FeW: A Lightweight Block Cipher

Manoj Kumar[1, 2], Saibal K. Pal[1] and Anupama Panigrahi[2]
[1]Scientific Analysis Group, DRDO, Delhi, INDIA
[2]Department of Mathematics, University of Delhi, INDIA
mktalyan@yahoo.com

Abstract: *In this paper, we propose a new lightweight block cipher called **FeW**[1] which encrypts 64-bit plaintext using key size 80/128 bits and produces 64-bit ciphertext. FeW is a software oriented design with the aim of achieving high efficiency in software based environments. We use a mix of Feistel and generalised Feistel structures (referred as Feistel-M structure hereinafter) to enhance the security of our design against basic cryptanalytic attacks like differential, linear, impossible differential and zero correlation attacks. Security analysis of this scheme proves its strength against present day cryptanalytic attacks.*

Keywords: Block Cipher, Feistel structure, Generalised Feistel structure, Lightweight Cryptography, SPN

1. Introduction

Lightweight cryptography [26] has emerged as a vast research direction in the area of cryptography. Research in this direction started in the beginning of 21st century to meet the requirement of cryptographic algorithms requiring very low implementation area and less power consumption. There was requirement for product designers and developers to provide security features in tiny and handheld devices. This was the time when Rijndael [10] was selected as AES and there was a need for lightweight ciphers for specific applications. RFID tags and sensor networks are the examples of products that employ lightweight cryptographic algorithms. Lightweight block ciphers like PRESENT [4], HIGHT [14], LBlock [38], TWINE [34], SIMON and SPECK family [7], TEA [39], DES Light weight variant [23] and other lightweight designs [8,12,13,19,31,40,15,22,35] with different design constructions [20] have been published in last 15 years. PRESENT and CLEFIA have been chosen as a lightweight encryption standard by International Organisation of Standardisation (ISO) and International Electro-technical Commission. Wide publicity of PRESENT, inspired researchers to designs new and more efficient lightweight block cipher. Various new cryptanalytic attacks and their combinations [21] have also been discovered and applied on block ciphers in last few years. Some cryptanalytic attacks have been applied on full round ciphers and some have been shown to be prone to various attacks.

---

[1] *FeW* refer to **Fe**ather **W**eight, which is a term in certain sports between lightweight and bantamweight.

Lightweight block cipher designs are mainly based on two structures: Feistel and SPN. The concept of confusion and diffusion given by Shannon [29] is used in both structures to design the secure cryptographic algorithms. These design principles are still the best concept to design a new block ciphers. Feistel structure was proposed by Horst Feistel and the first block cipher design based on this structure was LUCIFER [28]. In this structure, input plaintext is encrypted by dividing it in two equal parts and using a round function (comprising of non-linear substitution and linear permutation which gives confusion and diffusion) on one part while the other part remains unchanged for the next round. DES [6] is the first widely used Feistel Network based block cipher design which was in use for almost two decades. Schneier and Kelsey [32] examined the concept given by Feistel and they generalized this structure and called it unbalanced Feistel networks (UFNs). UFNs consists a series of rounds in which one part of block operates on the rest of the block. However, in a UFN the two parts need not be of same size. But UFNs did not receive much attention from the cryptographic community for designing the block cipher. The first choice between balance Feistel network (BFN) and UFN is obviously BFN based designs. Generalized balance Feistel networks are the generalization of Balance Feistel which encrypts the plaintext block by dividing it into n equal parts. If the confusion and diffusion is applied on the whole block length in each round while encryption then the structure is called SPN. In this structure we require that round subkeys are of same size as block length. The current cryptographic standard for block ciphers AES [10] have been designed using the SPN structure.

As compared to SPN based ciphers, Feistel based ciphers seem to be a better choice for lightweight ciphers as it does not need inversion of the round function and inverse of S-box involved in round function. Feistel based design performs less computation as compared to SPN based designs, because half of the input block is processed through round function, whereas SPN based designs apply substitution and permutation on full input block in every round. Therefore we choose the Feistel structure with SPN round function to design a new lightweight block cipher. Some previous designs also have used similar operations in round function as we have used in our design. SMS4 [9] block cipher is used in Chinese WAPI standard which uses shifts and xor on 32 bit words in its round function and key scheduling. We have used two different shift and xor operations on 16 bit words inside the round function. Key schedule of *FeW* is designed using the key expansion concept similar to the PRESENT. Generalised Feistel based designs CLEFIA [36] have used two different round functions but we have used two different functions which are applied on mixed input data of two 16 bit Feistel branches. Our design is based on Feistel-M structures, which proves to be very helpful in enhancing the security of our design against cryptanalytic attacks.

This paper is organised as follows. In section 2, we describe the design specifications of lightweight block cipher *FeW* in detail. Key schedule for key size 80-bit is presented in section 3. Security evaluation of *FeW* against some basic cryptanalytic attacks is described in section 4. Finally, we conclude the paper in section 5.

Notations: We have used the following notations in this paper while describing the lightweight block cipher *FeW*:

- P_m: 64-bit input plaintext block
- C_m: 64-bit output ciphertext block
- MK-80: 80-bit user supplied key
- MK-128: 128-bit user supplied key
- RK_i: 16-bit subkey extracted from Key register MK
- K_i: 32-bit subkey for round i (concatenation of RK_{2i} and RK_{2i+1})
- F: Round function
- WF_1: Weight Function 1 used inside F
- WF_2: Weight Function 2 used inside F
- $\oplus$ Bitwise exclusive-OR operation
- <<<n Left cyclic shift by *n* bits
- >>n Right shift by *n* bits
- $[i]_2$ Binary representation of integer i
- ||: Concatenation of two bit strings
- &: Bitwise And between two bit strings

2. *FeW*: Lightweight Block Cipher

We describe the Encryption algorithm and Key Schedule of lightweight block cipher *FeW* in this section*:*

2.1 Specifications of *FeW*

FeW encrypts plaintext in blocks of size 64 bits (This is the commonly preferred block size in case of lightweight block ciphers). Design of *FeW* (Fig.1) is based on Feistel-M structure which is broadly a Balanced Feistel based design but its round function process 32 bit word like generalised Feistel based designs. The round function F uses two different functions WF_1 and WF_2 and applies these on two 16 bit words. This type of mixing method is used first time in a block cipher. We have shown that this significantly improves the immunity of our design against cryptanalytic attacks.

