Key Mixing in Block Ciphers through Addition modulo 2^n

Debdeep Mukhopadhyay[1] and Dipanwita RoyChowdhury[2]

[1] PhD Student, Department of Computer Science and Engg.
[2] Associate Professor, Department of Computer Science and Engg.
Indian Institute of Technology, Kharagpur, India

Abstract. The classical technique to perform key mixing in block ciphers is through exclusive-or (exor). In this paper we show that when the n-bit key is mixed in a block cipher of size n bits via addition modulo 2^n, the bias of the linear approximations falls exponentially fast. Experimental results have been provided to show that such a scheme cannot be cryptanalyzed using Linear Cryptanalysis.

Keywords: Block Ciphers, Key Mixing, Linear Approximations, Piling-Up Lemma

1 Introduction

Linear Cryptanalysis [1] is one of the most powerful and significant attacks applicable to symmetric key block ciphers. The block ciphers have to be designed so that they provide resistance to Linear cryptanalysis (LC). Although some design methodologies have been proposed, in [2–5], the systematic development of block ciphers with resistance against linear cryptanalysis is still a challenging task.

Linear Cryptanalysis essentially deals with the probability of approximating the input and output of non-linear functions, used in the block cipher with linear expressions [6]. The objective of LC is to obtain the last round key of a R round block cipher from the linear approximations of $(R-1)$ rounds. The linear approximation is achieved by combining the smaller linear expressions with large bias [6]. The bias of the linear expression is obtained using the Piling-Up lemma and has to be suitably high for the attack to successfully reveal the last round keys. From this lemma it is evident that for a linear expression with a large bias, the biases of each individual sub-expressions have to be significant. If one of them is negligible (almost zero), then the bias of the resultant expression is also negligible (almost zero) and does not lead to a successful linear cryptanalysis.

In Substitution-Permutation Network (SPN) like AES, DES the key mixing step is performed by key exoring where the key bits are simply exored (that is added without carry) with the data bits before each round and after the last round. In [7] the linear approximations of addition modulo 2^n (with carry) was studied. The author derived an $\theta(logn)$-time algorithm to compute the correlation of linear approximations of addition modulo 2^n. The algorithm is optimal and generates all linear approximations with a given non-zero correlation coefficient, and also determines the distribution of the correlation coefficients. The present paper investigates if the replacement of key exoring step in an n bit block cipher with addition modulo 2^n can reduce the bias of linear expressions in the cipher. Indeed some block ciphers like MARS [8], IDEA [9] use addition modulo 2^n inside their rounds. The present paper show both analytically and experimentally that such a key mixing operation can help to foil the powerful linear cryptanalysis. In the present work the maximum bias of linear approximations of addition modulo 2^n have been computed. It has been shown that the bias of linear approximations of the addition step falls exponentially fast with the bit position. Finally, a SPN cipher named GPIG1 have been taken and successfully cryptanalayzed using LC. Results have been presented to demonstrate that when the key mixing is performed through modulo 2^n addition (in block cipher GPIG2) LC fails to reveal the key.

In the next section (*section* 2) the maximum bias of linear approximations for addition modulo 2^n has been evaluated. *Section* 3 presents the construction of the block ciphers GPIG1 and GPIG2. *Section* 4 shows theoretically that the bias of sample linear approximations for GPIG2 is much less compared to those in GPIG1. *Section* 5 compares the linear attack on GPIG1 with that over GPIG2 and demonstrates that GPIG2 is a stronger cipher. The work is concluded in *section* 6.

2 Best Linear Approximation of Addition modulo 2^n

Block Ciphers use simple bit-wise exclusive OR between the key bits associated with a round and the data block input to a round. Also at the end there is a key exoring step with a round key, so that a cryptanalyst cannot easily work his way backwards.

Linear Cryptanalysis (LC) tries to take advantage of high probability occurences of linear expressions involving plaintext bits, ciphertext bits and subkey bits. The basic idea is to approximate a portion of the cipher with an expression that is linear, where linearity refers to a mod-2 bitwise exclusive or operation. The approach in LC is to determine expressions of the form which have a high or low probability of occurence. Let us consider an expression of the form:

$< X_{i1} \oplus X_{i2} \oplus \ldots X_{iu} > \oplus < Y_{j_1} \oplus Y_{j_2} \ldots \oplus Y_{j_v} >= 0$

where X_i represents the i-th bit of the input $X = [X_1, X_2, \ldots]$ and Y_j represents the j-th bit of the output $Y = [Y_1, Y_2, \ldots]$. This equation is representing the exclusive OR of u input bits and v output bits.

If the bits are chosen randomly then the above approximated linear expression will hold with probability 1/2. If p_l is the probability with which the expression holds then the bias is defined as $|p_l - 1/2|$.

Inorder to extract the key bits the cryptanalyst forms linear approximations for $R - 1$ rounds (if R is the total number of rounds) with large probability bias. Then the cryptanalyst attacks the last round subkeys or round keys. The probability of various linear expressions are formed and are collected using the Piling-Up Lemma to form bigger equations. The lemma is stated underneath without proof.

