NUX: An Ultra lightweight Cipher design for security in pervasive computing

GAURAV BANSOD, SWAPNIL SUTAR, ABHIJIT PATIL, NARAYAN PISHAROTY
Electronics and Telecommunication
[1]Pune Institute of Computer Technology, [2,3,4]Symbiosis Institute of Technology, [4]Glocal University
Lavale, Pune, 412115, Maharashtra,
INDIA +((91) 2039116300)
gaurav249@gmail.com,swapnil.sutar@sitpune.edu.in, abhijit.patil@sitpune.edu.in,
narayan.pisharoty@gmail.com

Abstract: **- This paper proposes an ultra-lightweight cipher NUX. NUX is a generalized Feistel network. It supports 128/80 bit key length and 64 bit plaintext. For 128 bit key length, NUX needs only 1022 GEs which is very less as compared to all existing ciphers. NUX is the small-scale cipher design till date. NUX design results into less footprint and minimal memory size. This paper presents security analysis of NUX cipher design which shows cipher's resistance against basic attacks like Linear and Differential cryptanalysis. Advance attacks like Biclique attack implemented for NUX cipher design. Two different F function in NUX cipher design results in high diffusion mechanism which generates large number of active S-boxes. NUX cipher has total 31 rounds. NUX design will be best suited for critical application like smart grid, IoT, wireless sensor network, where memory size, footprint area and power are the main constraints.**

***Index Terms*— Lightweight cryptography, Feistel cipher, Block cipher, IoT, Encryption, Embedded security, ubiquitous computing**

I. INTRODUCTION

Lightweight cipher designs are very popular for IoT, RFID, WLAN (Wireless Local Area Network), and WBAN (Wireless Body Area Network) application. Such application uses pervasive devices, which results in ubiquitous computing. In ubiquitous computing, to avoid tapping of sensitive data security is essential. RFID used 10000 GEs for its design. To achieve security in low resourced device GEs should be lower than 2000-2200. AES and DES are the encryption standards which have GEs around 2400-3500. This led to the emergence of a new field called lightweight cryptography. Many existing ciphers designed with minimal memory size. Most of existing ciphers are block ciphers which allow high diffusion mechanism [1]. There are two types of block ciphers, Feistel network and SP network. PRESENT [2], RECTANGLE [3], LED [4] are the SP network based block cipher. CLEFIA [5], L-Block [6], XTEA [7] are the Feistel based block ciphers. Feistel network divided into classical Feistel and generalized Feistel. This paper proposes the NUX cipher, which uses generalized Feistel where block size is divided into four equal parts.

NUX cipher design is generalized Feistel design which has 31 rounds that will provide ample of security to resist against all possible types of attacks. GEs (Gate Equivalents) are considered to be the most dominant factor in designing lightweight ciphers. PRESENT, SIMON [8], SPECK [8], and RECTANGLE ciphers have lesser GEs (Gate Equivalent). RECTANGLE cipher has robust cryptanalytic properties. SIMON and SPECK are identified as ciphers which have good hardware and software performance. PRESENT has less footprint area and higher throughput. PRESENT cipher has strong permutation layer and it is good alternative to AES-128 for constrained devices.

This paper suggests an ultra-lightweight cipher which needs less GEs and minimal memory size. NUX cipher has robust 16 bit permutation layer, which helps to resist all possible types of attacks.

For NUX cipher we have used the following notations

PT	64-bit input plaintext block
CT	64-bit output cipher text block
RKi	128-bit Round sub key for round i
F	Function
$\oplus$	Bitwise exclusive-OR operation
<<<n	Left cyclic shift by n bits
>>>n	Right cyclic shift by n bits
RC_i	Round counter i
||	Concatenation of two strings
!	Bitwise NOT operation
64 bits	Maximum length of plain text

II. THE BLOCK CIPHER NUX

NUX cipher is based on generalized Feistel network [9] and it has 31 rounds. It takes 64 bit plaintext and support 128/80 bits key length. Fig.1 shows round function of NUX cipher.

The 64 bit plaintext is divided into 4 different blocks of 16 bit data, $P_i^4[63:48]$, $P_i^3[47:32]$, $P_i^2[31:16]$, $P_i^1[15:0]$. As shown in Fig.1, there are two F-functions. F1 function has four identical 4x4 S-box and left circular shift operator by 8 similarly F2 function has four identical 4x4 S-box and the right circular

shift operator by 3. P_i^2 is fed to F1 function and output of F1 function is XOR-ed with the right circular shifted P_i^1 data by 3 and the result is XOR-ed with 16 bit extracted key RK_i' [31:16] from 128 bit key scheduling algorithm. The P_i^3 is fed to F2 function and output of F2 function is XOR-ed with left circular shifted P_i^4 data by 8 and the result is XOR-ed with 16 bit extracted key RK [15:0] from same key scheduling algorithm. Block permutation layer shuffles these 16-bit blocks of data as shown in Fig.1 and is fed to the 16 bit permutation. This design uses a Robust S-box with 128 bit key scheduling algorithm. 128 bit key scheduling updates the key at every round.

