

Strengthening NLS against Crossword Puzzle Attack

Debojyoti Bhattacharya¹, Debdeep Mukhopadhyay², Dhiman Saha³
, and D. RoyChowdhury⁴

¹ IIT-Kharagpur, Kharagpur, India {debojyoti.bhattacharya@gmail.com}

² IIT-Madras, Chennai, India {debdeep@cse.iitm.ernet.in}

³ IIT-Kharagpur, Kharagpur, India {dhimans@cse.iitkgp.ernet.in}

⁴ IIT-Kharagpur, Kharagpur, India {drc@iitkgp.ac.in}

Abstract. NLS is a stream cipher proposal submitted to eSTREAM project. In SAC 2006 Cho and Pieprzyk presented a linear distinguishing attack called Crossword Puzzle attack on NLS where they have shown that the bias of the distinguisher is around $O(2^{-30})$. In this work we have proposed a new function modular *Slash* which is nonlinear in nature and strongly resistant against Linear Cryptanalysis. Replacing the modular addition in the nonlinear filter (NLF) of NLS we have shown that the Crossword puzzle attack presented by Cho and Pieprzyk can be prevented. In the modified NLS the bias of the linear distinguisher reduces to around $O(2^{-60})$. Also we have shown that the implementation cost of modular *Slash*, in terms of hardware and time delay, is less than modular addition. The proposed function could be an interesting alternative to modular addition, due to its better cryptographic properties and lesser implementation cost.

Key words : Stream ciphers, eSTREAM, Crossword Puzzle attack, Linear Approximations, Modular Addition, NLS

1 Introduction

A stream cipher project called eSTREAM [1] has been launched by the European Network of Excellence in Cryptography (ECRYPT), to come up with a collection of stream ciphers as de facto standard in industry and government institutions as secure and efficient cryptographic primitives. A variety of different design approaches has been followed by the designers in different submissions and a variety of cryptanalytic techniques are also submitted to cryptanalyze and assess the security of those submitted stream ciphers. In traditional stream ciphers, linear feedback shift register (LFSR) is used as one of the major components. The output of the shift registers are passed to a nonlinear filter (NLF) to produce the keystream. In recent days, modern stream ciphers are using nonlinear feedback shift register (NFSR) in place of LFSR. Several ciphers submitted to eSTREAM follow this approach. NLS [2] is one of the stream ciphers submitted to eSTREAM and also a candidate in phase 2, follows this design approach.

In [3], Joo Yeon Cho and Josef Pieprzyk studied the NLS cipher and its resistance against linear distinguishing attacks. Though the distinguishing attacks do not allow to recover cryptographic key or any secret element of the cipher under observation, the attack is important in the sense that it helps to distinguish the cipher under attack from a truly random cipher.

In [3], Joo Yeon Cho and Josef Pieprzyk proposed an excellent linear distinguishing attack namely “**Crossword Puzzle attack**” (CP attack) against NLS where they derived linear approximations for both the NFSR and the nonlinear filter (NLF) and combined those approximations to build a linear distinguisher to distinguish the output key-stream generated by NLS from a truly random cipher. They showed that the bias of the distinguisher is around $O(2^{-30})$ for NLSv1 and hence the complexity of the attack is $O(2^{60})$ keystream words. They also extend their attack to NLSv2 where the bias of the distinguisher is found to be around $O(2^{-48})$. Hence they claimed that the security margin of NLS is small to guarantee the claimed security level in future.

In this paper, we propose a new boolean function named *Slash* (denoted by \oslash) which offers high non-linearity keeping the hardware implementation overhead small. We cryptanalyzed our proposed function to show that it offers high security against Linear Cryptanalysis. We modified NLS by replacing the modular addition used in the nonlinear filter (NLF) by modular *Slash* and showed theoretically that the **CP** attack proposed in [3] fails. The bias of the linear distinguisher built in the method of Joo Yeon Cho and Josef Pieprzyk reduces to a value of around $O(2^{-60})$ and hence the complexity of the attack increases to around $O(2^{120})$ keystream words. We show our computations only for NLSv1. We also give the hardware architectural comparison of modular *Slash* function with modular addition to show that both the hardware complexity and the time delay to realize modular *Slash* are less than modular addition.