Lemma 1. *[1] For n independent, random binary variables $X_1, X_2, \ldots, X_n$, with bias $\epsilon_1, \epsilon_2, \ldots,$ ϵ_n,*

$$Pr(X_1 \oplus \ldots \oplus X_n = 0) = 1/2 + 2^{n-1} \prod_{i=1}^{n} \epsilon_i$$

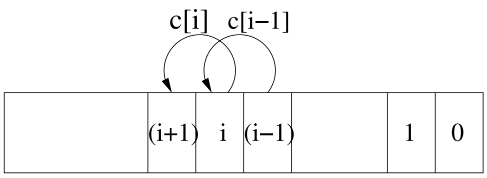
Thus if $X_1, X_2, \ldots, X_n$ are n linear approximations then the bias of the linear approximation made out of these n equations is denoted by [6, 10]:

$$\epsilon_{1,2,\ldots,n} = 2^{n-1} \prod_{i=1}^{n} \epsilon_i$$

Thus it is evident that if the bias of any of the linear approximation falls then the bias of the resultant equation also reduces. In the following theorems we compute the maximum bias of all possible linear approximations of addition modulo 2^n. Hence we obtain the best linear approximation of addition modulo 2^n in order to perform LC. Subsequently the biases are used to perform Linear Cryptanalysis (LC) against an SPN cipher, where the key is mixed using addition modulo 2^n. Results show that such a cipher becomes stronger against Linear Cryptanalysis.

Theorem 1. *For given n-bit inputs x and k the output is denoted by another n-bit number y=$(x$+$k)$ mod 2^n. The probability that each output bit $y[i]$ can be denoted by the linear function $x[i] \oplus k[i]$ is denoted by p_i, $0 \leq i < n$. Then $p_i = 1/2 + (1/2)^{i+1}$ and $1/2 < p_i \leq 1$.*

Proof. Let $c[i]$ denote the carry out from the addition of x and k after i bits, (refer figure 1). Clearly, $y[0] = x[0] \oplus k[0]$, with probability 1. Thus $p_0 = 1$.



1. The Output Register y which stores the sum of two registers x and k
2. 0, 1, ... ,(i−1), i, (i+1), ... indicates the bit positions of y
3. c[i−1] indicates the carry out after the addition of (i−1) bits are complete

Fig. 1. The Output State of the sum

Now, $y[1] = x[1] \oplus k[1]$ when there is no carry $c[0]$, fed from the 0^{th} bit.

Now, $c[0] = 0$, with probability $3/4$ and hence $p_1 = 3/4$.

Let, the event that the i_{th} bit of y can be expressed as a linear expression $x[i]$ and $k[i]$ has a probability p_i. Similarly the $(i+1)^{th}$ can be linearly expessed with a probability p_{i+1}.

Now, we note the following fact. The $(i+1)^{th}$ bit cannot be linearly expressed if there is a carry from the i^{th} bit, that is if $c[i]$=1.

This can be divided into two mutually exclusive cases. First the event say A, $c[i-1]$=0 and the addition of $x[i]$ and $y[i]$ generates a carry. Now, when $c[i-1]$ = 0, then $y[i]$ must have been linearly expressed (using the above fact) and the probability by definition is p_i. Thus the probability that A is true is $1/4.p_i$.

The other event B is the case where $c[i-1]$=1 and the addition of $x[i]$ and $y[i]$ propagates the carry. The probability that B is true is $3/4.(1-p_i)$.

Clearly if the event $(A \cup B)$ occurs then the $(i+1)^{th}$ bit cannot be linearly expressed and the probability is by definition $(1-p_{i+1})$.

Thus, $(1-p_{i+1}) = P(A \cup B) = P(A) + P(B)$ (because A and B are mutually exclusive)

$$= 1/4.p_i + 3/4.(1-p_i)$$

or, p_{i+1}=$1/4 + p_i/2$

Using the recurrence relation we have

$$\begin{aligned} p_{i+1} &= 1/4 + p_i/2 \\ &= 1/4 + 1/2(1/4 + p_{i-1}/2) \\ &= 1/4[1 + 1/2] + (1/2)^2 p_{i-1} \end{aligned}$$

Thus continnuing we have

$$\begin{aligned} p_{i+1} &= 1/4[1 + (1/2) + (1/2)^2 + \ldots + (1/2)^i] + (1/2)^{i+1} p_0 \\ &= 1/2[1 + (1/2)^{i+1}], \text{ since } p_0 = 1 \end{aligned}$$

Thus, $p_i = 1/2[1 + (1/2)^i] = 1/2 + (1/2)^{i+1}$.

Using the equation we have $p_0 = 1, p_1 = 3/4, p_2 = 5/8, p_3 = 9/16$ and so on.

Clearly, $1/2 < p_i \leq 1$.

Therefore, the bias of the linear approximation relating to the i^{th} bit position is $(p_i - 1/2) = 1/2^{i+1}$ and hence falls exponentially fast with i.