FeW takes 64-bit of plaintext data as input and produces a 64-bit data of ciphertext as output. There are total 32 rounds in *FeW*. We obtain 64-bit ciphertext by swapping the output words of the last round. *FeW* uses two options for the size of Master key MK: 80 bits and 128 bits. Based on two key sizes, we name the two versions of *FeW*, the first version with 80-bit key size as *FeW-80* and the second version with 128-bit key size as *FeW-128*.

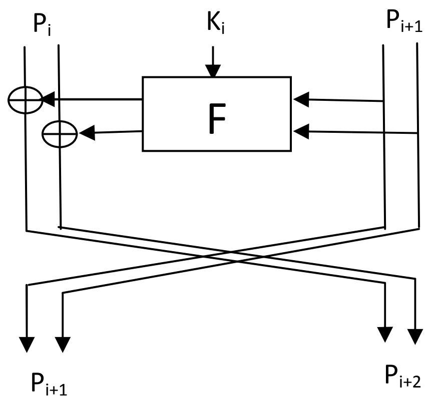

Fig.1: One round of *FeW* (Feistel-M)

2.2 Encryption Algorithm

First we divide 64-bit plaintext P_m into two halves namely P_0 and P_1. Each of these halves is of size 32-bit. We have 64-bit input plaintext P_m as concatenation of two 32-bit words P_0 and P_1 as follows:

$$P_0 \| P_1 \leftarrow P_m$$

Encryption procedure of *FeW* is described as follows:

(a) For i =0 to 31, apply round functions F on 32-bit word P_{i+1} and xor it with P_i to produce P_{i+2}:

$$P_{i+2} \leftarrow P_i \oplus F(P_{i+1}, K_i)$$

(b) Apply swap function on the output of the last round:

$$(P_{33}, P_{32}) \leftarrow (P_{32}, P_{33})$$

(c) We obtain two 32 bit ciphertext words:

$$(C_0, C_1) \leftarrow (P_{33}, P_{32})$$

Finally, we obtain the 64-bit ciphertext C_m as concatenation of C_0 and C_1 as follows:

$$C_m \leftarrow C_0 \| C_1$$

Now we describe below the Round functions F in detail:

2.3 Round function F

F is the round function of *FeW* and its internal structure is shown in Fig 2. It takes 32 bit input X_i and produces 32 bit output Y_i.

$$F: \{0, 1\}^{32} \longrightarrow \{0, 1\}^{32}$$

We have used two different weight functions WF_1 and WF_2 inside F, both of these functions take 16 bit input and produce 16 bit output. Weight functions WF_1 and WF_2 are described in section 2.3.1 and 2.3.2 in detail. In each round, F is applied on 32 bit input X_i as follows:

(i) $X_i \oplus K_i$

(ii) $C_{(8)} \| D_{(8)} \| E_{(8)} \| F_{(8)}$

(iii) $(A = C_{(8)} \| F_{(8)}) \| (B = E_{(8)} \| D_{(8)})$

A & B are processed through weight functions WF_1 and WF_2. Finally, 32-bit output Y_i is the concatenating of 16-bit outputs from WF_1 and WF_2. This 32-bit output Y_i from round function F is xored with 32-bit word P_i to get P_{i+2} as described below:

$$P_{i+2} \leftarrow P_i \oplus F(X_i)$$

$$\text{i.e. } P_{i+2} \leftarrow P_i \oplus Y_i$$

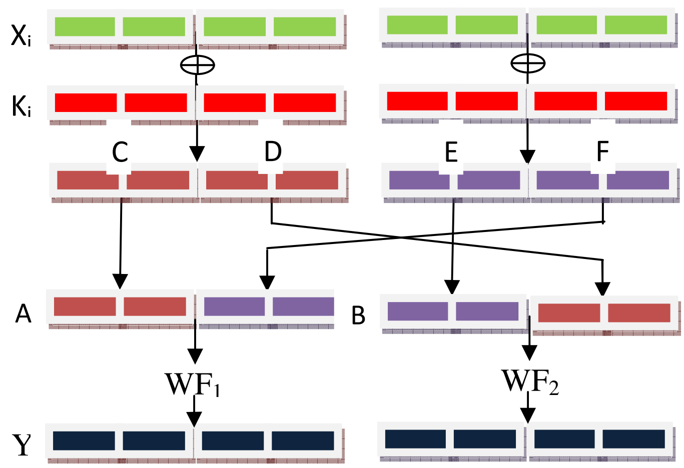

Fig. 2: Round Function F

2.3.1 Weight Function WF_1

This function takes 16-bit input and produces 16-bit output. Weight function WF_1 consists of application of S-box 4 times in parallel as non-linear operation and cyclic shifts and exclusive-or operation as linear mixing operation L_1. Weight function WF_1 is described below in detail:

$$WF_1: \{0, 1\}^{16} \longrightarrow \{0, 1\}^{16}$$

$$Y \leftarrow WF_1(A = A_0 \parallel A_1 \parallel A_2 \parallel A_3\ )$$

First, we apply 4x4 S-box in parralal 4 times on A to get U then apply cyclic shifs on U and xor these with U to get Y as output of function WF_1 as follows:

$$U_0 \leftarrow S(A_0)$$
$$U_1 \leftarrow S(A_1)$$
$$U_2 \leftarrow S(A_2)$$
$$U_3 \leftarrow S(A_3)$$

$$U \leftarrow (U_0 \parallel U_1 \parallel U_2 \parallel U_3)$$

$$Y \leftarrow (U \oplus U{<}{<}{<}1 \oplus U{<}{<}{<}5 \oplus U{<}{<}{<}9 \oplus U{<}{<}{<}12)$$