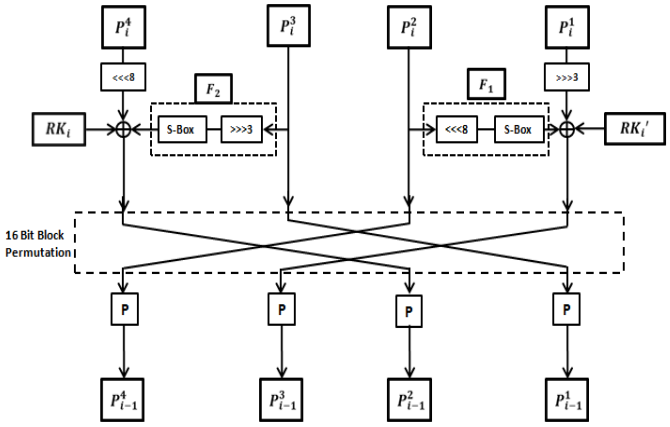

Fig.1: Block diagram of NUX cipher

1. Design rationale and goals:

Lightweight block ciphers are designed for constraint environment. It is important to note that we are not only designing a block cipher which provides security but also designing a cipher which is suitable for low resourced and constraint environment. Such cipher should have to follow conventions which are listed as follow:

- The cipher should have minimum number of GEs (less than 1500 for lightweight) so that it is feasible to implement for low resourced and constraint environment for better hardware performance.

- The cipher should need minimum bytes of Flash and RAM memory in term of code size as 8 bit microcontrollers has very limited memory space.

- To decrease the gate count i.e. GEs, we have minimized the use of logic gates in round function of NUX cipher and maximize the use circular shift operator, bit permutation, & block permutation because they needs only wires. Again serial ASIC implementation shares single S-box between data and key layer so the GEs required for another 15 S-boxes will be reduced. This strategy has helped us to design cipher in less than 1100GEs. In serial ASIC implementation we are giving 4 inputs at a given time.

- Cipher uses only 16 bit permutation i.e. only 16 bit will be shuffled according to bit permutation Table 2. To declare 16 bit permutation we need only 16 bytes of array which needs only 16 bytes of memory. 64 bit plaintext and 128 bit key occupies 24 bytes of memory. We have reduced the use global and local variables in code to achieve less memory consumption with competitive throughput. This results in lesser execution time also.

- We have adopted the 128/80 bit key encryption for NUX cipher to boost security level. 80 bit key encryption is enough to provide security to applications like IoT [2].

- Cipher should resist basic attacks like linear and differential attacks. To resist this attack, it is important to maximize the minimum number of active S-box in minimum number of rounds. The S-box which have non zero input or non zero output known as “active S-box”. It is necessary to have high diffusion and confusion property in cipher design to achieve the required count for active S-box.

- A suitable choice of S-box makes the cipher design robust. The S-box which we have used is compact in its hardware implementation which needs around 24 GEs. Moreover because of this S- box, NUX cipher gives maximum number of active S-boxes in minimum number of rounds which thwarts linear and differential attack.

- Every bits in bit permutation is distributed in such a way that that cipher design result in good Avalanche effect i.e. change in single bit result in change in more than 50% more bit for given block length..

Keeping such consideration in mind, we have decided to make NUX cipher with 64 bit plaintext and 80/128 bit key scheduling algorithm. The encryption and decryption of NUX cipher result in lightweight cipher design. We have tested software performance on platform of ARM7 LPC2129.. This paper presents a NUX cipher design and its resistance against basic attack as well as advance attacks. As per our design rationale, NUX will be best suited for application like IoT (internet of things) and Wireless Sensor Networks (WSN) and provides enough security with its lightweight design implementation.

2. Encryption flow:

Block of 64 bit is divided into 4 equal parts of 16 bit plaintext P_0^4, P_0^3, P_0^2 and P_0^1.

$$PT \leftarrow P_0^4 \,||\, P_0^3 \,||\, P_0^2 \,||\, P_0^1$$

a) Apply F1 function on P_0^2, left circular shift (LCS) operator on P_0^1 and XORed it with the round key RK_i'.

$$P1 \leftarrow F1\,(P_0^2) \oplus LCS\,(P_0^1, 8) \oplus RK_i'$$

b) P2 ← P_0^2
c) P3 ← P_0^3

d) Apply F2 function on P_0^3, right circular shift (RCS) operator on P_0^4 and XORed it with the round key RK_i.

$$P4 \leftarrow F2\ (P_0^3) \oplus RCS\ (P_0^4, 3) \oplus RK_i$$

e) Shuffled blocks are P1, P2, P3, and P4.

$$B1 \leftarrow P3,\ B2 \leftarrow P4,\ B3 \leftarrow P1,\ B4 \leftarrow P2$$

f) Apply permutation layer on B1, B2, B3, and B4.

