Diffusion Analysis of a Scalable Fiestel Network

Subariah Ibrahim and Mohd Aizaini Maarof

***Abstract*—**A generalization of the concepts of Feistel Networks (FN), known as Extended Feistel Network (EFN) is examined. EFN splits the input blocks into n > 2 sub-blocks. Like conventional FN, EFN consists of a series of rounds whereby at least one sub-block is subjected to an *F* function. The function plays a key role in the diffusion process due to its completeness property. It is also important to note that in EFN the *F*-function is the most computationally expensive operation in a round. The aim of this paper is to determine a suitable type of EFN for a scalable cipher. This is done by analyzing the threshold number of rounds for different types of EFN to achieve the completeness property as well as the number of F-function required in the network. The work focuses on EFN-Type I, Type II and Type III only. In the analysis it is found that EFN-Type II and Type III diffuses at the same rate and both are faster than Type-I EFN. Since EFN-Type-II uses less *F* functions as compared to EFN-Type III, therefore Type II is the most suitable EFN for use in a scalable cipher.

***Keywords*—** Cryptography, Extended Feistel Network, Diffusion Analysis.

I. Introduction

A block cipher can be characterized by its basic parameters: the block size, *b*, the key size, *k* and the number of rounds, *r*. The security of a cipher depends on these parameters, the larger the size or the number, the more secure the cipher is. A cipher is called scalable, if these basic parameters are alterable by design [1]. A scalable cipher is becoming a trend since a call for Advanced Encryption Standard (AES), which requested for a minimal requirement of 128-bit block size and 128/192/256 bit key sizes [2].

Scalability of key can easily be achieved since iterative block ciphers usually use a master key for generating round keys or sub-keys by using key scheduling algorithm. Scalability of number of rounds can also be easily achieved. However, scalability of block size can be achieved through a modification of the size of the basic parameters or the modification of the structure of the cipher [1][3]. The scalability through primitives will take more memory space to store the cipher code. According to [1], doubling the size of *f* requires exponentially more resources. A scheme that may be used in scaling through structure is by doubling the number of processed sub-blocks while maintaining the original cryptographic primitives. This scheme is employed in RC6 [4], CAST256 [5] and MARS [6]. This structure is known as Extended Feistel Network (EFN).

EFN splits the input blocks into *n* > 2 sub-blocks. Like conventional Feistel Networks (FN), EFN consists of a series of rounds whereby at least one sub-block is subjected to an *F*-function. The function plays a key role in the diffusion process due to its completeness property [7]. A completeness property is an important cryptographic criterion whereby all output bits of a cipher are dependent on all input bits and vice-versa. This property was defined by Kam and Davida [8].

The aim of this paper is to determine an EFN scheme, which is suitable for a scalable cipher. For a scalable cipher, we need to choose a scheme that diffuses very fast so that as the block size increases, the number of rounds required does not increase exponentially. It is also important to note that in EFN the *F*-function is the most computationally expensive operation in a round. Therefore this work will also consider the number of *F*-function needed to achieve completeness. The work focuses on EFN-Type I, Type II and Type III only. Some related work in the analysis of EFN schemes were discussed in [9][5][10]. The diffusion analysis technique that is employed in this work is similar to Adams *et al*. [5] in his analysis for CAST-256.

First, some definitions of EFN schemes will be given in section II. A diffusion analysis of the three EFN schemes is discussed in section III and finally a conclusion is made in section IV.

II. Scalable Block-Size Cipher With Extended Feistel Network

As mentioned in section I, the fundamental mechanism for scaling the block-size to a larger one is by using an EFN structure. In a conventional FN, the plaintext block is divided evenly into two sub-blocks. The round function *F* operates on the right sub-block and then combined with the left sub-block via bitwise exclusive or (XOR). The two sub-blocks are then swapped and become the input of the next round. However, an EFN splits the input block into n > 2 sub-blocks [7]. These sub-blocks are then mixed through repeated application of

Manuscript received March 3, 2005. This work was supported by Universiti Teknologi Malaysia.