2.3.2 Weight Function WF_2

Similar to WF_1, WF_2 also takes 16-bit input and produces 16-bit output. WF_2 consists of application of S-box 4 times in parallel and apply cyclic shifts on V and xor these with V to get Z as output of WF_2 as linear mixing operation L_2 which is different from L_1. WF_2 is described below in detail:

$$WF_2: \{0, 1\}^{16} \longrightarrow \{0, 1\}^{16}$$

$$Z \longleftarrow WF_2(B = B_0 \parallel B_1 \parallel B_2 \parallel B_3)$$

First, we apply 4x4 S-box in parralal 4 times on B to get V then apply shift and xor V to get Z as output of function WF_2 as follows:

$$V_0 \longleftarrow S(B_0)$$

$$V_1 \longleftarrow S(B_1)$$

$$V_2 \longleftarrow S(B_2)$$

$$V_3 \longleftarrow S(B_3)$$

$$V \longleftarrow V_0 \parallel V_1 \parallel V_2 \parallel V_3$$

$$Z \longleftarrow V \oplus V{<}{<}{<}4 \oplus V{<}{<}{<}7 \oplus V{<}{<}{<}11 \oplus V{<}{<}{<}15$$

2.4 S-Box

We have used the same 4x4 S-box in encryption, decryption and key schedule of lightweight block cipher *FeW-80* and *FeW-128.* This S-box has already been used in block cipher HummingBird2 [11]. Saarinan [30] also has given cryptographic analysis of all 4x4 bit S-boxes and this S-box falls in the category of Golden S-boxes:

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	2	E	F	5	C	1	9	A	B	4	6	8	0	7	3	D

Table 1: S-box S

2.5 Key Schedule for MK-80

There is no related key attack reported on PRESENT like key schedule till now. We also prefer the same type of key schedule for our design. First, store MK-80 in a key register called MK as

$$MK= k_0\, k_1\, k_2\, k_3 \ldots\ldots\ldots\ldots k_{78}\, k_{79}.$$

We obtain round subkeys RK_0 by extracting leftmost 16 bits of current contents of MK and proceed in the following way to obtain the other round subkeys:

a. While $i < 64$, update the register MK in the following steps:
   1. MK<<<13
   2. $[k_0\, k_1\, k_2\, k_3] \leftarrow S[k_0\, k_1\, k_2\, k_3]$
      $[k_{64}\, k_{65}\, k_{66}\, k_{67}] \leftarrow S[k_{64}\, k_{65}\, k_{66}\, k_{67}]$
      $[k_{76}\, k_{77}\, k_{78}\, k_{79}] \leftarrow S[k_{76}\, k_{77}\, k_{78}\, k_{79}]$
   3. $[k_{68}\, k_{69}\, k_{70}\, k_{71}\, k_{72}\, k_{73}\, k_{74}\, k_{75}] \leftarrow [k_{68}\, k_{69}\, k_{70}\, k_{71}\, k_{72}\, k_{73}\, k_{74}\, k_{75}] \oplus [i]_2$

b. Increment *i* by 1 and extract leftmost 16 bits of current contents of MK as round subkey RK_i.

3. Security Analysis

There is a large variety of cryptanalytic attacks which can be applied on block ciphers. We give secrurity estimates of our design against some basic cryptanalytic attacks in this section.

3.1 Differential Cryptanalysis

Differential attack [6] is one of the most basic cryptanalytic attacks applied on block ciphers. This attack was invented by Biham and Shamir in 1990 and applied on DES. This attack exploits the high probability differences in input and ouput of an encryption system and these high probability input and output occurences of certain pairs are used to recover round subkeys from the outermost rounds. Linear components of a cipher produces the certain outputs with probability 1, while this is not the case for the non linear components (S-box in our design). These are examined and high probability input and output differences of these componenets (S-box) are used to form differential trail of the cipher by joining 1 round high probability differentials trails. *FeW* uses a 4x4 S-box as its only non linear component. We give a Difference Distribution table (DDT) for this S-box by counting the occurences of all possible input and output differences. DDT (16x16) of S-box is given in table 5 (Appendix D).

Maximum differential probability for arbitrary input difference producing a output difference in a single S-box application is $4/16 = 2^{-2}$. This value ensures that even if there is only one active S-box in each round, still differential attack will require 2^{64} chosen palintexts(full codebook) to distinguish it from random permutation.

3.1.1 Experimental results on L_1 and L_2

We applied L_1 and L_2 on all possible input differences and observed the output differences using computer programs, the following observations (table 2) are made between input and output differences which are very useful in proving our design secure against the Differential and Linear attacks:

Minimum number of non zero nibbles (L_1 and L_2)	
Input Difference	Output Difference
1	4
2	3
3	2
4	1

Table 2: Number of non zero nibbles

There are two cases on the bases of observations made on linear permutation layers of round functions WF_1 and WF_2:

1. Input difference with 1 non zero nibble gives output difference with at least 4 non zero nibbles and next round input difference with 4 non zero nibbles produces output difference with 1 non zero nibble and vice versa.
2. Input difference with 2 non-zero nibbles gives 3 non zero nibbles and the process continues with 3 non zero nibbles as input to the next round and 2 non zero nibbles as output and vice versa.

Branch number of a function is defined by Rijmen [27] and Kanda [16] for SPN based designs and Feistel based designs with SPN type round function. We define below the Branch number of the linear permutation layers used in *FeW* and differential & linear Branch number of *FeW*. We use the similar techniques as in [18,37] to show the resistance of *FeW* to Differential and linear attacks.