$$P_{i+1}^4 \leftarrow P\ [B4]$$
$$P_{i+1}^3 \leftarrow P\ [B3]$$
$$P_{i+1}^2 \leftarrow P\ [B2]$$
$$P_{i+1}^1 \leftarrow P\ [B1]$$

After 31 round we will get 64 bit cipher text as

$$CT \leftarrow P_{31}^4 \,||\, P_{31}^3 \,||\, P_{31}^2 \,||\, P_{31}^1$$

3. F-Function -

NUX cipher has two F-functions. In F1 function, 16 bits of $P_2^i[31:16]$ is left circular shifted by 8. The result of this circular shift is fed to the four identical 4x4 S-box. In F2 function, 16 bits of $P_3^i[47:32]$ is right circular shifted by 3 and fed to the four identical 4x4 S-box. Both functions F1 & F2 have adopted same S-box as shown in Table 1 and the operation of these two functions clearly depicted in Fig. 1. These both F-functions uses only circular shifting and S-box where S-box required GEs but circular shifting needs only wires hence it helps us to reduce the number of GEs which is the important metric in lightweight block cipher.

4. S-box –

The S-box used in NUX cipher design is of 4-bit to 4-bit S-box S: $F_2^4 \rightarrow F_2^4$. 4x4 S-box of NUX results in compact hardware implementation which is around 24 GEs. TABLE 1 represents the hexadecimal values for the Substitution layer. This is a strongest S-box as it holds good CAR_{lc} and CAR_{dc}. CAR_{lc} is number of non zero elements in linear approximation table for given S-box at position where hamming weight of input and output mask is equals to '1'. Similarly, CAR_{dc} is the number of non zero elements in difference distribution table for given S-box at position where hamming weight of input and output difference is equal to '1'. If CAR_{lc}= CAR_{dc}=0, then it is strongest S-box but it is practically impossible. In this cipher design we have achieved CAR_{lc} & CAR_{dc} as minimum as possible. For the S-box represents in Table 1, CAR_{lc}=2 and CAR_{dc}=2

TABLE 1
S-BOX OF NUX CIPHER

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

5. Bit permutation -

In NUX cipher, 16 bits P_i^n (where n = 1, 2, 3, 4) are permuted according to designed P-layer. The permutation is of all 16 bits depicted in TABLE 2. The bit permutation will increase the minimum number of active S-box as it operates along the circular shift operator. Criterions used for designing a Permutation layer are as follows and is mentioned in paper [10]:

1. At round r, the output of S-box is distributed in such a way that two of them affect the middle bits of S-box at round r+1 and other two affects the end bits.
2. Four outputs from each S-box affects the four different S-boxes.

TABLE 2
BIT PERMUTATION TABLE FOR NUX

i	00	01	02	03	04	05	06	07
P[i]	15	11	07	03	02	14	10	06
i	08	09	10	11	12	13	14	15
P[i]	05	01	13	09	08	04	00	12

Bit permutation is 16 bits only so memory requirement for this operation is less. PRESENT utilizes 64 bit permutation hence it need more memory as compared to NUX cipher. In NUX cipher design, 16 bit block permutation layer is shown in Fig. 1 which helps the design to increase confusion property and results in high diffusion which gives good avalanche effect. Due to this design decisions, NUX cipher results in best hardware and software performance.

Encryption Algorithm

Input-
Plaintext: $A_{64} \rightarrow a^{63}\, a^{62}\, a^{61}\, a^{60} \ldots a^3\, a^2\, a^1 a^0$, S[16],BP[32],

Output-
Ciphertext: C_{64}

For i = 0 to 31 do
$P_i^4 \rightarrow a^{63}\, a^{62}\, a^{61}\, a^{60} \ldots a^{51}\, a^{50}\, a^{49} a^{48}$
$P_i^3 \rightarrow a^{47}\, a^{46}\, a^{45}\, a^{44} \ldots a^{35}\, a^{34}\, a^{33} a^{32}$
$P_i^2 \rightarrow a^{31}\, a^{30}\, a^{29}\, a^{28} \ldots a^{19}\, a^{18}\, a^{17} a^{16}$
$P_i^1 \rightarrow a^{15}\, a^{14}\, a^{13}\, a^{12} \ldots a^{03}\, a^{02}\, a^{01} a^{00}$

$Pt1 \rightarrow RCS\ (P_i^1, 3) \oplus$ S-box $[LCS\ (P_i^2, 8)] \oplus RK_i'$
$Pt2 \rightarrow LCS\ (P_i^4, 8) \oplus$ S-box $[RCS\ (P_i^3, 3)] \oplus RK_i$

$P_i^1 = P\ [P_i^3]$
$P_i^3 = P\ [Pt1]$
$P_i^4 = P\ [P_i^2]$
$P_i^2 = P\ [Pt4]$

$A_{64} \rightarrow P_i^4 \,||\, P_i^3 \,||\, P_i^2 \,||\, P_i^1$
i = i+1
End
$C_{64} \rightarrow A_{64} \rightarrow P_{31}^4 \,||\, P_{31}^3 \,||\, P_{31}^2 \,||\, P_{31}^1$

6. Key Schedule of 80-bit and 128-bit key length -

The NUX cipher's key scheduling is motivated from the PRESENT cipher key scheduling design. No attacks till date are reported on the PRESENT cipher key scheduling. In the NUX cipher, key scheduling algorithm generates total of 31 sub-keys each of size 32 bit.