In the following theorems we compute the maximum value of the biases of all possible linear approximations of the sum bits. We show in the following theorem that the bias cannot be more than $1/2^{i+1}$.

Theorem 2. *For given n-bit inputs x and k the output is denoted by another n bit number $y = (x + k) \bmod 2^n$. The largest bias of a linear approximation of $y[i]$ is $(1/2)^{i+1}$.*

Proof. It is evident that, $y[i] = x[i] \oplus k[i] \oplus c[i-1]$, where $c[i-1]$ is the carry in of the i^{th} bit of the addition. The carry in is the non-linear part of the equation. Thus in order to obtain various linear approximations for the non-linear part linear approximations have to be found out for the carry in term. Each possible approximation of $c[i]$, denoted by $L[i]$ will give rise to different biases which are equal to the bias of a linear approximation of $y[i]$.

The equation for $c[0] = x[0]k[0]$, which is a boolean function for two variables.

Likewise, $c[1] = \text{majority}(x[1], k[1], c[0])$
$= \text{majority}(x[1], k[1], x[0]k[0])$
$= x[1]k[1] \oplus x[1]x[0]k[0] \oplus k[1]x[0]k[0]$.

Thus $c[1]$ is a boolean function of four variables.

Likewise, $c[i]$ is a boolean function for $2(i+1)$ variables.

The maximum non-linearity for an m variable boolean function, where m is even, is $2^{m-1} - 2^{m/2-1}$. Hence, the probability of match for the best linear approximation of a boolean function operating on an even number of variables is: $1 - \frac{2^{m-1}-2^{m/2-1}}{2^m} = \frac{1}{2} + 2^{-(m/2+1)}$.

Thus, the probability of matching for the best linear approximation for $c[i]$ is $1/2 + 2^{-(i+2)}$, substituting $m = 2(i+1)$.

The output $y[i] = x[i] \oplus k[i] \oplus c[i-1]$ can thus be approximated by a linear equation, $y'[i] = x[i] \oplus k[i] \oplus L[i-1]$, where $L[i-1]$ is the best linear approximation for $c[i-1]$.

Hence, the largest probability with which a linear approximation can match $y[i]$ is $1/2 + 2^{-(i-1+2)} = 1/2 + 2^{-(i+1)}$. Thus, the largest bias of a linear approximation for $y[i]$ is $(1/2)^{i+1}$.

From the above results it is evident that:

Corollary 1. *The best linear approximation for* $s[i]$ *is* $a[i] \oplus k[i]$*, where the probability of match is* $1/2 + 2^{-(i+1)}$ *and hence the bias is* $2^{-(i+1)}$.

So, if the key-mixing step in the block cipher is an addition modulo 2^n step, the probability of any linear expression relating to the key elements may be estimated using the above result and the Piling-Up lemma. If the resulting linear expression involves any particular bit position, say the i^{th} bit of the key, the bias of the resulting equation is lesser than $(1/2)^{i+1}$ and as the following table suggests the biases become negligible very fast.

The bias of the linear expression relating the key bits have been computed using the above expression and tabulated in table 1.

Table 1. Biases of Linear Approximations Involving Key Bits

Key Bit Position	0	1	2	3	4	5	↓ 6	7	8	9	10
Bias	0.5	0.25	0.125	0.0625	0.0313	0.0156	0.0079	0.0039	0.0020	0.0010	0.0004

We see that the bias of the linear approximations involving the key bits falls ver fast. With an expected key size of 128 the bias of the linear approximations is almost zero (negligible) beyond a bit position of six (marked in table 1). This fact makes the finding of linear approximations in the cipher with a large bias a more difficult task. Discovering the key through Linear Cryptanalysis becomes improbable.

In order to observe the effects of key mixing through addition on linear cryptanalysis we construct two SPN ciphers, GPig1 and GPig2. GPig2 differs from GPig1 in the fact that the key mixing is performed through addition modulo 2^n. First, the construction of the two block ciphers are highlighted in the following section.

3 Construction of the SPN Ciphers : GPig1 and GPig2

In this section we present the construction of Substitution-Permutation networks, GPig1 and GPig2, which is subsequently cryptanalyzed using linear cryptanalysis. The cipher, named GPig1, has been chosen from the tutorial presented in [10, 6]. The cipher GPig1 is essentially a traditional SPN block cipher, where the key mixing is performed by exoring between the data and the round keys. GPig1 is modified into another cipher and named GPig2, the only modification in the latter cipher being that the key mixing step is performed through addition modulo 2^n. In subsequent sections linear cryptanalysis against the modified cipher has been compared with that of the original cipher to demonstrate the benefit of the change.

3.1 The Substitution-Permutation Network-GPig1

In *figure 2* the unmodified block cipher *GPig1* is illustrated. The cipher takes a 16-bit input block and processes the block by repeating the basic operations of a round four times. Each round consists of

– Substitution
– a Transposition of bits (Permutation)
– a Key Mixing Step

This basic structure is the Fiestel Network and the basic operations are similar to those found in DES and in many modern ciphers, including Rijndael. Thus, the experimentation performed on the SPN cipher with respect to linear cryptanalysis is also applicable in case of standard and more practical block ciphers, without loss of generality.