The rest of the work is organised as follows. Section 2 discusses some preliminaries required for this work. Section 3 explores our proposed function. Performance of the proposed function against Linear Cryptanalysis has been discussed in Section 4. Section 5 briefly describes the NLS cipher and our suggested modification. A brief description of the framework of the CP attack has been discussed in Section 6. The analysis of the linear approximations for the NFSR and the NLF for both the original NLS and the modified NLS are given in Section 7. In Section 8 the complexity comparison of the CP attack on the original NLS and the modified NLS has been discussed. Hardware and time delay comparison of the new function with addition modulo 2^n are given in Section 9. Section 10 concludes the work.

2 Preliminaries

Some basic definitions and notations have been discussed in this section. A boolean function of n variables $g(x)$ is a map $g(x): F_2^n \rightarrow F_2$, where F_2^n is a vector space defined over F_2 . The operation $x \oplus y$ on two binary strings x and y is the bitwise exclusive *OR* operation between the strings x and y . The

Hamming weight of a binary string x is the number of 1's in the string and is denoted by $wt(x)$. The Hamming distance between two binary strings of equal length (say x and y) is the number of positions where x and y differ and is measured by $wt(x \oplus y)$.

Definition 1 A boolean function $g(x)$, where x is an n variable binary string, can be uniquely written as a sum (XOR) of products (AND). This is known as Algebraic Normal Form (ANF).

$$g(x_1, x_2, \dots, x_n) = p_0 \oplus p_1 x_1 \oplus p_2 x_2 \oplus \dots \oplus p_n x_n \oplus p_{1,2} x_1 x_2 \oplus p_{n-1,n} x_{n-1} x_n \oplus \dots \oplus p_{1,2,\dots,n} x_1 x_2 \dots x_n.$$

The values of $(p_0, p_1, \dots, p_{1,2,\dots,n} \in \{0, 1\})$ uniquely represent a boolean function.

Definition 2 An n variable boolean function $g(x_1, x_2, \dots, x_n)$ is said to be an affine function if the ANF of g is of the form $g(x_1, x_2, \dots, x_n) = \bigoplus_{i=0}^n p_i x_i \oplus q$ for $p_i, q \in \{0, 1\}$. If q is 0, then the function is said to be linear.

Definition 3 Non-linearity of an n variable boolean function g is defined as the minimum Hamming distance from the set of all affine functions of n variables, i.e., $N_f = \min_{a \in A_n} d_H(g, a)$, where Hamming distance is defined as, $d_H(g, a) = \{\#x | g(x) \neq a(x)\}$. A_n is the set of all n variable affine functions.

Definition 4 [4] A boolean function $g(x)$ of n variable, where n is even, is called a Bent function if it has a non-linearity value $2^{n-1} - 2^{\frac{n}{2}-1}$. This is the highest possible non-linearity for an n variable boolean function if n is even.

Theorem 1. [5] The boolean function $g(x_{n-1}, x_{n-2}, \dots, x_0) = x_{n-1} x_{n-2} \oplus x_{n-3} x_{n-4} \oplus \dots \oplus x_3 x_2 \oplus x_1 x_0$ is a bent function having nonlinearity value $2^{n-1} - 2^{\frac{n}{2}-1}$, where ' n ' is even and x_n, x_{n-1}, \dots, x_0 are n independent random boolean variables.

Definition 5 A bias $\epsilon(a, b)$ is defined as $P = \frac{1}{2}(1 + \epsilon)$, $|\epsilon| > 0$ where P is the probability that an approximation holds.

Piling-up Lemma [6] If we have n independent approximations having biases $\epsilon_1, \dots, \epsilon_n$, then the bias of the approximation combining these n approximations becomes $\prod_{i=1}^n \epsilon_i$.

3 Proposed Boolean Operator : Slash

Definition 6 Slash: It is defined as an operation ' \oslash ' which operates on two 1 bit boolean variables A and B and produces a 2 bit output C_{out} , O such that $A \oslash B = C_{out} \parallel O$. The output bits are defined as, $O = A \oplus B$ and $C_{out} = AB$. For three 1 bit boolean variables, A, B , and C_{in} the definition extends to $A \oslash B \oslash C_{in}$ and the output bits are expressed as $O = A \oplus B \oplus C_{in}$ and $C_{out} = AB \oplus C_{in}$.

We present a function modular *Slash* using *Slash* operator below with proof of its reversibility.