128-bit key scheduling

A user defined 128-bit key is stored in the register KEY, 64-bit LSB's from KEY register is extracted as follows
$K^i = K_{31}\ K_{30} \ldots K_2\ K_1\ K_0$
$KEY = K_{127}\ K_{126}\ K_{125} \ldots K_2\ K_1\ K_0$
After extracting key of 64-bits, register KEY is updated in the following manner
1. KEY <<< 13.
2. $[K_3\ K_2\ K_1\ K_0] \leftarrow S\ [K_3\ K_2\ K_1\ K_0]$
3. $[K_7\ K_6\ K_5\ K_4] \leftarrow S\ [K_7\ K_6\ K_5\ K_4]$
4. $[K_{63}\ K_{62}\ K_{61}\ K_{60}\ K_{59}] \leftarrow [K_{63}\ K_{62}\ K_{61}\ K_{60}\ K_{59}] \oplus RC^i$

For 0 to 24 rounds, 5-bits of round counter i is XOR-ed with the 5-bits of key register KEY i.e. from $K_{59}$to K_{63}.

80 bit Key scheduling

A user defined 80-bit key is stored in key register KEY and LSB bits from it are used as round subkeys.
$K^i = K_{31}\ K_{30} \ldots K_2\ K_1\ K_0$
$KEY = K_{79}\ K_{78}\ K_{77} \ldots K_2\ K_1\ K_0$
After extracting 64-bit key, register KEY is updated as follows
1. KEY<<< 13.
2. $[K_3\ K_2\ K_1\ K_0] \leftarrow S\ [K_3\ K_2\ K_1\ K_0]$.
3. $[K_{63}\ K_{62}\ K_{61}\ K_{60}\ K_{59}] \leftarrow [K_{63}\ K_{62}\ K_{61}\ K_{60}\ K_{59}] \oplus RC^i$

III. Security Analysis of NUX

Cryptanalysis is way to know useful information about the key either from cipher text or plaintext. This information served us the strength of designed lightweight cipher. This paper focuses on the basic cryptanalysis technique like linear and differential cryptanalysis, as well as advance attack such as Biclique attack. In cipher design, selection of S-box is crucial as it makes the design robust. S-box is the only non linear element use in the NUX cipher. By using principle of pilling up lemmas in NUX, minimal numbers of active S-boxes are calculated.

A. Design criteria of the S-box

Robust cipher design is result of good substitution layer. Compactness and resistance against linear and differential attack are the two parameters which are considered in NUX cipher design while selecting a S-box. NUX cipher uses 4x4 S-box.

A complete design criterion of the NUX's S-box is as follows:

1. For any nonzero input and output differences $\Delta A, \Delta B \in F_2^4$ respectively we have

   $DC(\Delta A, \Delta B) = \#\{A \in F_2^4 \mid S(x) \oplus S(x \oplus \Delta A) = \Delta B\} \leq 4$, where DC is differential cardinality. Differential cardinality is number of counted non zero values in Difference Distribution Table at location where hamming weight input and output difference is one.

2. For any nonzero input and output differences $\Delta A, \Delta B \in F_2^4$ respectively we have such that hamming weight represented by $Hw(\Delta A) = Hw(\Delta B) = 1$, where Hw(x) denotes the Hamming weight of x, we have

   $Set_{DC} = DC(\Delta A, \Delta B) = \#\{A \in F_2^4 \mid S(x) \oplus S(x \oplus \Delta A) = \Delta B\} = 0$

3. Cardinality of Set_{DC} can be given as Car_{DC}, we have $Car_{DC} = 2$.

4. For any nonzero input sum and output sum such that $A, B \in F_2^4$ so we have LC(A, B)

   $LC(A, B) = \#\{x \in F_2^4 \mid A \bullet x = B \bullet S(x)\} - 8| \leq 4$
   LC is linear cardinality. LC is a number of non zero values in Linear Approximation Table at the location where hamming weight of input and output mask is one.