Subariah Ibrahim is with Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Skudai 81300, Johore, Malaysia (phone: 607-553-2386; fax: 607-556-5044; e-mail: subariah@ fsksm.utm.my).

Mohd Aizaini Maarof is with Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Skudai 81300, Johore, Malaysia (phone: 607-553-2009; fax: 607-556-5044; e-mail: maarofma@ fsksm.utm.my).

keyed, non-linear *F*-functions in order to generate a permutation of the input block [11]. The swapping of sub-blocks can be viewed as a circular shift. There are various types of transformations in EFN. The rate of diffusion for different types of transformations is different. For the purpose of this paper, we described three types of EFN, namely, EFN-Type I, EFN-Type II and EFN-Type III.

EFN-Type I employs only one *F*-function in its design. The output cipher-text for each round can be described as follows:

$$C_1, C_2, \ldots, C_{n-1}, C_n = P_2 \oplus F(P_1), P_3, \ldots, P_{n-1}, P_n, P_1. \quad (1)$$

where P_i is the *i*th sub-block. EFN-Type II uses one *F*-function for every two consecutive sub-blocks. Similarly, this type of transformation can be defined by:

$$C_1, C_2, \ldots, C_{n-1}, C_n = P_2 \oplus F(P_1), P_3, P_4 \oplus F(P_3), \ldots, P_n \oplus F(P_{n-1}), P_1. \quad (2)$$

Finally EFN-Type III has one *F*-function for every sub-block and is defined as follows:

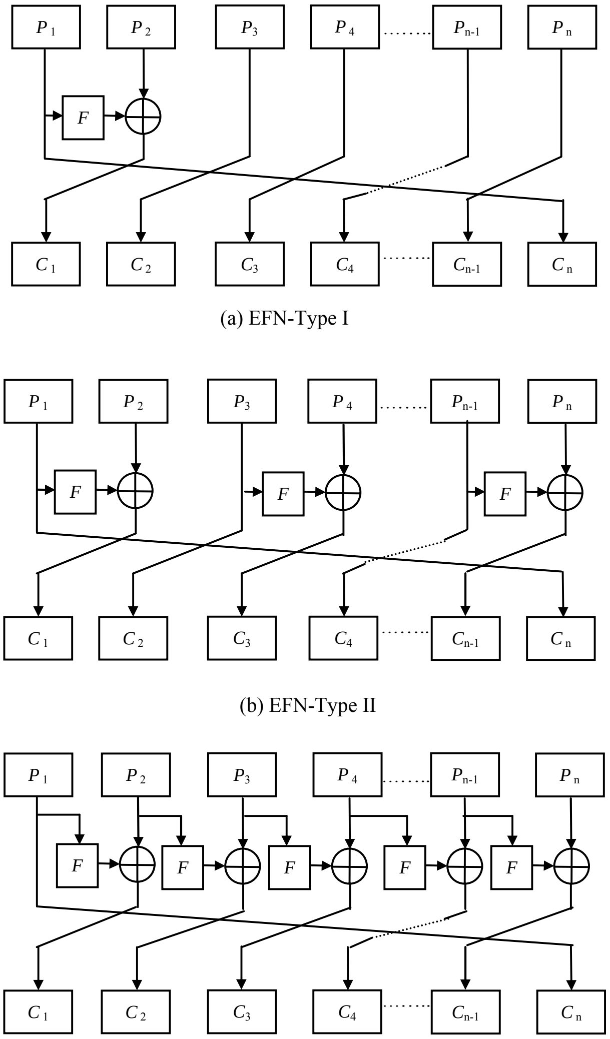
$$C_1, C_2, \ldots, C_{n-1}, C_n = P_2 \oplus F(P_1), P_3 \oplus F(P_2), \ldots, P_n \oplus F(P_{n-1}), P_1. \quad (3)$$

The structures for the three EFN transformations are depicted in FIG. I.