- **Forward** : Let $X = (x_{n-1}, x_{n-2}, \dots, x_0)$ and $Y = (y_{n-1}, y_{n-2}, \dots, y_0)$ be two n -bit data and $Z = (z_{n-1}, z_{n-2}, \dots, z_0)$ be the n -bit output, where

x_0, y_0, z_0 denote the LSBs and $x_{n-1}, y_{n-1}, z_{n-1}$ denote the MSBs. We define the function $Z = F(X, Y)$ as below:

$$\begin{aligned} z_i &= x_i \oplus y_i \oplus c_{i-1} \\ c_i &= x_i y_i \oplus c_{i-1} \\ c_{-1} &= 0 \end{aligned}$$

c_i is the carry term propagating from i^{th} bit position to $(i+1)^{th}$ bit position. The definition of c_i is recursive as shown in the equation. The end carry c_{n-1} is neglected. This defines the operation $Z = F(X, Y) = (X \oslash Y) \bmod 2^n$ (Definition 6).

- **Inverse** : Let $Z = (z_{n-1}, z_{n-2}, \dots, z_0)$ be an n -bit input, $Y = (y_{n-1}, y_{n-2}, \dots, y_0)$ be another n -bit input and $X = (x_{n-1}, x_{n-2}, \dots, x_0)$ be the n -bit output, notation of LSB and MSB being the same as of the forward. We define the inverse function $X = G(Z, Y)$ as below:

$$\begin{aligned} x_i &= z_i \oplus y_i \oplus d_{i-1} \\ d_i &= x_i y_i \oplus d_{i-1} \\ d_{-1} &= 0 \end{aligned}$$

d_i is the carry term propagating from i^{th} bit position to $(i+1)^{th}$ bit position. The definition of d_i is recursive as shown in the equation. It can be noted here that $d_i = (z_i \oplus y_i \oplus d_{i-1})y_i \oplus d_{i-1} = z_i y_i \oplus y_i \oplus d_{i-1} y_i \oplus d_{i-1} = y_i(\neg z_i) \oplus d_{i-1}(\neg y_i)$. This definition of d_i has been used in the hardware design and result is shown in Table 2. The end carry d_{n-1} is neglected.

Theorem 2. *If X, Y, Z be three n -bit data such that $Z = F(X, Y)$, where $z_i = x_i \oplus y_i \oplus c_{i-1}$, $c_i = x_i y_i \oplus c_{i-1}$ and $c_{-1} = 0$ and G is defined as $X = G(Z, Y)$, where $x_i = z_i \oplus y_i \oplus d_{i-1}$, $d_i = x_i y_i \oplus d_{i-1}$ ($\forall 0 \leq i < n$) and $d_{-1} = 0$ then G is the inverse function of F .*

Proof. Let, $z_i \oplus y_i \oplus d_{i-1} = p_i$
Given that, $c_i = x_i y_i \oplus c_{i-1}$

$$\begin{aligned} \therefore c_{i-1} &= x_{i-1} y_{i-1} \oplus c_{i-2} \\ &= x_{i-1} y_{i-1} \oplus \dots \oplus c_0 \\ &= x_{i-1} y_{i-1} \oplus \dots \oplus x_0 y_0 \end{aligned}$$

According to the definition of F , $z_i = x_i \oplus y_i \oplus c_{i-1}$
According to the definition of G , $d_i = x_i y_i \oplus d_{i-1}$

$$\begin{aligned} \therefore d_{i-1} &= x_{i-1} y_{i-1} \oplus d_{i-2} \\ &= x_{i-1} y_{i-1} \oplus \dots \oplus d_0 \\ &= x_{i-1} y_{i-1} \oplus \dots \oplus x_0 y_0 \\ &= c_{i-1} \end{aligned}$$

$$\begin{aligned}
\therefore p_i &= z_i \oplus y_i \oplus d_{i-1} \\
&= z_i \oplus y_i \oplus c_{i-1}, (\text{ putting value of } d_{i-1}) \\
&= (x_i \oplus y_i \oplus c_{i-1}) \oplus y_i \oplus c_{i-1} \\
&= x_i
\end{aligned}$$

Hence the proof.

The following corollary follows from the definition of $F(X, Y)$.