5. For any nonzero input sum and output sum such that $A, B \in F_2^4$, such that $Hw(a) = Hw(b) = 1$, we have

   $Set_{LC} = LC(A, B) = \#\{x \in F_2^4 \mid A \bullet x = B \bullet S(x)\} - 8| \neq 0$

6. Cardinality of Set_{LC} can be given as Car_{LC}, we have $Car_{LC} = 2$.

7. Bijective i.e. $S(a) \neq S(b)$ for all values of $a \neq b$.

8. No static point i.e. $S(a) \neq a$ for all values of $a \in F_2^4$.

B. Linear cryptanalysis

Linear cryptanalysis [11] is powerful and one of the significant attack. It is also known as known plaintext attack. It uses the high probability occurrences of linear expression containing plaintext bits, cipher text bits and sub key bits. This expression is used for mounting a linear attack on a cipher. To mount a linear attack, the attacker needs to have the knowledge about a subset of the plaintext and its corresponding cipher text. The attacker will be able to find out trails and minimum number of active S-box with help of LAT (Linear Approximation Table). If P_L is the linear probability then its bias can be given as $|P_L - 1/2|$, bias (ε) for the NUX cipher S-box is 2^{-2}. Matsui`s Piling-up lemma [11] is used to

calculate probability bias for 'n' rounds. It is essential that lightweight ciphers should resist such attack.

The best way to resist against linear cryptanalysis is
1] Optimizing the bias in the LAT. For ideal S-box bias, values should be 1/8 which is practically not possible to achieve.
2] Increase the number of active S-boxes in the cipher structure.
TABLE 3 represents the minimum number of active S-boxes from differential trails

TABLE 3
MINIMUM NUMBER OF ACTIVE S-BOX FROM LINEAR TRAIL

#Round	# Min. active S-boxes
1	0
2	1
3	4
4	9
5	13

Theorem1:
For 25 rounds of NUX, it has total 65 active S-boxes and the total bias for 25 rounds is 2^{-73}.

Proof:
It was found that for 5 rounds NUX has minimum 13 active S-boxes. Maximum bias for the NUX cipher S-box is 2^{-2} by using Matsui`s Pilling up Lemma for 5 rounds of NUX cipher the total bias can be given as,

$$2^{12} \times (2^{-2})^{13} = 2^{-14}$$

By applying the same lemma for 25 rounds the total bias (ε) can be given as,

$$\varepsilon = 2^{4} \times (2^{-14})^{5} = 2^{-66}$$

By calculating required number of known plaintext / cipher text, complexity of linear attack can be computed and can be given as,

$$N_L = 1/(\varepsilon)^2$$

For 25 rounds of the NUX cipher, the required number of known plaintext / cipher text can be given as

$$N_L = 1/(\varepsilon)^2 = 1/(2^{-66})^2$$
$$N_L = 2^{132}$$

The required number of known plaintext / cipher text is 2^{132} which is greater than the available limit i.e. 2^{64}. Hence, the complete round of the NUX cipher shows a good resistance against a Linear Attack.

C. Differential cryptanalysis

Differential attack [11] is another most significant attack similar as linear attack applied by Biham and Shamir on DES in 1990. To mount the differential attack for a specific number of rounds in an encryption system, pairs of high probability input and output occurrences are used to recover the round keys. Non linear layer is examined by DDT (Difference Distribution table). If S-box has non zero input difference and non zero output difference, such S-box referred as active S-box in differential cryptanalysis. Differential probability of NUX cipher S-box is 2^{-2}.

TABLE 4 represents the Difference Distribution TABLE for the NUX Cipher S-box. There are two approaches for providing security against differential cryptanalysis

1] By minimizing the differential probability, for an ideal S-box, this probability is 1/16.
2] It is necessary to find a structure that maximizes the minimum number of active S-boxes.

TABLE 5
MINIMUM NUMBER OF ACTIVE S-BOXES FROM DIFFERENTIAL TRAIL

#Round	# Min. active S-boxes
1	0
2	1
3	2
4	5
5	9

TABLE 5 represents the minimum number of active S-Boxes obtained from differential trails using Difference Distribution Table. For 5 rounds of the NUX cipher, there are a minimum of 9 active S-boxes. So, for 25 rounds, there will be a minimum of 45 active S-boxes. Total differential probability P_d is given as $(2^{-2})^{45} = 2^{-90}$.

The complexity of the differential attack can be computed by calculating the required number of chosen plaintext / cipher text and can be given as,

$$N_d = C/P_d$$

Where C = 1 and $P_d = 2^{-90}$, so the required number of chosen plaintext / cipher text are $N_d = 1/2^{-90} = 2^{90}$. The required numbers of chosen plaintext / cipher text is 2^{90} which are greater than available limit i.e. 2^{64}; hence complete rounds of the NUX cipher provide resistance against a Differential Attack.

D. Biclique attack [20]

This attack is an extension of Meet-In-The-Middle attack.
In this paper, biclique attack is mounting on NUX. A 3-dimensional Biclique was constructed for round 28 ~ 31 of NUX. For these rounds, the partial keys used are (K^{28}, K^{29}, K^{30}, K^{31}) which are described as follows:
$K^{28} = K_{51}, K_{50} \ldots . K_{20}$
$K^{29} = K_{38}, K_{37} \ldots \ldots . K_{7}$
$K^{30} = K_{25}, K_{24}, \ldots K_{0}, K_{127} \ldots . K_{122}$
$K^{31} = K_{12}, K_{11}, \ldots K_{0}, K_{127} \ldots . K_{109}$

From the above equations, it was found that by varying the following sub keys (K_{33}, K_{34},K_{35}) and (K_{07},K_{08},K_{09}), it gives Biclique on the full NUX cipher design.

