain so they had to settle
for
an acronymof INTegrated ELectronics.
```

Figure 10: Blowfish Decryption

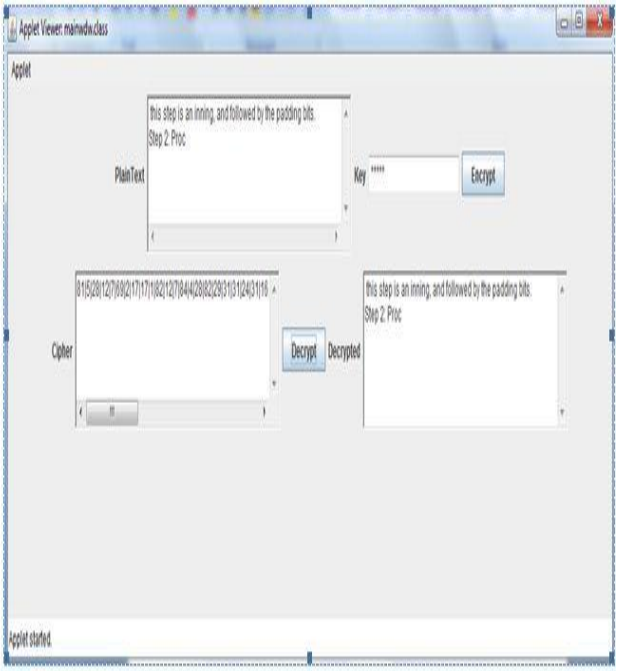


PlainText: this step is an inning, and followed by the padding bits.
Step 2: Proc

Key: *****

Cipher: 81|5|28|12|7|69|2|17|17|1|82|12|7|84|4|28|82|29|31|31|24|31|16

Figure 11: RC4 Encryption



PlainText: this step is an inning, and followed by the padding bits.
Step 2: Proc

Key: *****

Cipher: 81|5|28|12|7|69|2|17|17|1|82|12|7|84|4|28|82|29|31|31|24|31|16

Decrypted: this step is an inning, and followed by the padding bits.
Step 2: Proc

Figure 12: RC4 Decryption

Table 1: Comparison of Execution time

Time Vs Algorithm	**Start Time (ms)**	**End Time (ms)**	**Elapsed Time (ms)**
Original Blowfish Algorithm	1289281669804	1289281670225	499
Modified blowfish algorithm	1289282873275	1289282873706	431

Thus it is experimentally proved that the execution time of modified blowfish algorithm is 13.5% lesser than the original algorithm.

SIGNIFICANT FEATURES

This proposal has several merits to be appreciated.
First, on considering the File splitting and merging modules,

- Its simpler design & easy in implementing it.
- The Design and working is fashioned in such a way that, it is infeasible to breach.
- It also leaves no suspicion about Splitting & Merging of files.
- Thus our ultimate aim of providing a tool kit which offers “strong and infeasible to get attacked” is achieved.
- This software tool is modular, so any encryption algorithm can fit well in the place of our modified Blowfish algorithm.

Next, On considering the Encryption & Decryption Phase,

- The execution time of Blowfish algorithm is approximately reduced up to 13.5% on comparing with the original Blowfish Algorithm.
- Although, we used 2-XOR gates and 1-ADDER but the original F-function uses 2-ADDERs and 1-XOR gate and there is no abrupt change in the execution time or clock cycles required for execution. This is because all fundamental logical operations like AND,OR,XOR takes more or less equal time when running under any programming languages since those languages are logically driven.
- It’s quite hard for the eavesdroppers to realize that the F-function is modified and hence probability of attack is less on comparing with the original Blowfish algorithm.
- Since our proposed system bring modifications only to the order of execution and no changes is made to the actual functionalities (i.e., we didn’t added or removed new operations rather we changed only the order of execution of existing Xor and Adders) so performing cryptanalysis is not necessary.
- Block and stream ciphers have an entirely different cryptanalytic procedures. So there always remains an inconsistency in information due to their nature of cipher texts.

CONCLUSION

This paper will satisfy our foremost aim of providing a system which is “infeasible to get breached”. It also provides a high end data security when transmitting over any insecure medium. Intruders will not have any idea about our modification both in terms of algorithm as well as in our design, so breaching this system is highly impossible. We are sure that this software tool is unique of its kind and it can also be tuned in terms of higher performance and security in mere future by adding or replacing cryptographic part because of its modularity in design. That is, it has a good performance without compromising the security and the modified F-function also enhances the performance by reducing the clock cycles up to 33% and reduces the execution time up to 14% [2].

REFERENCES

[1] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed., John Wiley & Sons, 1995.

[2] Manikandan.G, Krishnan.G, “A Novel Approach to the Performance and Security Enhancement Using Blowfish Algorithm”, International journal of Advanced Research in Computer Science, 2011.

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

[4] William Stallings, Cryptography and Network Security, 3rd Ed, Wiley, 1995.

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

[6] Dr.V.Ramaswamy, Kishnamurthy.G.N, Mrs. Leela.G.H, Ashalatha M.E, “Performance enhancement of CAST –128 Algorithm by modifying its function” Proceedings of International Conference on Computers, Information, System Sciences and Engineering 2007, University of Bridgeport, Bridgeport, CT, USA.

[7] L. Knudsen, "Block Ciphers: A Survey", State of the Art in Applied Cryptography: Course on Computer Security and Industrial Cryptography (Lecture Notes in Computer Science no. 1528), Springer-Verlag, pp. 18-48, 1998.

[8] Encryption Technology White paper, http://security.resist.ca/crypt.htm.

[9] G. Gong, K. C. Gupta, M. Hell, and Y.Nawaz, “Towards a general RC4-like keystream generator,” in SKLOIS Conf. Inf.Security and Cryptol., CISC 2005 LNCS 3822. New York: Springer-Verlag, 2005, pp.162–174.

[10] S. Paul and B. Preneel, “On the (in)securityof stream ciphers based on arrays andmodular addition,” in Adv. Cryptol.—Asiacrypt 2006, LNCS4284. New York:Springer-Verlag, 2006, pp. 69–83.