Definition 1: (Branch Number) If X is 16 bit input to the function *f* and X is written as concatenation of 4 nibbles x_0, x_1, x_2 and x_3 each of size 4 bit. By defining the number of non zero nibbles in *f* by Hw(*f*), we define the branch number of the function

$$f: \{0,1\}^{16} \rightarrow \{0,1\}^{16}$$

by $\beta(f)$ as follows:

$$\beta(f) = \min_{X \neq 0, X \in \{0,1\}^{16}} (\mathrm{Hw}(X) + \mathrm{Hw}(f(X)))$$

Definition 2: Differential Branch number β_d of a linear permutation layer L is defined as:

$$\beta_d(L) = \min_{\Delta X \neq 0, X_1, X_2 \in \{0,1\}^{16}} (\mathrm{Hw}(\Delta X) + \mathrm{Hw}(L(\Delta X)))$$

where $\Delta X = X_1 \oplus X_2$ is input difference to the linear layers of *FeW* and $L(\Delta X) = L(X_1) \oplus L(X_2)$ is output difference. In case of *FeW*, Differential Branch number of the linear layer L_1 and L_2 used in F is 5, which is verified by a Computer programme (Table 2).

Theorem 1: If $P_i \parallel P_{i+1}$ is the 64-bit input to i^{th} round of *FeW* and X_i is the 32 bit input and Y_i is the 32 bit output to the round function F at i^{th} round. We obtain the following relationship between the input and output of three consecutive rounds (i.e. i^{th}, $i+1^{th}$ and $i+2^{th}$ rounds).

$$X_i \oplus X_{i+2} = Y_{i+1}$$

Proof: We draw 3 rounds of *FeW* in Fig. 3, We consider structure of *FeW* broadly a Feistel structure. We have the following relations between the intermediate states, input and output to the round function:

$$X_i = P_{i+1} \quad \text{(i)}$$

$$X_{i+1} = P_{i+2} \quad \text{(ii)}$$

$$X_{i+2} = P_{i+3} \quad \text{(iii)}$$

$$Y_i = P_i \oplus P_{i+2} \quad \text{(iv)}$$

$$Y_{i+1} = P_{i+1} \oplus P_{i+3} \quad \text{(v)}$$

$$Y_{i+2} = P_{i+2} \oplus P_{i+4} \quad \text{(vi)}$$

We have the following desired relation using equations (i), (iii) & (v) between input and output to the round function W:

$$Y_{i+1} = X_i \oplus X_{i+2} \quad \text{(vii)}$$

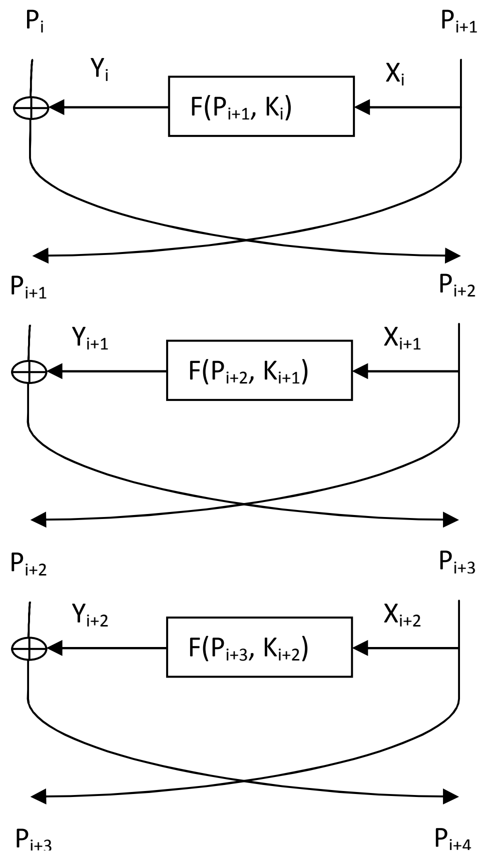

Fig. 3: Three consecutive rounds of *FeW* (Broadly Feistel structure)

Theorem 2: If $\Delta X_i \oplus \Delta X_{i+2}$ is not equal to zero, then three consecutive rounds of *FeW* have at least 5 differentially active S-boxes.

Proof: We denote the linear transformation layers (L_1 and L_2) in round function F by L and use theorem 1 with linearity of L, we have the following relation:

$$\Delta X_i \oplus \Delta X_{i+2} = \Delta Y_{i+1} = L(L^{-1}(\Delta Y_{i+1}) = L(\ \Delta L^{-1} (Y_{i+1})) \quad \text{(viii)}$$

Since applying inverse of L on the output to round function at round i, we get the same number of non zero nibbles as there are in the input to round function. Therefore, we have the following relation:

$$Hw(\Delta X_{i+1}) = Hw(\Delta L^{-1}(Y_{i+1})) \qquad \text{(ix)}$$

We know the relation $Hw(\alpha) + Hw(\beta) \geq Hw(\alpha \oplus \beta)$ between the number of non zero nibbles in two binary strings α and β [16]. Using this relation, we get:

$$Hw(\Delta X_i) + Hw(\Delta X_{i+2}) \geq Hw(\Delta X_i \oplus \Delta X_{i+2}) \qquad \text{(x)}$$

Using (viii), (ix) & (x) and $\beta_d(L)$ we have the following relation which asserts that any 3 consecutive rounds will have at least 5 differentially active S-boxes if $\Delta X_i \oplus \Delta X_{i+2} \neq 0$:

$$\begin{aligned} Hw(\Delta X_i) + Hw(\Delta X_{i+1}) + Hw(\Delta X_{i+2}) &= Hw(\Delta X_i) + Hw(\Delta X_{i+2}) + Hw(\Delta X_{i+1}) \\ &= Hw(\Delta X_i) + Hw(\Delta X_{i+2}) + Hw(\Delta L^{-1}(Y_{i+1})) \end{aligned}$$

$$\geq Hw(\Delta X_i \oplus \Delta X_{i+2} \oplus \Delta L^{-1}(Y_{i+1}))$$
$$= Hw(L(\Delta L^{-1}(Y_{i+1})) \oplus \Delta L^{-1}(Y_{i+1}))$$
$$= 5$$

Theorem 3: Any four consecutive rounds of *FeW* (i^{th} to $i+3^{rd}$ round) can have at the most one differentially inactive round function.