To construct the Δi-differential, sub keys (K_{33},K_{34},K_{35}) have been considered and for the ∇j-differential, sub keys (K_7,K_8,K_9) have been considered. Let, f is a sub-cipher from round 28 to round 31. The Δi-differential affects the 64-bits of the cipher text as shown in Fig. 3. The data complexity does not exceed 2^{64}. Red color arrows at the 31st round in Fig. 2 represent the data complexity. The total computational complexity of NUX is computed as follows.

$$C_{total} = 2^{k-2d}\,(C_{biclique} + C_{precomp} + C_{recomp} + C_{falsepos}) = 2^{127.10}$$

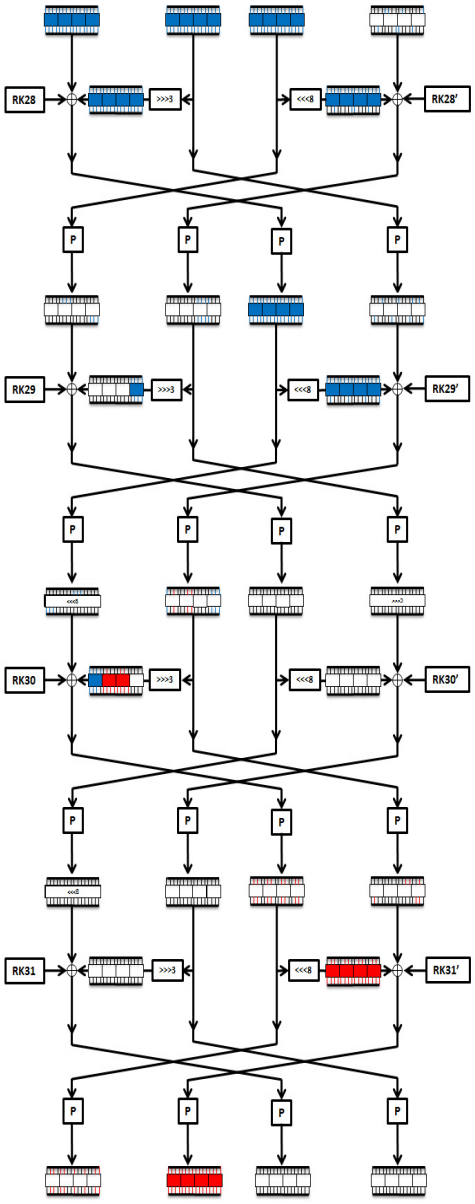

Fig. 3: 3 Dimensional biclique for NUX cipher

In the Fig. 3, red block indicates the active S-boxes due to 3 dimensional key RK29' while moving in forward path i.e. from 29th round to round 31st . This results in data complexity at the last round. The blue blocks also represent the active S-box due to 3 dimensional key at round 30 i.e. RK30 but it is traversing in reverse direction i.e. from round 30th to 28th . An important criterion to mount biclique is red trails and blue trails should not active the same S-boxes.

IV. Security Comparison With Standard Algorithm

In this section, the NUX cipher has been compared with the other existing lightweight ciphers. TABLE 8 compares the linear complexity and differential complexity by considering the minimum number of active S-boxes for particular rounds.

TABLE 8
LINEAR & DIFFERENTIAL ATTACK COMPARISON

Cipher Name	NUX	PRESENT	L-Block	FEW	PICCOLO
#Rounds	31	25	15	27	30
#Known Plaintext	$\mathbf{2^{132}}$	2^{102}	2^{66}	2^{90}	2^{120}
#Chosen Plaintext	2^{90}	2^{100}	2^{64}	2^{90}	2^{120}
Reference	This paper	[2]	[13]	[14]	[15]

TABLE 9 compares the data complexity and computational complexity of NUX with other ciphers.

TABLE 9
BICLIQUE ATTACK COMPARISON

Cipher Name	Rounds	Data Complexity	Computational Complexity	Reference
NUX	Full(31)	2^{24}	$2^{127.10}$	This Paper
PRESENT-80	Full(31)	2^{23}	$2^{79.54}$	[12]
PRESENT-128	Full(31)	2^{19}	$2^{127.42}$	[12]
PICCOLO-80	Full(25)	2^{48}	$2^{79.13}$	[12]
PICCOLO-128	Full(31)	2^{24}	$2^{127.35}$	[12]
LED-64	Full(48)	2^{64}	$2^{63.58}$	[12]
LED-80	Full(48)	2^{64}	$2^{79.37}$	[12]
LED-96	Full(48)	2^{64}	$2^{95.37}$	[12]
LED-128	Full(48)	2^{64}	$2^{127.37}$	[12]

V. Hardware and Software Performance of NUX Cipher

Design of NUX cipher is constructed in such a way that it results in optimum hardware and software performance. The NUX cipher uses S-box and XOR which consumes GEs otherwise shift operator, bit and block permutation only need wires. This compact structure of the NUX cipher results in a small footprint area and requires lesser memory size in software. The 32-bit ARM 7 LPC2129 processor is considered for analyzing the software performance of the NUX cipher. Other ciphers are also implemented on same platform. 16 bit permutation and 4x4 S-box occupies only 256 bytes of memory by using 8 bit data type in compiler. Reuse of variable and optimized function helps us to improve memory size of the cipher design.