The various blocks used in the block cipher are detailed next.

Substitution: In the cipher, the 16 bit data block data is subdivided into four groups (sub-blocks). Each sub-block forms an input to a 4×4 S-box (a substitution with 4 input and 4 output bits), which can be implemented easily with a table lookup of sixteen 4-bit values, indexed by the integer represented by the 4 input bits. For the cipher, the same S-box is chosen for all the rounds and is chosen from the S-boxes of DES. It is the first row of the first S-box. In table 2, the most significant bit of the hexadecimal notation represents the leftmost bit of the S-box in *figure 2*.

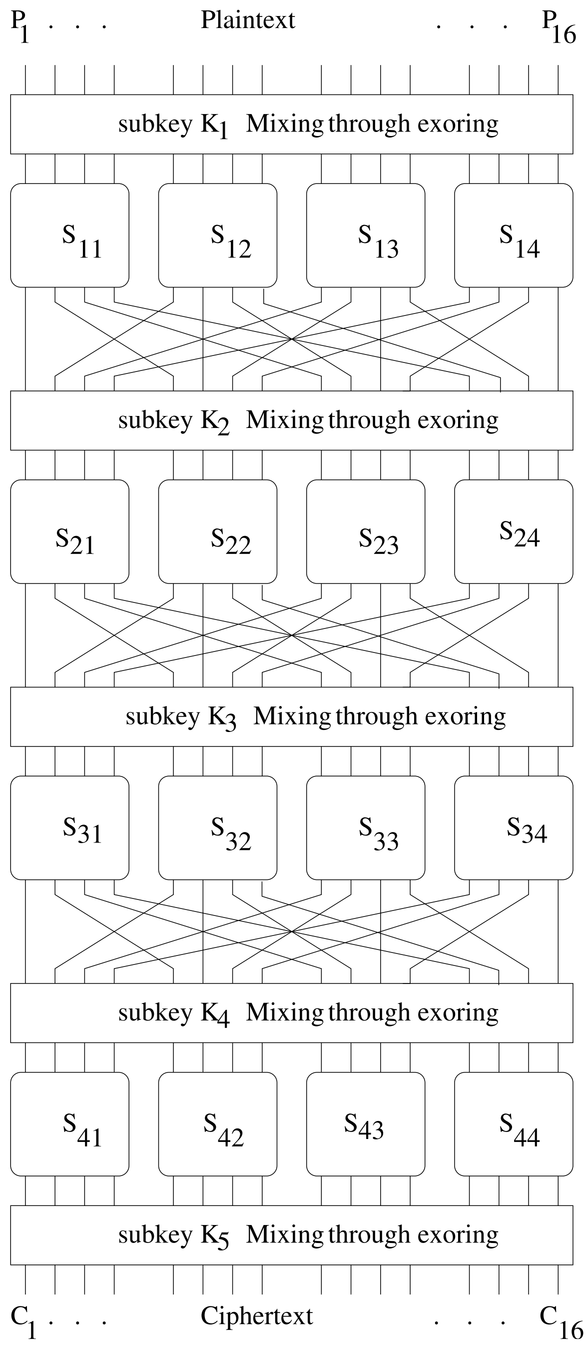


Fig. 2. The Structure of GPig1

Table 2. S-box Representation (in hexadecimal)

input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
output	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

Permutation: The permutation portion of a round is simply the transposition of the bits or the permutation of the bit positions. The permutation of *figure 2* is given in table 3 (where the numbers represent bit positions in the block, with 1 being the leftmost bit and 16 being the rightmost bit) and can be simply described as: the i^{th} input bit is connected to the j^{th} output bit (see *figure 2*).

Table 3. Permutation

input	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
output	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

Key Mixing: The key mixing is achieved in the block cipher through bit-wise exclusive-OR between the key bits associated with a round (referred to as a subkey) and the data block input to a round. The subkey for a round is derived from the master's key through a process known as the key schedule. In the cipher, we shall assume that all the subkeys are independently generated and are unrelated.

Decryption: In order to decrypt, data is essentially passed backwards through the network. However the S-boxes have to be bijective. Also, the subkeys have to be applied in the reverse order for proper decryption.

3.2 The modified SPN Cipher-GPig2

GPig2 is a similar block cipher as GPig1 with the only difference being in the key mixing step. Instead of exor operations bewteen the data of the i^{th} round (X_i) and the i^{th} round key (K_i), the key mixing in GPig2 is performed through addition modulo 2^{16}. Thus, we replace the key mixing step of the i^{th} round:

$$Y_i = X_i \oplus K_i$$

with, $Y_i = (X_i + K_i)\%2^{16}$, where $+$ represents the arithmetic addition operation. The symbol % is the modulo operation, st $0 \leq Y_i \leq 2^{16}$.