Proof: We use the Fig. 3 to prove the theorem. If round function to i^{th} round is differentially active then we must have 64-bit input to this round of the form $(\Delta P_i \neq 0)||(\Delta P_{i+1}=0)$, which implies that $\Delta X_i = 0$ & $\Delta Y_i = 0$. Number of active S-boxes in this round are zero. Input to $i+1^{th}$ round will be of the form $(\Delta P_{i+1} = 0)||(\Delta P_{i+2} = \Delta P_i \neq 0)$, which gives us that $\Delta X_{i+1} \neq 0$ & $\Delta Y_{i+1} \neq 0$. Minimum number of active S-boxes in this round is 1. We obtain input to $i+2^{nd}$ round of the form $(\Delta P_{i+2} = \Delta P_i \neq 0)||(\Delta P_{i+3} \neq 0)$, which means $\Delta X_{i+2} \neq 0$ & $\Delta Y_{i+2} \neq 0$. Minimum number of active S-boxes for this round are 4. For the input ot the round function of $i+3^{rd}$ round (i.e ΔX_{i+3}) to be be differentially passive, we should get input to this round of the form $(\Delta P_{i+3} = \Delta P_i \neq 0)||(\Delta P_{i+4} = 0)$. This type of input $(\Delta P_{i+4} = 0)$ is possible in the case when $\Delta P_i \oplus \Delta Y_{i+2}=0$. Susbstituting this value in terms of input, we get $\Delta P_i \oplus F(\Delta P_{i+3}) =0$. Finally, writing P_{i+3} in terms of ΔP_{i+2}, we get $\Delta P_i \oplus F(F(\Delta P_{i+2}))=0$, which can be represented in terms of input to i^{th} round as follows:

$$\Delta P_i \oplus F(F(\Delta P_i)) =0$$

We searched this relation for all possible input differences $\Delta P_i \neq 0$ using a Computer programme but this relation was not satisfied for any $\Delta P_i \neq 0$.

We derived the above results considering the strucure of *FeW* broadly a Feistel structure, while *FeW* is based on Fesitel-M structure, which mix the Feistel branches and apply two different functions. We found the minimum number of active S-boxes (Table 3) in each round of *FeW* with Feistel-M structure. Maximum differential probability of the 4x4 S-box used in *FeW* is 2^{-2}. Table 3 shows that the minimum number of active S-boxes in 11 rounds is 34, which confirms that single differential charcatersitics is bounded by 2^{-68}. It is not possible to mount any useful differential attack beyond 16 rounds.

#Round	#Active S-boxes(min)
1	0
2	1
3	5
4	10
5	14
6	17
7	20
8	24
9	27
10	30
11	34

Table 3: Number of Active S-boxes

To show full round *FeW* immune to Differential attack, we provide a lower bound on the number of active S-boxes in 27 round differential charcteristic. The following theorem shows the resistance of full round *FeW* against the Differential attack.

Theorem 4: Any differential characteristic for 27 rounds of *FeW* has a minimum 45 active S-boxes and hence the probability of this differential characteristic is 2^{-90}.

Proof: We can easily prove this using the fact that any 3 round of *FeW* has a minimum of 5 differentially active S-boxes. Therefore 3x9=27 rounds will have minimum of 5x9=45 differentially active S-boxes. So, the probability of a single 27 round differential trail is $(2^{-2})^{45}=2^{-90}$. If we use 27 round trail to recover round subkeys for 32 round *FeW*, it will require 2^{90} chosen plaintext which is more than the amount of data available. This theorem ensures that full round *FeW* is secure enough against differential attack.

3.2 Impossible Differential Cryptanalysis

Impossbile Differential attack [3] is an extension of basic differential attack This attack works with differentials of probability 0 as opposed to basic differential attack which requires differentials of high probability. This attack recovers keys using impossible differential by dropping the keys from the list of all possible key candidates which satisfies the impossible differntial and the key (or keys) remaning in the list are the candidates for the correct key. This attack has given best result on some ciphers like CLEFIA. We obtain the following best 6 round impossible differential trail for *FeW*:

$$(\alpha000\ 0000\ 0000\ 0000)\ 6R\rightarrow (0000\ 0000\ 0000\ \alpha000)$$

where α denote any non zero 4 bit nibble and * denote any 4 bit nibble. We get contradication at round 3 between two events of probability 1 (Table 4). Using this impossible differential trail and allowing the attacker to add 3 round on the top and 6 rounds at the bottom of this trail , on can still break *FeW* reduced to at the most 15 rounds.

#R	6 R Impossible Differential	Pr
0	α000 0000 0000 0000	
1	0000 0000 α000 0000	1
2	α000 0000 **** 0000	1
3	**** 0000 **** ****	1
3	**** **** **** 0000	1
4	**** 0000 000α 0000	1
5	000α 0000 0000 0000	1
6	0000 0000 0000 000α	

Table 4: Impossible Differential trail

3.3 Linear Cryptanalysis

FeW can be shown resistant to Linear cryptanalysis [24] similar to the case of resistence to differential attack. First we define Linear Branch number of the linear layer L of *FeW*:

Definition 3: Linear Branch number β_l of a function L is defined as:

$$\beta_l(L) = \min_{\alpha \neq 0, \alpha \in \{0,1\}^{16}} (Hw(\alpha) + Hw(L^*(\alpha))$$

where α is output mask value and $L^*(\alpha)$ is input mask value to the linear layer . L^* is the linear function concerned to L. In case of *FeW*, Linear Branch number of the linear layers L_1 and L_2 used in F is 5, which is verified by a Computer programme.

Theorem 4: Any linear characteristic for 27 rounds of *FeW* has a minimum 45 active S-boxes and hence the maximal bias of this 27 round linear trail is 2^{-90}.