Footprint size (GEs) is computed with the ARM standard cell library for the IBM 8RF (0.13 micron).The area of some basic gates in this library are: NOT 0.75, AND 1.25, OR 1.25, XOR 2.00, 2-1 MUX 2.25, D flip-flop 4.25.All other ciphers are written in Embedded C and implemented on a 32-bit processor for comparison with the NUX cipher. Fig. 3 represents the memory comparison of the existing lightweight ciphers with the NUX cipher.

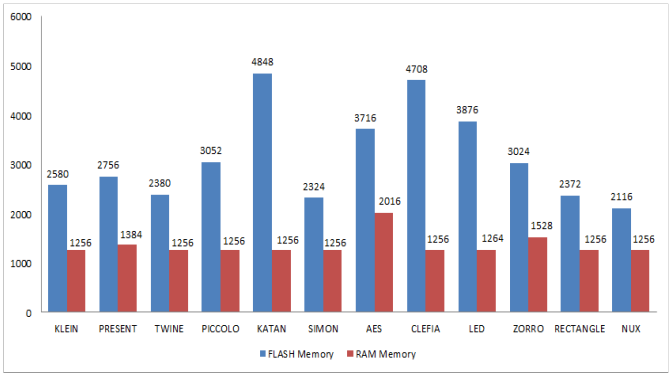

Fig. 3: Flash memory and RAM memory Comparison of Standard algorithms with NUX Cipher implemented on LPC2129

Table 10
A Memory Requirement Comparison Of NUX Cipher With Existing Ciphers

	PRESENT	LED	SIMON	TWINE	CLEFIA
NUX	-24.52%	-46.33%	-10.49%	-12.60%	-55.81%

Serialized architecture data path for NUX cipher is shown in Fig. 5.

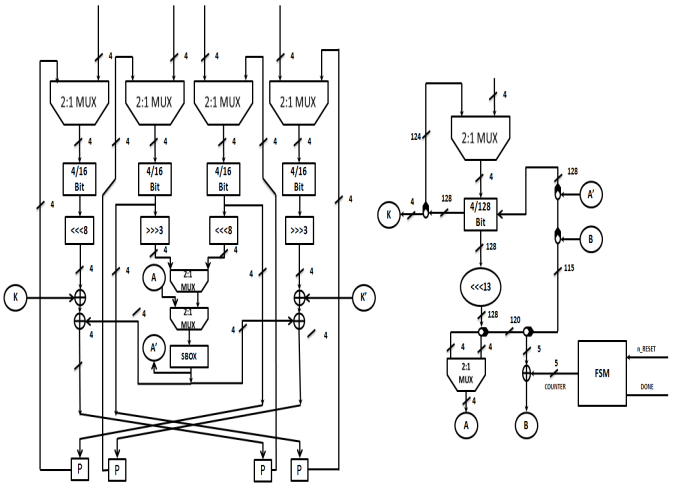

Fig. 4: Data path for NUX cipher for 64-bit plaintext and 128-bit key

Gate equivalent (GEs) is computed with ARM standard cell library for the IBM 8RF (0.13 micron). GEs calculation for the NUX cipher is represented in TABLE 11.

Table 11
Calculation Of GEs For NUX-128

Data Layer	GE's	Key Layer	GE's
D Reg.	272	D Reg.	544
2:1 MUX	13.5	2:1 MUX	4.5
XOR	32	XOR	10
SBOX	24	FSM	122
Shift Operator	0	Shift Operator	0
Total	341.5	Total	680.5
Total No. of gates required for 128 bit key = 1022			

Fig. 5 shows GEs comparison of other existing ciphers with NUX cipher

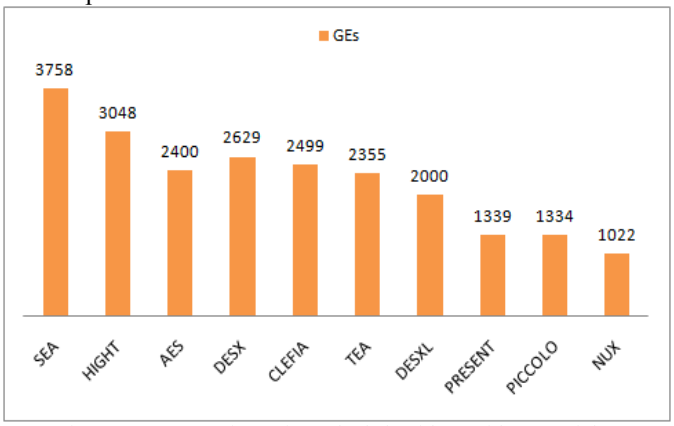

Fig. 5: GEs Comparison of Standard algorithms with NUX cipher

Following TABLE 12 shows the gate equivalent comparison of NUX cipher to existing cipher. From TABLE 12 we can conclude that NUX cipher has achieved the best result in GEs as compared to existing lightweight cipher. The NUX cipher has 770.5 GEs for 80 bit key.