Corollary 1. *If X and Y are two n -bit numbers, then the operation F is commutative, i.e $F(X, Y) = F(Y, X)$.*

4 Performance of *Slash* against Linear Cryptanalysis

In this section we give the performance measurement of our proposed function against Linear Cryptanalysis (LC). Throughout the analysis we will consider $X = (x_{n-1}, \dots, x_0)$ and $Y = (y_{n-1}, \dots, y_0)$ are two mutually independent random variables of n bits each and each of the n bits of X and Y are mutually independent random boolean variables.

Theorem 3. *If two n bit numbers, $X = (x_{n-1}, x_{n-2}, \dots, x_0)$ and $Y = (y_{n-1}, y_{n-2}, \dots, y_0)$ generate an n bit number $Z = (z_{n-1}, z_{n-2}, \dots, z_0)$ such that, $Z = F(X, Y)$, then the probability p_i of denoting z_i , each output bit of Z by the linear function $x_i \oplus y_i$ is $p_i = \frac{1}{2}(1 + (\frac{1}{2})^i)$ and p_i lies in the range $\frac{1}{2} < p_i \leq 1$ as i lies in $0 \leq i < n$.*

Proof. We denote the carry propagated from the i^{th} bit position to $(i+1)^{th}$ bit position as c_i . As per the definition of F , $c_{-1} = 0$. Hence $z_0 = x_0 \oplus y_0$ with probability 1. Therefore $p_0 = 1$. Now, $z_1 = x_1 \oplus y_1$ if there is no carry c_0 . But, $c_0 = 0$ holds with a probability of $\frac{3}{4}$ as $c_0 = x_0 y_0$. Hence $p_1 = \frac{3}{4}$. Let p_i be the probability of denoting z_i as $z_i = x_i \oplus y_i$. Similarly z_{i+1} can be expressed linearly with a probability of p_{i+1} .

Fact: It is clear hereby that, z_{i+1} can be expressed linearly if the carry term from i^{th} bit position, $c_i = 0$.

This scenario can be expressed as the union of the following two cases.

- **Event A :** This is the case when $c_{i-1} = 0$ and $x_i \odot y_i$ generates a carry. If $c_{i-1} = 0$, then z_i could have been expressed linearly (using the **Fact** stated above) and the probability of that by definition is p_i . Hence, the probability that A is true is $\frac{1}{4}.p_i$.
- **Event B :** This is the case where $c_{i-1} = 1$ and $x_i \odot y_i$ generates a carry. The probability that event B is true is $\frac{3}{4}.(1 - p_i)$.

From the above two events it is clear that z_{i+1} cannot be expressed linearly if the event $(A \cup B)$ occurs. By definition the probability of this event is $(1 - p_{i+1})$, as p_{i+1} is the probability that z_{i+1} can be expressed linearly.

$$\begin{aligned}
\therefore (1 - p_{i+1}) &= P(A \cup B) \\
&= P(A) + P(B) \text{ (A and B are mutually} \\
&\quad \text{exclusive events)} \\
&= \frac{1}{4} \cdot p_i + \frac{3}{4} \cdot (1 - p_i) \\
\Rightarrow p_{i+1} &= \frac{1}{4} + \frac{1}{2} \cdot p_i
\end{aligned}$$

Using the above recurrence relation we can write,

$$\begin{aligned}
p_{i+1} &= \frac{1}{4} + \frac{1}{2} \cdot p_i \\
&= \frac{1}{4} + \frac{1}{2} \cdot \left(\frac{1}{4} + \frac{1}{2} \cdot p_{i-1} \right) \\
&= \frac{1}{4} \cdot \left(1 + \frac{1}{2} \right) + \left(\frac{1}{2} \right)^2 p_{i-1} \\
&\quad \vdots \\
&= \frac{1}{4} \cdot \left(1 + \frac{1}{2} \right) + \dots + \left(\frac{1}{2} \right)^i + \left(\frac{1}{2} \right)^{i+1} p_0 \\
&= \frac{1}{2} \cdot \left(1 + \left(\frac{1}{2} \right)^{i+1} \right), \text{ as } p_0 = 1
\end{aligned}$$

Hence, $p_i = \frac{1}{2} \cdot \left(1 + \left(\frac{1}{2} \right)^i \right)$.

Therefore, $p_i = 1$, when $i = 0$ and p_i tends to $\frac{1}{2}$ for high value of i . Hence the proof.

From Theorem 3 it can be inferred that the bias of the linear approximation relating to the i^{th} bit position is $\left(\frac{1}{2} \right)^i$. In the following theorem, the maximum value of the biases of all possible linear approximations of the output bits is computed.