III. Diffusion Analysis

Diffusion is a spread of the influence of input bits to output bits. An important cryptographic criterion of a cipher with regard to diffusion is completeness property whereby all output bits of a cipher are dependent on all input bits and vice versa. In this section we will discuss the analysis on the completeness property of EFN-Type I, EFN-Type II and EFN-Type III scalable structure models. In this analysis, it is assumed that the round function, *F*, is complete.

In this work, we employ the diffusion analysis used in [5] whereby the output sub-block for each round is determined. The sub-block is represented by P_i, for example, a cipher with 4 sub-blocks is represented by P_1, P_2, P_3, and P_4. For the output sub-block for each round, the letter in front the bracket represents the block affected by the round function and the bracketed letters represent the blocks which influence the affected block (i.e. the blocks which influence input bits to the round function). Table I illustrates the dependencies of the cipher data at the output of each round for DES. The table indicates that in DES, after one round, the bits corresponding to the plaintext sub-block P_1 are now non-degenerate in the bits of plaintext sub-block P_2. After round two the sub-block, P_2 is influenced by all the bits of all sub-blocks. Then after the third round, a complete dependency of the output bits on the input bits has been achieved since both sub-blocks are influenced by all bits of the plaintext. Therefore, we can say that DES achieves complete dependency after 3 rounds.



(a) EFN-Type I

(b) EFN-Type II

(c) EFN-Type II

Fig. I EFN transformation structures

TABLE I
Dependencies for DES cipher

Round	Dependencies
1	$P_1(P_2)$
2	$P_2(P_1)$
3	$P_1(P_1,P_2)$

In this work, we analyzed diffusion rates for 3-6 sub-blocks for EFN-Type I and EFN-Type III models. For EFN-Type II models, we analyzed for 4 and 6 sub-blocks only since the model only use even number of sub-blocks. Table II illustrates the comparison of the threshold number of rounds needed to achieve complete dependency for these three models. As shown in the table, EFN-Type I model achieves complete dependency after 5, 7,9 and 11 number of rounds for 3, 4, 5 and 6 sub-blocks respectively. EFN-Type II model achieves complete dependency after 5 and 7 rounds for 4 and 6 sub-blocks respectively while EFN-Type II model requires 4, 5, 6 and 7 number of rounds for 3, 4, 5, and 6 sub-blocks respectively. From these analyses, we can conclude that for n sub-blocks, EFN-Type I requires ($2n$-1) number of rounds while EFN-Type II and EFN-Type III requires n+1 number of rounds to achieve completeness. The threshold number of rounds needed to achieve completeness for each model is listed in column 2 of TABLE III. Please refer to the APPENDIX for the detailed diffusion analysis.

TABLE II
COMPARISON OF THRESHOLD NUMBER OF ROUNDS FOR EFN-TYPE I, EFN-TYPE II AND EFN-TYPE III MODELS FOR DIFFERENT BLOCK SIZE.

Block Size (bits)	Number of Sub-blocks	Threshold Number of Rounds To Achieve Completeness		
		EFN-Type I	EFN-Type II	EFN-Type III
64 (DES)	2	3	3	3
96	3	5	n.a.	4
128	4	7	5	5
160	5	9	n.a.	6
196	6	11	7	7

Next, we analyzed the number of F functions needed for the threshold number of rounds required by each transformation to achieve completeness. Column 3 of TABLE 3 compares the performance of the three types of EFN transformation in terms of the number of F-functions required to achieve completeness. The table indicates that EFN-Type I, EFN-Type II and EFN-Type III requires $(2n - 1)$, $n(n + 1)/2$ and $n(n + 1)$ F functions respectively to achieve completeness.

TABLE III
RESULTS OF DIFFUSION ANALYSIS

EFN Type	Threshold No. of Rounds to Achieve Completeness	No. of F-functions
EFN-Type I	$2n - 1$	$2n - 1$
EFN-Type II	$n + 1$	$n(n + 1)/2$
EFN-Type III	$n + 1$	$n(n + 1)$

The analysis shows that EFN-Type I diffuses slower than EFN-Type II and EFN-Type III, while EFN-Type II and EFN-Type III diffuse at the same rate. However, EFN-Type III requires twice the number of F functions needed by EFN-Type II to achieve completeness. Since EFN-Type II model requires less F function computations than EFN-Type III model, therefore EFN-Type II model is the most suitable model amongst the three models for use in a scalable cipher.