Proof: We again consider *FeW* as a Feistel structure only. Linear branch number of linear layers L_1 and L_2 of *FeW* is 5 and the maximal bias of the S-box is 2^{-2}. Any 3 rounds linear trail of *FeW* has a minimum 5 linearly active S-boxes which can be easily verified (Fig. 3). Any output mask value α to round function F corresponds to input mask value β and this becomes output mask value to the next round. If we get α as input mask value again then there are minimum 5 linearly active S-boxes in 3 round linear trail since the branch number of Layer L is 5.

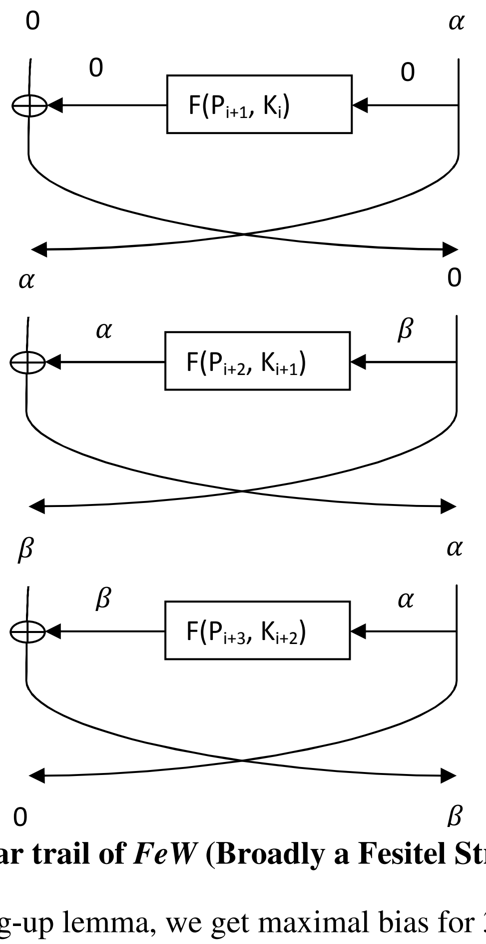

Fig. 4: Linear trail of *FeW* (Broadly a Fesitel Structure)

By using Matsui`s [24] Piling-up lemma, we get maximal bias for 3 round linear trail:

$$2^4 x (2^{-2})^5 = 2^{-6}$$

Applying the same lemma again, we get the maximal bias of 27 round linear trail:

$$2^8 x (2^{-6})^9 = 2^{-46}$$

If we assume that full round is attacked using 27 round differential trail, then the amount of known plaintext/ciphertext data requirement is of order 2^{90} which is more than the available data limit.

3.4 Zero Correlation Cryptanalysis

Zero correlation attack [5] is an extension of linear attack and this is similar to the impossible differential attack which is the extension of differential attack. This attack was published by Bogdanov & Rijmen and they applied this attack on CLEFIA reduced to 13 rounds using 9 round zero correlation trail. CLEFIA like structures have the maximum 9 round zero correlation trail. But, our design is Feistel-M based design so it is not possible to get even 9 round zero correlation trail for this type of designs. Similar to the impossible differential cryptanalysis of *FeW*, zero correlation trail for *FeW* exist ony for 6 rounds and *FeW* can be attacked using this attack upto the maximum 15 rounds.

3.5 Related Key Cryptanalysis

We exploit the weakness of Key schedule in Related key attack[1]. We have designed the Key schedule of *FeW* in a similar way to the lightweight block cipher PRESENT`s Key schedule. Our key schedule is stronger than the Key schedule of PRESENT. We have made 3 application of non linear S-box for each 16-bit subkey derivation. As a result, all subkey bits are nonlinear function of key bits after 11 rounds. Till now, there is no significant attack on PRESENT`s key schedule, therefore we assume that this attack can not be applied to *FeW* which has a stronger key schedule than PRESENT.

3. Conclusion

We described a new lightweight block cipher *FeW* by introducing a new type of mixing between Feistel and generalised Feistel structures based designs. We called this structure a Feistel-M structure which is used for the first time in the design of lightweight block cipher *FeW*. We analysed the security of *FeW* against some basic cryptanalytic attacks and this design is secure enough against these attacks. There is a large variety of attacks which can be applied to block ciphers, therefore we invite all researchers to apply and report their attacks on *FeW*.

Acknowledgements

The first author would like to thank Dr. PK Saxena, Director SAG for his support and Ms. Pratibha Yadav for her valuable guidance and continuous support to work in this direction. I am thankful to Dr. Dhananjoy Dey for giving his precious time in valuable discussions on design principles & cryptanalysis and Ms. Noopur Shrotriya for her valuable guidance in computer implementations. I thank Mr. Sohan Lal and Ms. Anita Dwivedi for help in testing the data of Cipher. I also thank Dr. PR Mishra for spending his valuable time in reading the paper and suggesting the useful and important corrections in the paper.