It is clear that the step is a reversible step, since $X_i = (Y_i - K_i)\%2^{16}$, where $-$ refers to signed arithmetic subtraction.

In the present section both GPig1 and GPig2 are analyzed under the light of linear attack. In order to start with the analysis we first need to analyze the S-box components and obtain linear approximations for the S-box, which is the same in both the ciphers.

4 Linear Cryptanalysis of GPig1 and GPig2

The linear approximations of the S-box is presented in [6, 10]. We summarise the result with a brief description. As *figure 3* shows, the input bits of the S-box are represented by X_1, X_2, X_3, X_4 and the output by Y_1, Y_2, Y_3, Y_4. A linear approximation involving the input bits is denoted by $a_1X_1 \oplus a_2X_2 \oplus a_3X_3 \oplus a_4X_4$, where $a_i \in \{0,1\}$. The approximation can be represented by a hexadecimal value $a_1a_2a_3a_4$, where a_1 is the most significant bit. Similarly the linear approximation involving the output bits, $b_1X_1 \oplus b_2X_2 \oplus b_3X_3 \oplus b_4X_4$, where $b_i \in \{0,1\}$, is denoted by the hexadecimal value $b_1b_2b_3b_4$. In order to obtain the probability of a linear approximation, all the 16 possible input values for X are applied, and the corresponding output values of Y are examined. The number of matches between the output Y and the linear approximation of the output is obtained (N). Thus the bias is $\frac{N}{16} - \frac{1}{2}$.

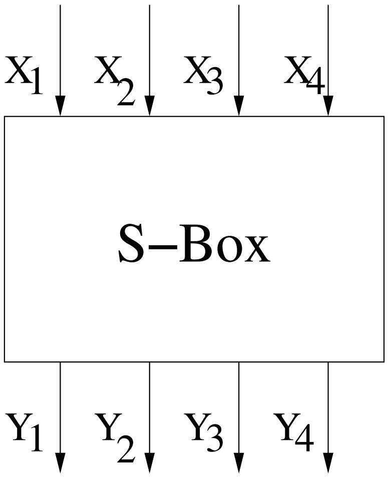


Fig. 3. S-box Mapping

For example, for the expression,

$X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4,$

it is observed that out of the 16 cases, 12 is the number of matches. Thus the probability of the linear approximation is $\frac{12}{16} = \frac{3}{4}$ and the bias is $\frac{3}{4} - \frac{1}{2} = \frac{1}{4}$.

A complete enumeration of all the linear approximations of the S-box in the cipher is given in table 4 [6]. The entries of the table are filled up with the values $N-8$. Thus, the bias for a linear approximation is obtained by dividing an entry in the table by 16. Hence, for the above example the input sum in hexadecimal is 6 and the corresponding output sum is B. Thus the corresponding entry in the table is +4 and therefore the bias is $+\frac{4}{16} = \frac{1}{4}$.

Table 4. Linear Approximation Table (LAT) of S-box

		Output Sum															
		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	0	0	-2	+6	+2	+2	0	0	+2	+2	0	0
	2	0	0	-2	-2	0	0	-2	-2	0	0	+2	+2	0	0	-6	+2
I	3	0	0	0	0	0	0	0	0	+2	-6	-2	-2	+2	+2	-2	-2
n	4	0	+2	0	-2	-2	-4	-2	0	0	-2	0	+2	+2	-4	+2	0
p	5	0	-2	-2	0	-2	0	+4	+2	-2	0	-4	+2	0	-2	-2	0
u	6	0	+2	-2	+4	+2	0	0	+2	0	-2	+2	+4	-2	0	0	-2
t	7	0	-2	0	+2	+2	-4	+2	0	-2	0	+2	0	+4	+2	0	+2
	8	0	0	0	0	0	0	0	0	-2	+2	+2	-2	+2	-2	-2	-6
S	9	0	0	-2	-2	0	0	-2	-2	-4	0	-2	+2	0	+4	+2	-2
u	A	0	+4	-2	+2	-4	0	+2	-2	+2	+2	0	0	+2	+2	0	0
m	B	0	+4	0	-4	+4	0	+4	0	0	0	0	0	0	0	0	0
	C	0	-2	+4	-2	-2	0	+2	0	+2	0	+2	+4	0	+2	0	-2
	D	0	+2	+2	0	-2	+4	0	+2	-4	-2	+2	0	+2	0	0	+2
	E	0	+2	+2	0	-2	-4	0	+2	-2	0	0	-2	-4	+2	-2	0
	F	0	-2	-4	-2	-2	0	+2	0	0	-2	+4	-2	-2	0	+2	0

4.1 Linear Approximations for the complete Ciphers

The biases of the linear approximations have been obtained for the S-boxes of the SPN networks. By concatenating appropriate linear approximations of the S-boxes, the linear approximations of the complete cipher involving plaintext bits and data bits from the output of the second last round of S-boxes are obtained, using the Piling-Up lemma. Following is an example of the calculation of the bias of a linear approximation of both the ciphers. It is evident from the results that the bias of a linear approximation for GPig2 is much lesser than that for GPig1.