IV. CONCLUSION

In this paper we studied the diffusion analysis of three types of EFN transformation by determining the threshold number of rounds as well as the number of F functions computations needed to achieve completeness property. This analysis is done in order to determine a suitable model for a scalable cipher that support various block-sizes. From the analysis, it was found that EFN-Type II model is the optimal transformation model amongst the three models studied after considering the threshold number of rounds as well as the number of F functions computations needed. EFN-Type II model is also suitable for cipher that employs many sub-blocks as the model can achieve the same level of security with less number of rounds as compared to EFN-Type I model. EFN-Type III is computationally expensive because it requires many F-functions for its transformation.

APPENDIX

TABLE A-I
DEPENDENCIES FOR EFN-TYPE I TRANSFORMATION

Round	Dependencies			
	3 sub-blocks	4 sub-blocks	5 sub-blocks	6 sub-blocks
1	$P_2(P_1)$	$P_2(P_1)$	$P_2(P_1)$	$P_2(P_1)$
2	P_3 (P_1,P_2)	P_3 (P_1,P_2)	$P_3(P_{1,}$ $P_2)$	$P_3(P_1,P_2)$
3	P_1 (P_1,P_2,P_3)	P_4 (P_1,P_2, P_3)	$P_4(P_{1,}$ $P_2, P_3)$	$P_4(P_1,P_2,P_3)$
4	P_2 (P_1,P_2,P_3)	P_1 (P_1,P_2, P_3, P_4)	$P_5(P_{1,}$ $P_2, P_3, P_4)$	$P_5(P_1,P_2,P_3,P_4)$
5	P_3 (P_1,P_2,P_3)	P_2 (P_1,P_2, P_3, P_4)	$\mathrm{P_1(P_{1,}P_2,P_3, P_4, P_5)}$	$P_6(P_1,P_2,P_3,P_4,P_5)$
6		P_3 (P_1,P_2, P_3, P_4)	$P_2(P_{1,}P_2,P_3, P_4, P_5)$	$P_1(P_1,P_2,P_3,P_4,P_5,P_6)$
7		P_4 (P_1,P_2, P_3, P_4)	$P_3(P_{1,}P_2,P_3, P_4, P_5)$	$P_2(P_1,P_2,P_3,P_4,P_5,P_6)$
8			$P_4(P_{1,}P_2,P_3, P_4, P_5)$	$P_3(P_1,P_2,P_3,P_4,P_5,P_6)$
8			$P_5(P_{1,}P_2,P_3, P_4, P_5)$	$P_4(P_1,P_2,P_3,P_4,P_5,P_6)$
10				$P_5(P_1,P_2,P_3,P_4,P_5,P_6)$
11				$P_6(P_1,P_2,P_3,P_4,P_5,P_6)$

TABLE A-II
DEPENDENCIES FOR EFN-TYPE II TRANSFORMATION

Round	Dependencies				
	4 sub-blocks		6 sub-blocks		
1	$P_2(P_1)$	$P_4(P_3)$	$P_2(P_1)$	$P_4(P_3)$	$P_6(P_5)$
2	P_3 (P_1,P_2)	P_1 (P_3,P_4)	$P_3(P_1,P_2)$	$P_5(P_3,P_4)$	$P_1(P_5,P_6)$
3	P_4 (P_1,P_2, P_3)	P_2 (P_1,P_3, P_4)	$P_4(P_1,P_2,P_3)$	$P_6(P_3,P_4,P_5)$	$P_2(P_1,P_5, P_6)$
4	$P_1(P_1,P_2, P_3,P_4)$	$P_3(P_1,P_2, P_3,P_4)$	$P_5(P_1,P_2,P_3,P_4)$	$P_1(P_3,P_4,P_5,P_6)$	$P_3(P_1,P_2, P_5,P_6)$
5	$P_2(P_1,P_2, P_3,P_4)$	$P_4(P_1,P_2, P_3,P_4)$	$P_6(P_1,P_2,P_3,P_4,P_5)$	$P_2(P_1,P_3,P_4,P_5,P_6)$	$P_4(P_1,P_2, P_3,P_5,P_6)$
6			$P_1(P_1,P_2,P_3,P_4,P_5, P_6)$	$P_3(P_1,P_2,P_3,P_4,P_5, P_6)$	$P_5(P_1,P_2, P_3,P_4,P_5, P_6)$
7			$P_2(P_1,P_2,P_3,P_4,P_5, P_6)$	$P_4(P_1,P_2,P_3,P_4,P_5, P_6)$	$P_6(P_1,P_2, P_3,P_4,P_5, P_6)$