References

1. Biham, E.: New Types of Cryptanalytic Attacks Using Related Keys. EUROCRYPT 93, LNCS vol. 765, pp. 398-409, Springer-Verlag, 1994
2. Bogdanov, A.: Analysis and Design of Block Cipher Constructions. Ph.D thesis 2009
3. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. EUROCRYPT 1999, LNCS, vol. 3027, pp. 12-23, Springer-Verlag, 1999
4. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. CHES 2007, LNCS, vol. 4727, pp. 450-466, Springer, 2007
5. Bogdanov, A., Rijmen, V.: Linear Hulls with correlation Zero and Linear Cryptanalysis of Block Ciphers. Cryptology ePrint Archive, Report 2011/123, http://eprint.iacr.org
6. Biham, E., Shamir, A.: Differential Cryptanalysis of DES-like Cryptosystems. Journal of Cryptology, vol. 4, no. 1, pp. 372, 1991
7. Beaulieu, R., Shors, D., Smith, J., Clark, S.T., Weeks, B., Wingers, L.: The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, http://eprint.iacr.org
8. Cannire, C., Dunkelman, O., Knezevi, M.: Katan and Ktantana family of small and efficient hardware-oriented block ciphers", CHES 2009, LNCS, vol. 5747, pp. 272-288, 2009
9. Diffie, W., Ledin, G. (translators): SMS4 Encryption Algorithm for Wireless Networks. Cryptology ePrint Archive, Report 2008/329, http://eprint.iacr.org
10. Daemen, J., Rijmen, V.: The Design of Rijndael. Berlin: Springer-Verlag (2002)
11. Engels, D., Saarinen, M.-J. O., Schweitzer, P., Smith, E. M.: The Hummingbird-2 Lightweight Authenticated Encryption Algorithm. RFID Sec 2011, The 7th Workshop on RFID Security and Privacy, 26–28 June 2011, Amherst, Massachusetts, USA, 2011
12. Gong, Z., Nikova, S., Law, Y. W.: KLEIN: A New Family of Lightweight Block Ciphers. RFID Sec 2011, LNCS vol. 7055, pp. 1-18, 2011
13. Guo, J., Peyrin, T., Poschmann, A.: The LED Block Cipher. Cryptographic Hardware and Embedded Systems - CHES 2011, LNCS, 2011
14. Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., Chee, S., "HIGHT: A New Block Cipher Suitable for Low-Resource Device" CHES 2006, LNCS, vol. 4249, pp. 46-59, 2006
15. Izadi, M., Sadeghiyan, B., Sadeghian, S., Khanooki, H., "MIBS: A New LightweightBlock Cipher" CANS 2009, LNCS, vol. 5888, pp. 334-348, 2009
16. Kanda, M.: Practical Security Evaluation against Differential and Linear Cryptanalysis for Feistel Ciphers with SPN Round Function Kanda. SAC 2000, LNCS 2012, pp. 324-338, Springer-Verlag 2001
17. Kim, J., Hong, S., Sung, J., Lee, C., Lee, S., "Impossible differential cryptanalysis for block cipher structure. In: Johansson" INDOCRYPT 2003, LNCS, vol. 2904, pp. 82-96, 2003

18. Kim, T., Kim, J., Hong, S., Sung, J.: Linear and Differential Cryptanalysis of Reduced SMS4 Block Cipher. Cryptology ePrint Archive, Report 2008/281, http://eprint.iacr.org
19. Knudsen, L.R., Leander, G., Poschmann, A., Robshaw, M.J.B.: PRINTcipher: A Block Cipher for IC Printing. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010, LNCS, vol. 6225, pp. 16-32, 2010
20. Kumar, M., Pal, S.K., Yadav, P.: Mathematical Constructs of Lightweight Block Ciphers-A Survey. American Jr. of Mathematics and Sciences, Vol. 2, no. 1, January, 2013
21. Knudsen, L., Robshaw, MJB: Block cipher companion. Book Springer
22. Lim, C., Korkishko, T.: mCrypton - A Lightweight Block Cipher for Security of Low-cost RFID Tags and Sensors. WISA 2005. LNCS, vol. 3786, pp. 243-258, 2006
23. Leander, G., Paar, C., Poschmann, A.: New Lightweight DES Variants. FSE 2007, LNCS, vol. 4593, pp. 196-210, 2007
24. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. Advances in Cryptology EUROCRYPT 1993, LNCS 765, pp. 386-397, Springer-Verlag, 1994
25. Nyberg, K: Perfect nonlinear S-boxes. Eurocrypt 1991, LNCS 547, 1991
26. Poschmann, A.Y.: LIGHTWEIGHT CRYPTOGRAPHY Cryptographic Engineering for a Pervasive World. Ph.D thesis 2009
27. Rijmen, V.: Cryptanalysis and Design of iterated Block Cipher. Ph.D. Thesis 1997
28. Sorkin, A.: LUCIFER: a cryptographic algorithm. *Cryptologia*, **8**(1), 22–35, 1984
29. Shannon, C. E.: Communication Theory of Secrecy Systems. Bell Systems Technical Journal, pp. 656-715, 1949
30. Saarinen, M.O.: Cryptographic Analysis of all 4x4 bit S-boxes. Cryptology ePrint Archive, Report 2011/218, http://eprint.iacr.org
31. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.:Piccolo: An Ultra-Lightweight Blockcipher. CHES 2011, LNCS, vol. 6917, pp. 342-357, 2011
32. Schneier, B., Kelsey, J.: Unbalanced Feistel Networks and Block-Cipher Design. FSE 1996, pp. 121-144, 1996
33. Suzaki, T., Minematsu, K, "Improving the Generalized Feistel" FSE 2010, LNCS, vol. 6147, pp. 19-39, 2010
34. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: Twine: A Lightweight, Versatile Blockcipher. ECRYPT Workshop on Lightweight Cryptography (2011), http://www.uclouvain.be/crypto/ecrypt lc11/static/post proceedings.pdf. 2011
35. Standaert, F.-X., Piret, G., Gershenfeld, N., Quisquater, J.-J. "SEA: A Scalable Encryption Algorithm for Small Embedded Applications" CARDIS 2006. LNCS, vol. 3928, pp. 222-236, 2006
36. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-bit Block cipher CLEFIA (Extended Abstract), FSE 2007, LNCS, vol. 4593, pp. 181-195, 2007
37. Su, B., Wu, W., Zhang, W.: Differential Cryptanalysis of SMS4 Block Cipher. Cryptology ePrint Archive, Report 2010/62, http://eprint.iacr.org

38. Wu, W., Zhang, L.: LBlock: Lightweight Block Cipher. Cryptology ePrint Archive, Report 2011/345, http://eprint.iacr.org
39. Wheeler, D., Needham, R. "TEA, a Tiny Encryption Algorithm" FSE 1994, LNCS, vol. 1008, pp. 363-366, 1995
40. Yap, H., Khoo, K., Poschmann, A., Henricksen, M.: EPCBC - A Block Cipher for Electronic Product Code Encryption. CANS 2011, LNCS, 2011