TABLE 12
GES COMPARISON OF NUX

	PRESENT	CLEFIA	AES	PICCOLO
NUX	-23.67%	-59.10%	-57.41%	-25.48%

VI. CONCLUSION

In this paper, we proposed a generalized Feistel based cipher "NUX", which has maximal data complexity i.e. 2^{24} and results in maximum number of active S-boxes for fewer rounds. NUX cipher needs only 1022 GEs for 128 bit key length which is very less as compared to all existing lightweight ciphers. NUX cipher design is robust and is best suited for applications where footprints area, GE's are the major constraints. We believe NUX cipher is the smallest cipher design till date in terms of Gate Equivalents. With this cipher design we have achieved less GEs and competitive memory space.

NUX cipher not only resists basic attacks but also it resists advance attacks like MITM,and Biclique. NUX cipher design will have a positive impact in the field of lightweight cryptography, specifically this kind of cipher designs will prove to be a crusader in making applications like IoT feasible.

Test Vectors (For 128 bit key)

Plain text	Key	Cipher text
00000000 00000000	00000000 00000000 00000000 00000000	abfb02a9 87667de7
12345678 9ABCDEF0	00000000 00000000 00000000 00000000	192391A9 7EA83F66

REFERENCES

[1] Gaurav Bansod, Nishchal Raval, Narayan Pisharoty, "Implementation of a New Lightweight Encryption Design for Embedded Security", IEEE Transactions on Information Forensics and Security, Issue 1,Vol 10,Jan 2015.

[2] A. Bogdanov, G. Leander, L.R. Knudsen, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT - An Ultra-Lightweight Block Cipher," In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems — CHES 2007*, Vol. 4727 in *LNCS*, pages 450-466, Springer Berlin Heidelberg, 2007.

[3] Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, "RECTANGLE: A bit-slice ultra-lightweight block cipher suitable for multiple Platforms" *Cryptology ePrint Archive, Report 2014/084, 2014*. Available at https://eprint.iacr.org/2014/084.pdf

[4] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED Block Cipher," In *Cryptographic Hardware and Embedded Systems CHES 2011, LNCS*, Vol. 6917/2011, pages 326-341, Springer, 2011.

[5] "The 128 bit blockcipher" CLEFIA: Algorithm specification." On-linedocument, 2007. Sony Corporation.

[6] Wu, W., Zhang, L.: LBlock: A Lightweight Block Cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS. Lecture Notes in Computer Science, vol. 6715, pp. 327–344 (2011)

[7] S. Hong, D. Hong, Y. Ko, D. Chang, W. Lee, S. Lee: Differential Cryptanalysis of TEA and XTEA. In: ICISC'03, LNCS, vol. 2971, pp. 402–417, Springer-Verlag, 2004.

[8] Beaulieu, Ray, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. "The SIMON and SPECK lightweight block ciphers." In *Proceedings of the 52nd Annual Design Automation Conference*, p. 175. ACM, 2015.

[9] K. Nyberg. Generalized Feistel networks. In Kwangjo Kim and Tsutomu Matsumoto, editors, Advances in Cryptology - ASIACRYPT'96, LNCS 1163, pages 91–104. Springer Verlag, 1996.

[10] D. Coppersmith, "The data encryption standard (DES) and its strength against attacks," IBM Journal of Research and Development, Vol. 38, No. 3, May 1994, pp. 243–250.

[11] Howard M. Heys, "A Tutorial on Linear and Differential Cryptanalysis" ,http://citeseer.nj.nec.com/443539.html

[12] Jeong, K., Kang, H., Lee, C., Sung, J., Hong, S.: "Biclique Cryptanalysis of Lightweight Block Ciphers PRESENT, Piccolo and LED", *Cryptology ePrint Archive, Report 2012/621.*

[13] Wu, W., Zhang, L. "L-Block: A Lightweight Block Cipher". In: Lopez, J., Tsudik, G. eds. (2011) *Applied Cryptography and Network Security. Springer, Heidelberg, pp. 327-344*

[14] M Kumar, SK Pal and APanigrahi. "FeW: A Lightweight Block Cipher". Scientific Analysis Group, DRDO, Delhi, INDIA, Department of Mathematics, University of Delhi, INDIA2014.

[15] KyojiShibutani, TakanoriIsobe, HarunagaHiwatari, Atsushi Mitsuda, Toru Akishita, and TaizoShirai,"Piccolo: An Ultra-Lightweight Blockcipher", pages 342-357, Volume-6917 Springer Berlin Heidelberg, 2011.