TABLE A-III
DEPENDENCIES FOR EFN-TYPE III TRANSFORMATION

(a) 3 AND 4 SUB-BLOCKS

Round	Dependencies				
	3 sub-blocks		4 sub-blocks		
1	$P_2(P_1)$	$P_3(P_2)$	$P_2(P_1)$	$P_3(P_2)$	$P_4(P_3)$
2	P_3 (P_1,P)	$P_1(P_2,P_3)$	$P_3(P_1,P_2)$	$P_4(P_2, P_3)$	$P_1(P_3,P_4)$
3	$P_1(P_1,P_2, P_3)$	$P_2(P_1,P_2,P_3)$	P_4 (P_1,P_2, P_3)	$P_1(P_2, P_3, P_4)$	$P_2(P_1,P_3, P_4)$
4	P_2 (P_1,P_2, P_3)	$P_3(P_1,P_2,P_3)$	P_1 (P_1,P_2, P_3, P_4)	$P_2(P_1,P_2, P_3, P_4)$	$P_3(P_1,P_2, P_3, P_4)$
5			$P_2(P_1,P_2, P_3, P_4)$	$P_3(P_1,P_2, P_3, P_4)$	P_4 (P_1,P_2, P_3, P_4)

(b) 5 SUB-BLOCKS

Round	Dependencies			
	5sub-blocks			
1	$P_2(P_1)$	$P_3(P_2)$	$P_4(P_3)$	$P_5(P_4)$
2	$P_3(P_1,P_2)$	$P_4(P_2,P_3)$	$P_5(P_3, P_4)$	$P_1(P_4,P_5)$
3	$P_4(P_1, P_2, P_3)$	$P_5(P_2, P_3, P_4)$	$P_1(P_3, P_4, P_5)$	$P_2(P_1, P_4, P_5)$
4	$P_5(P_1, P_2, P_3, P_4)$	$P_1(P_2, P_3, P_4, P_5)$	$P_2(P_1,P_3,P_4, P_5)$	$P_3(P_1, P_2,P_4,P_5)$
5	$P_1(P_1,P_2, P_3,P_4, P_5)$	$P_2(P_1,P_2, P_3, P_4,P_5)$	$P_3(P_1, P_2,P_3,P_4,P_5)$	$P_4(P_1, P_2, P_3,P_4,P_5)$
6	$P_2(P_1,P_2, P_3,P_4, P_5)$	$P_3(P_1,P_2, P_3,P_4, P_5)$	$P_4(P_1,P_2,P_3, P_4, P_5)$	$P_5(P_1,P_2, P_3,P_4, P_5)$