Theorem 4. *If two n bit numbers $X = (x_{n-1}, x_{n-2}, \dots, x_0)$ and $Y = (y_{n-1}, y_{n-2}, \dots, y_0)$ generate an n bit number $Z = (z_{n-1}, z_{n-2}, \dots, z_0)$ such that, $Z = F(X, Y)$, then the bias of the best linear approximation of the i^{th} output bit of Z is 2^{-i} .*

Proof. From the definition of F it is evident that $z_i = x_i \oplus y_i \oplus c_{i-1}$, where c_{i-1} is the carry propagated from i^{th} bit position. The carry c_{i-1} is the only nonlinear term of the equation. Therefore, in order to obtain various linear approximations for the nonlinear part, linear approximations have to be found out for the carry term. Each possible approximation of c_{i-1} , denoted by L_{i-1} will give rise to different biases which are equal to the bias of a linear approximation of z_i .

By the definition of c_i , we know that,

$c_i = x_i y_i \oplus x_{i-1} y_{i-1} \oplus \dots \oplus x_0 y_0$, i.e. c_i is a boolean function of $2(i+1)$ variables. It has been proved in Theorem 1, that c_i is a bent function. Hence, it has a nonlinearity value $2^{2(i+1)-1} - 2^{\frac{2(i+1)}{2}-1}$. Hence, the probability of match for the best linear approximation of c_i is : $1 - \frac{2^{2(i+1)-1} - 2^{2(i+1)/2-1}}{2^{2(i+1)}} = \frac{1}{2} + 2^{-(i+2)}$.

Therefore the output, $z_i = x_i \oplus y_i \oplus c_{i-1}$ can be approximated by a linear

equation, $z'_i = x_i \oplus y_i \oplus L_{i-1}$, where L_{i-1} is the best linear approximation for c_{i-1} .

Now, the largest probability that L_{i-1} matches c_{i-1} is $\frac{1}{2} + 2^{-(i-1+2)} = \frac{1}{2} + 2^{-(i+1)} = \frac{1}{2}(1 + 2^{-i})$.

Thus, the largest bias of the best linear approximation for c_{i-1} and hence z_i is 2^{-i} .

We observe from the above theorems that the bias of any linear approximations reduces considerably and makes the finding of linear approximations in the cipher with a large bias more difficult.

5 Brief description of NLS stream cipher

NLS key-stream generator uses NFSR whose outputs are given to a nonlinear filter NLF that produces output key-stream bits. Detail of the cipher can be found in [2].

NLS has two components, one NFSR and one NLF whose work is synchronised by a clock. The state of NFSR at time t is denoted by $\sigma_t = (r_t[0], \dots, r_t[16])$ where $r_t[i]$ is a 32-bit word. The state is determined by 17 words. The transition from the state σ_t to the state σ_{t+1} is defined as follows :

1. $r_{t+1}[i] = r_t[i + 1]$ for $i = 0, \dots, 15$;
2. $r_{t+1}[16] = f((r_t[0] \lll 19) \boxplus (r_t[15] \lll 9) \boxplus Konst) \oplus r_t[4]$;
3. if $t = 0 \pmod{f16}$, $r_{t+1}[2] = r_{t+1}[2] \boxplus t$;

where $f16$ is 65537 and \boxplus is the addition modulo 2^{32} . The *Konst* value is a 32-bit key dependent constant. The function $f: \{0, 1\}^{32} \rightarrow \{0, 1\}^{32}$ is constructed using an S-box with 8-bit input and 32-bit output and defined as $f(a) = \text{S-box}(a_H) \oplus a$ where a_H is the most significant 8 bits of 32-bit word a . Each output key-stream word ν_t of NLF is obtained as

$$\nu_t = NLF(\sigma_t) = (r_t[0] \boxplus r_t[16]) \oplus (r_t[1] \boxplus r_t[13]) \oplus (r_t[6] \boxplus Konst). \quad (1)$$

The cipher uses 32-bit words to ensure a fast keystream generation.

5.1 Suggested Modification

The NLS key-stream generator has two components, one NFSR and one NLF. We keep the NFSR same and change the Non-Linear Filter (NLF) only. We replace the modular additions used in the NLF by our proposed *Slash* function. We use *Slash* modulo 2^{32} . Hence in the modified NLS the output key-stream word ν'_t is obtained as :

$$\nu'_t = NLF'(\sigma_t) = (r_t[0] \odot r_t[16]) \oplus (r_t[1] \odot r_t[13]) \oplus (r_t[6] \odot Konst). \quad (2)$$

Here \odot is *Slash* modulo 2^{32} .