Appendix A: Test vectors (*FeW*-80)

plaintext	key	ciphertext
00000000 00000000	00000000 00000000 0000	**70954e26 8a5b327b**
00000000 00000000	FFFFFFFF FFFFFFFF FFFF	**45381557 e3c84bdd**
FFFFFFFF FFFFFFFF	00000000 00000000 0000	**a308ea91 57a81d66**
FFFFFFFF FFFFFFFF	FFFFFFFF FFFFFFFF FFFF	**b5c4b383 48c989e8**

Appendix B: Key schedule (MK-128)

First, store MK-128 in a key register called MK as MK= k_0 k_1 k_2 k_3.........k_{127}. We obtain round subkeys RK_0 by extracting leftmost 16 bits of current contents of MK and proceed in the following way to obtain the other round subkeys:

a. While *i* < 64, update the register MK in the following steps:
   (i) MK<<<13
   (ii) $[k_0\ k_1\ k_2\ k_3] \leftarrow S[k_0\ k_1\ k_2\ k_3]$
   $[k_4\ k_5\ k_6\ k_7] \leftarrow S[k_4\ k_5\ k_6\ k_7]$
   $[k_{64}\ k_{65}\ k_{66}\ k_{67}] \leftarrow S[k_{64}\ k_{65}\ k_{66}\ k_{67}]$
   $[k_{76}\ k_{77}\ k_{78}\ k_{79}] \leftarrow S[k_{76}\ k_{77}\ k_{78}\ k_{79}]$
   (iii) $[k_{68}\ k_{69}\ k_{70}\ k_{71}\ k_{72}\ k_{73}\ k_{74}\ k_{75}] \leftarrow [k_{68}\ k_{69}\ k_{70}\ k_{71}\ k_{72}\ k_{73}\ k_{74}\ k_{75}] \oplus [i]_2$

b. Increment *i* by 1 and extract leftmost 16 bits of current contents of MK as round subkey RK_i.

Appendix C: Decryption Algorithm (FeW-80)

FeW is balance Feistel-M based design, therefore decryption algorithm of *FeW* uses the same round function as encryption algorithm. The only difference is that the round subkey is used in reverse direction. Decryption procedure is described as follows:

First we divide 64-bit ciphertext C_m into two halves of size 32 bit each namely C_0 and C_1. We have 64-bit input ciphertext C_m as below:

$$C_0 \| C_1 \leftarrow C_m$$

Decryption procedure of FeW is expressed as follows:

(a) For i =0 to 31, apply round functions F on 32-bit word C_{i+1} and xor it with C_i to produce C_{i+2}

$$C_{i+2} \leftarrow C_i \oplus F(C_{i+1} \oplus K_{31-i})$$

(b) Finally, apply the following swap function on the output of last round

$$(C_{33}, C_{32}) \leftarrow (C_{32}, C_{33})$$

(c) We obtain the following 32 bit plaintext words P_0 and P_1:

$$(P_0, P_1) \leftarrow (C_{33}, C_{32})$$

We obtain the 64-bit plaintext as concatenation of two 32 bit words P_0 and P_1 as follows:

$$P_m \leftarrow P_0 \| P_1$$

Appendix D: Difference & Linear Distribution Tables

OD: output difference, ID: Input difference, OM: output mask, IM: input mask

OD→ ID↓	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	2	2	0	2	2	2
2	0	0	0	2	0	2	0	0	0	0	0	2	4	2	4	0
3	0	2	2	2	2	2	2	0	0	2	0	0	0	0	2	0
4	0	0	0	2	0	2	4	0	0	2	4	0	0	0	0	2
5	0	0	2	0	4	2	0	0	0	2	0	0	2	2	0	2
6	0	0	0	2	2	0	2	2	2	2	2	0	0	2	0	0
7	0	2	0	2	0	0	0	4	2	2	0	0	2	0	0	2
8	0	0	0	2	0	0	2	4	0	0	0	2	2	2	0	2
9	0	0	0	0	0	0	2	2	4	2	2	0	2	0	2	0
A	0	2	2	0	0	0	0	0	4	0	0	4	0	2	2	0
B	0	0	4	0	2	4	2	0	0	2	0	2	0	0	0	0
C	0	2	2	0	0	4	0	0	2	0	2	0	2	2	0	0
D	0	2	4	2	0	0	0	0	2	0	2	0	0	0	0	4
E	0	4	0	0	2	0	0	2	0	0	0	4	0	2	2	0
F	0	2	0	0	4	0	2	0	0	0	2	0	2	0	2	2

Table 4: Difference Distribution Table of S-box

OM→ IM↓	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	2	2	2	-2	0	-4	0	0	2	2	-2	2	4	0
2	0	-2	0	2	-2	-4	-2	0	0	2	0	-2	-2	0	-2	4
3	0	2	2	0	0	-2	-2	0	-4	2	-2	0	0	2	-2	-4
4	0	2	0	2	-2	0	2	4	0	-2	0	-2	-2	4	2	0
5	0	-2	-2	4	0	2	-2	0	0	2	-2	0	4	2	2	0
6	0	0	0	-4	0	0	-4	0	0	0	0	-4	0	0	-4	0
7	0	0	-2	2	2	-2	0	0	-4	-4	2	-2	2	-2	0	0
8	0	0	2	-2	0	0	-2	2	-2	-2	0	4	2	2	0	4
9	0	4	0	0	2	2	2	-2	-2	2	-2	-2	0	0	0	4
A	0	-2	2	0	2	0	0	-2	2	-4	-4	-2	0	2	-2	0
B	0	-2	0	-2	-4	-2	4	-2	-2	0	-2	0	2	0	2	0
C	0	2	2	0	-2	0	0	-2	2	0	4	-2	4	2	-2	0
D	0	2	4	2	0	-2	0	2	2	0	-2	0	2	-4	2	0
E	0	0	2	2	-4	4	-2	-2	-2	-2	0	0	-2	-2	0	0
F	0	4	-4	0	-2	-2	-2	-2	2	-2	-2	2	0	0	0	0

Table 5: Linear approximation Table of S-box