(c) 6 SUB-BLOCKS

Round	Dependencies				
	6 sub-blocks				
1	$P_2(P_1)$	$P_3(P_2)$	$P_4(P_3)$	$P_5(P_4)$	$P_6(P_5)$
2	$P_3(P_1,P_2)$	$P_4(P_2,P_3)$	$P_5(P_3, P_4)$	$P_6(P_4,P_5)$	$P_1(P_5,P_6)$
3	$P_4(P_1,P_2, P_3)$	$P_5(P_2, P_3, P_4)$	$P_6(P_3, P_4, P_5)$	$P_1(P_4, P_5, P_6)$	$P_2(P_1,P_5, P_6)$
4	$P_5(P_1,P_2, P_3,P_4)$	$P_6(P_2, P_3, P_4, P_5)$	$P_1(P_1,P_3, P_4,P_5)$	$P_2(P_1,P_2, P_4,P_5)$	$P_3(P_1,P_2, P_5,P_6)$
5	$P_6(P_1,P_2, P_3,P_4,P_5)$	$P_1(P_1,P_2, P_3,P_4,P_5)$	$P_2(P_1, P_2, P_3,P_4,P_5)$	$P_3(P_1, P_2, P_3,P_4,P_5)$	$P_4(P_1,P_2, P_3, P_5,P_6)$
6	$P_1(P_1,P_2, P_3,P_4,P_5, P_6)$	$P_2(P_1,P_2, P_3,P_4, P_5,P_6)$	$P_3(P_1,P_2, P_3,P_4, P_5,P_6)$	$P_4(P_1,P_2, P_3, P_4,P_5,P_6)$	$P_5(P_1,P_2,P_3, P_4,P_5, P_6)$
7	$P_2(P_1,P_2, P_3,P_4,P_5, P_6)$	$P_3(P_1,P_2, P_3,P_4, P_5,P_6)$	$P_4(P_1,P_2, P_3,P_4,P_5, P_6)$	$P_5(P_1,P_2, P_3,P_4,P_5, P_6)$	$P_6(P_1,P_2,P_3, P_4,P_5, P_6)$

REFERENCES

[1] Canda, V. and Trung, T. Scalable Block Ciphers Based on Feistel-Like Structure). *Proceedings of Tatracrypt 2001, Tatra Mountains, Mathematica Pub*. 25:39-57, 2002.

[2] Biryukov,A. Block Ciphers and Stream Ciphers: The State of the Art. *State of the Art in Applied Cryptography, Course on Computer Security and Industrial Cryptogtraphy, LNCS.* B.Preneel (Ed.) Springer-Verlag, 2003.

[3] Fibikova, L., Provable Secure Scalable Block Ciphers. PHD Dissertation. University Duisburg – Essen, Germany, 2003.

[4] Rivest, R.L., Robshaw, M.J.B., Sidney, R. and Yin, Y.L., The RC6 Block Cipher. *NIST AES,* 1998.

[5] Adams, C., Heys, H.M., Tavares, S.E. and Wiener, M. An Analysis of the CAST-256 Cipher. 1999. Available: http://www.engr.mun.ca/~howard/PAPERS/cast256.ps.

[6] Burwick, C., Coppersmith, D., D'Avignon, E., Genmaro, R., Halevi, S., Jutla, C., Matyas, S.M., O'Connor, L., Peyravian, M., Safford, D. and Zunic, N. (1999). MARS - A Candidate Cipher for AES. Available: http://www.research.ibm.com/security/mars.pdf

[7] Nakahara, J., Vanderwalle, J. and Preneel, B. Diffusion Analysis of Feistel Networks. *Proceedings of 20th Symposium on Information Theory in the Benelux*, Hasrode, Belgium. May 27-28. pp. 101-108, 1999.

[8] Kam, J.B. and Davida G.I. Structured Design of Substitution-Permutation Encryption Networks. *IEEE Transactions on Computers*, Vol. C-28, No. 10, pp. 747-753, 1979.

[9] Moriai S. and Vaudenay, S. Advances in Cryptology ASIACRYPT'00, Kyoto, Japan. *LNCS* 1976, pp. 289-302. Springer-Verlag, 2000.

[10] Zheng, Y. On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypothesis. Advances in Cryptology – *In Proceedings of CRYPTO'89 LNCS* 435, pp. 461-480. Springer-Verlag, 1989.

[11] Schneier, B. and Kelsey, J. Unbalanced Feistel Networks and Block Cipher Design. *In Proceedings of Fast Software Encryption 1996. LNCS 1039*, pp. 121-144, Springer-Verlag, 1996.