In the following example, $U_i(V_j)$ represents the 16-bit block of bits at the input (output) of the round i S-boxes and $U_{i,j}$ $(V_{i,j})$ represent the j^{th} bit of block $U_i(V_j)$ (where the bits are numbered from 1 to 16 from left to right in *figure 2*). In case of GPig1 the 16-bit block key for the i^{th} round , K_i, is exclusive-ORed at the input to round i. However, K_5 is the key exclusive-ORed at the output of round 4. In the case of GPig2, instead of exclusive-OR, as already pointed out, the key bits are added modulo 2^n to the data blocks.

Example 1. Comparision of the probability biases of linear approximations for the first 3 rounds of GPig1 and GPig2

Sample Linear Approximation: $U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 = 0$

GPig1:

In order to obtain the linear approximation for the first two rounds we consider the following linear expressions:

1. $V_{1,6} = U_{1,5} \oplus U_{1,7} \oplus U_{1,8}$, with bias $\frac{1}{4}$

2. $U_{1,5} = P_5 \oplus K_{1,5}$, with bias $\frac{1}{2}$
3. $U_{1,7} = P_7 \oplus K_{1,7}$, with bias $\frac{1}{2}$
4. $U_{1,8} = P_8 \oplus K_{1,8}$, with bias $\frac{1}{2}$
5. $U_{2,6} = V_{2,6} \oplus V_{2,8}$, with bias $-\frac{1}{4}$
6. $U_{2,6} = V_{1,6} \oplus K_{2,6}$, with bias $\frac{1}{2}$

The concatenation of the above expression leads to the following approximation:

$$V_{2,6} \oplus V_{2,8} \oplus P_5 \oplus P_7 \oplus P_8 \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} = 0 \qquad (1)$$

The Piling-Up Lemma predicts that the bias of *equation 1* is equal to
$\beta_1 = 2^5(\frac{1}{4}\frac{1}{2}\frac{1}{2}\frac{1}{2}(-\frac{1}{4})\frac{1}{2}) = -\frac{1}{8}$.
Similarly, in order to obtain the linear approximation for the third round we consider the following expressions:

1. $U_{3,6} = V_{2,6} \oplus K_{3,6}$, with bias $\frac{1}{2}$
2. $U_{3,14} = V_{2,8} \oplus K_{3,14}$, with bias $\frac{1}{2}$
3. $U_{3,6} = V_{3,6} \oplus V_{3,8}$, with bias $-\frac{1}{4}$
4. $U_{3,14} = V_{3,14} \oplus V_{3,6}$, with bias $-\frac{1}{4}$

Combining the equations we arrive at the expression:

$$V_{3,6} \oplus V_{3,8} \oplus V_{3,14} \oplus V_{3,16} \oplus V_{2,6} \oplus K_{3,6} \oplus V_{2,8} \oplus K_{3,14} = 0 \qquad (2)$$

, with a bias of $\beta_2 = 2^3(\frac{1}{2}\frac{1}{2}(-\frac{1}{4})(-\frac{1}{4})) = +\frac{1}{8}$.
Combining *equation 1* and *equation 2* we get the expression:

$$V_{3,6}\oplus V_{3,8}\oplus V_{3,14}\oplus V_{3,16}\oplus P_5\oplus P_7\oplus P_8\oplus K_{1,5}\oplus K_{1,7}\oplus K_{1,8}\oplus K_{2,6}\oplus K_{3,6}\oplus K_{3,14} = 0 \qquad (3)$$

The following expressions:

1. $U_{4,6} = V_{3,6} \oplus K_{4,6}$
2. $U_{4,8} = V_{3,14} \oplus K_{4,8}$
3. $U_{4,14} = V_{3,8} \oplus K_{4,14}$
4. $U_{4,16} = V_{3,16} \oplus K_{4,16}$

, each having a bias of $\frac{1}{2}$, are combined with *equation 3* to finally obtain:

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 \oplus \sum_K = 0 \qquad (4)$$

,where $\sum_K = K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,14} \oplus K_{4,16}$.
Hence, using Piling-Up Lemma the bias of the equation is:
$\beta_3 = 2^5(\beta_1\beta_2(\frac{1}{2^4})) = 2^5((-\frac{1}{8})(\frac{1}{8})(\frac{1}{2^4})) = -\frac{1}{32}$.
Now, since $\sum_K$ is fixed (that is either 0 or 1 depending on the key bits), the linear approximation

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 = 0 \quad (5)$$

holds with probability $\frac{1}{2} - \frac{1}{32} = \frac{15}{32}$. or $1 - \frac{15}{32}$, depending on whether $\sum_K$ is 0 or 1.

Thus, the bias of the linear expression (*equation 5*) has a magintude of $\frac{1}{32}$.

Next, we compute the bias of the linear expression in the case of the cipher GPig2.

GPig2:

In order to obtain the bias of the linear approximation a similar calculation is performed.

The biases of the linear approximations of the S-boxes are identical for both the ciphers. Only the biases of the linear expressions involving the key bits are different for GPig2 and are computed using Theorems 2 and 3. We first enumerate the linear expressions involving the key bits and the corresponding biases:

1. $U_{1,5} = P_5 \oplus K_{1,5}$, with bias $\frac{1}{2^6}$
2. $U_{1,7} = P_7 \oplus K_{1,7}$, with bias $\frac{1}{2^8}$
3. $U_{1,8} = P_8 \oplus K_{1,8}$, with bias $\frac{1}{2^9}$
4. $U_{2,6} = V_{1,6} \oplus K_{2,6}$, with bias $\frac{1}{2^7}$

Thus, the bias of *equation 1* in case of GPig2 is :
$\beta_1{}' = 2^5(\frac{1}{4}\frac{1}{2^6}\frac{1}{2^8}\frac{1}{2^9}(-\frac{1}{4})\frac{1}{2^7}) = -\frac{1}{2^{29}}$.

In order to obtain the linear approximation for round 3, the linear expression involving the key bits are:

1. $U_{3,6} = V_{2,6} \oplus K_{3,6}$, with bias $\frac{1}{2^7}$
2. $U_{3,14} = V_{2,8} \oplus K_{3,14}$, with bias $\frac{1}{2^{15}}$

Thus the bias of *equation 2* becomes:
$\beta_2{}' = 2^3(\frac{1}{2^7}\frac{1}{2^{15}}(-\frac{1}{4})(-\frac{1}{4})) = \frac{1}{2^{23}}$.

In order to arrive at the final expression (*equation 3*), the expressions involving the key bits of round 4 are

1. $U_{4,6} = V_{3,6} \oplus K_{4,6}$, with bias $\frac{1}{2^7}$
2. $U_{4,8} = V_{3,14} \oplus K_{4,8}$, with bias $\frac{1}{2^9}$
3. $U_{4,14} = V_{3,8} \oplus K_{4,14}$, with bias $\frac{1}{2^{15}}$
4. $U_{4,16} = V_{3,16} \oplus K_{4,16}$, with bias $\frac{1}{2^{17}}$

Thus, the bias of *equation 5* is:
$\beta_3{}' = 2^5(\beta_1{}'\beta_2{}'\frac{1}{2^7}\frac{1}{2^9}\frac{1}{2^{15}}\frac{1}{2^{17}}) = 2^5((-\frac{1}{2^{29}})(\frac{1}{2^{23}})(\frac{1}{2^{48}})) = \frac{1}{2^{95}} \approx 0$.

The above example demonstrates that when the key mixing step in the SPN block cipher is performed with the help of addition modulo 2^n, the bias of the linear expressions are almost zero, and thus cannot be used in linear cryptanalysis.

In the following section we perform a linear attack on both the ciphers, GPig1 and GPig2 and evaluate the strength of the second cipher against the cryptanalysis.

5 Experimental Extraction of Key Bits

In this section it is experimentally shown that GPig1 is successfully cryptanalyzed using the linear expression, mentioned in the example. It is also demonstrated that for GPig2 such an attack does not work. The reason being, in order for linear cryptanalysis to be successful the bias of $(R-1)$ round linear expressions (approximations) for an R round block cipher has to be large. However, in the case of GPig2 the biases of linear expressions falls very fast to zero and hence such equations cannot be exploited in a conventional linear attack.

5.1 Experimental Setup

The procedure adopted to evaluate the last round keys are as follows:

1. A large number $(10,000)$ of cipher-texts are obtained by encrypting plaintexts i,e we generate $10,000$ known plaintext/ ciphertext pairs.
2. The attacker considers the linear approximation (mentioned in the example) of the first 3 rounds of the ciphers. To restate the expression is:

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 = 0 \tag{6}$$

   The terms $U_{4,6}$, $U_{4,8}$ and $U_{4,14}$ affects the S-boxes S_{42} and S_{44}. Hence, the attacker guesses $(K_{5,5},\ldots,K_{5,8})$ and $(K_{5,13},\ldots,K_{5,16})$. In case of GPig1 he exors them with the ciphertext bits to obtain $(V_{5,5},\ldots,V_{5,16})$. Then he performs the inverse of the S-Box operations to obtain the values of $U_{4,6}$, $U_{4,8}$ and $U_{4,14}$. If their values satisfy *equation 6* then a count is incremented for the guessed key bits $(K_{5,5},\ldots,K_{5,8},K_{5,13},\ldots,K_{5,16})$. The partial subkey which has the count which differs greatest from half the number of plaintext/ ciphertext samples $(50,000)$ is assumed to represent the correct values of the guessed key bits. An incorrect subkey is assumed to be equivalent to a random guess to the bits of the linear expression and this holds with probability close to $1/2$. The same attack is also performed on GPig2. Only we assume that the attacker knows the values of the key bits $(K_{5,9},\ldots,K_{5,12})$. Thus here he guesses the partial keys $(K_{5,5},\ldots,K_{5,8},K_{5,13},\ldots,K_{5,16})$ and subtracts the key bits from the ciphertext to arrive at the required values of $(V_{5,5},\ldots,V_{5,16})$ and finally the values of $U_{4,6}$, $U_{4,8}$ and $U_{4,14}$. The rest of the attack is similar. This gives a best case scenario to the attacker.
   From, the table we see that the attack works fine for GPig1, where the correct subkey bits (last round) keys $(K_{5,5},\ldots,K_{5,8},K_{5,13},\ldots,K_{5,16}) =$ [2,4] leads to the largest bias of 0.0308 and is thus detected. The bias is also close to the calculated bias of 1/32=0.03125.
   However the same attack on GPig2 shows that the bias of the expression for the correct key bits [2,4] is only 0.0010 which is less than the biases of the incorrect key bits. The result implies that the probability of linear expressions to hold in case of GPig2 is much close to 1/2 and is thus very hard to differentiate from a random guess. Thus GPig2 offers a much better resistance

to linear cryptanalysis than GPig1. Also note that in the experimentation it was observed that the highest bias (0.0139) occured for a key bit = $E9$, which is an incorrect key.

Table 5. Experimental Results for Linear Attack

Partial SubKey	Bias		Partial SubKey	Bias	
$[K_{5,5}, .., K_{5,8}, .., K_{5,13}]$	XOR	ADD	$[K_{5,5}, .., K_{5,8}, .., K_{5,13}]$	XOR	ADD
1C	0.0023	0.0027	2A	0.0099	0.0030
1D	0.0042	0.0084	2B	0.0053	0.0044
1E	0.0013	0.0006	2C	0.0060	0.0120
1F	0.0055	0.0034	2D	0.0107	0.0034
20	0.0011	0.0023	2E	0.0074	0.0061
21	0.0061	0.0053	2F	0.0024	0.0012
22	0.0028	0.0049	30	0.0137	0.0002
23	0.0075	0.0067	31	0.0151	0.0043
24	0.0308	0.0010	32	0.0104	0.0048
25	0.0156	0.0079	33	0.0151	0.0010
26	0.0148	0.0022	34	0.0090	0.0025
27	0.0011	0.0003	35	0.0130	0.0048
28	0.0266	0.0009	36	0.0078	0.0034
29	0.0107	0.0046	37	0.0025	0.0020

Max Bias for XOR: 0.0308 for the correct Key 24H
Max Bias for Add: 0.0139 for an incorrect Key E9H

6 Conclusion

In the present paper the conventional key mixing have been altered from exor to addition modulo 2^n. The largest bias of linear approximations for the output bit of such a key mixing have been computed. Both theoretically and experimentally it has been demonstrated that such a modification makes the cipher strong against Linear Cryptanalysis.

References

1. Mitsuru Matsui, "Linear Cryptanalysis method for DES cipher," in *Advances in Cryptology-Eurocrypt 1993*. 1993, pp. 386–397, Springer, volume 765 of LNCS.
2. Florent Chabaud and Serge Vaudenay, "Links between Differential and Linear Cryptanalysis," in *Advances in Cryptology-Eurocrypt 1994*. 1994, pp. 356–365, Springer, volume 950 of LNCS.

3. Mitsuru Matsui, “New structure of block ciphers with provable security against linear and differential cryptanalysis,” in *Fast Sofware Encryption 1996*. 1996, pp. 205–218, Springer, volume 1039 of LNCS.
4. Kaisa Nyberg, “Linear approximations of block ciphers ,” in *Advances in Cryptology-Eurocrypt 1995*. 1995, pp. 439–444, Springer, volume 950 of LNCS.
5. Joan Daemen, *Cipher and Hash Function Design: Methods Based on Linear and Differential Cryptanalysis*, Ph.D. thesis, Katholieke Universiteit Leuven, March, 1995.
6. Howard M. Keys, “A Tutorial on Linear and Differential Cryptanalysis,” www.engr.mun.ca/ howard/PAPERS/ldc_tutorial.ps.
7. Johan Wallen, “Linear approximations of Addition Modulo 2^n ,” in *Fast Sofware Encryption 2003*. 2003, pp. 261–273, Springer, volume 2887 of LNCS.
8. C. Burwick, D.Coppersmith, E. D Avignon, R. Gennaro, S. Halevi, C. Jutla, S. M. Matyas, L. O Connor, M. Peyravian, D. Safford and N. Zunic, “Mars - a candidate cipher for AES,” in *First Advanced Encryption Standard (AES) Conference, Ventura, CA*, 1998.
9. X. Lai and J.L. Massey, “A Proposal for a New Block Encryption Standard,” in *Advances in Cryptology-Eurocrypt'90*. 1991, pp. 389–404, Springer Verlag.
10. Douglas R. Stinson, *Cryptography : Theory and Practice*, chapter 3, pp. 79–88, CRC Press Company, 2002.