6 Brief description of Crossword Puzzle (CP) Attack

The CP attack proposed in [3] is based on linear distinguisher [7] which uses linear approximations of both the NFSR and the NLF. The roles of the two nonlinear components are :

- NFSR transforms the current state σ_i to the next state σ_{i+1} using some function $NF1$, $\sigma_{i+1} = NF1(\sigma_i)$.
- NLF produces an output ν_i from the current state σ_i through a function $NF2$, $\nu_i = NF2(\sigma_i)$.

The basic steps of the attack are :

1. Find a linear approximation of the non-linear state transition function used by NFSR : $l_1(\sigma_i) = \sigma_{i+1}$ with bias of ϵ_1 .
2. Find a linear approximation of the non-linear function applied by NLF : $l_2(\sigma_j) \oplus l_3(\nu_j) = 0$ with bias of ϵ_2 .
3. Obtain two sets of clock I and J such that $\sum_{i \in I} (l_1(\sigma_i) \oplus \sigma_{i+1}) = \sum_{j \in J} l_2(\sigma_j)$.
4. Build a distinguisher by computing

$$\sum_{i \in I} (l_1(\sigma_i) \oplus \sigma_{i+1}) \oplus \sum_{j \in J} (l_2(\sigma_j) \oplus l_3(\nu_j)) = \sum_{j \in J} l_3(\nu_j) = 0$$

which has bias of $\epsilon^{|I|} \cdot \epsilon^{|J|}$.

This is the basic outline of the attack. The attackers obtained linear approximations of both the NFSR and the NLF and combined them to build a linear distinguisher with high bias value. In the following subsections we introduce our suggested modification and show that how the attack can be thwarted using this modification. We show that in the modified version the bias of the distinguisher decreases to such a low value that any practical attack using this linear distinguisher is impossible.

7 Analysis of NFSR and NLF

As we have not changed the structure of the NFSR, the analysis given in [3] holds. We briefly describe the analysis of NFSR here. Let α_t be a 32-bit output of the S-box that defines the transition function f . Then, the following equation holds for the least significant bit.

$$\alpha_{t,(0)} \oplus r_t[0]_{(13)} \oplus r_t[15]_{(23)} \oplus Konst_{(0)} \oplus r_t[4]_{(0)} \oplus r_{t+1}[16]_{(0)} = 0 \quad (3)$$

where $\alpha_{t,(i)}$ and $x_{(i)}$ stand for the i -th bit off the 32 bit words α_t and x respectively. (Throughout the paper, this notation will be used). To make the analysis simpler initially $Konst$ is taken as zero.

7.1 Linear approximation of $\alpha_{t,(0)}$ and NFSR

In Table 1 the linear approximations for $\alpha_{t,(0)}$ has been given. For detail, reader can refer to [3]. Linear approximation of the NFSR can be obtained using the

Table 1. Linear approximations for $\alpha_{t,(0)}$ when $Konst = 0$

$r_t[0]_{(10)} \oplus r_t[0]_{(6)} \oplus r_t[15]_{(20)} \oplus r_t[15]_{(16)} \oplus r_t[15]_{(15)}$	$\frac{1}{2}(1 + 0.048828)$
$r_t[0]_{(10)} \oplus r_t[0]_{(6)} \oplus r_t[0]_{(5)} \oplus r_t[15]_{(20)} \oplus r_t[15]_{(16)}$	$\frac{1}{2}(1 + 0.048828)$
$r_t[0]_{(12)} \oplus r_t[15]_{(22)}$	$\frac{1}{2}(1 - 0.045410)$
$r_t[0]_{(12)} \oplus r_t[0]_{(11)} \oplus r_t[0]_{(10)} \oplus r_t[15]_{(22)} \oplus r_t[15]_{(21)} \oplus r_t[15]_{(20)}$	$\frac{1}{2}(1 - 0.020020)$

linear approximation for $\alpha_{t,(0)}$. If the first approximation from Table 1 is chosen, then the following linear equation :

$$\alpha_{t,(0)} = r_t[0]_{(10)} \oplus r_t[0]_{(6)} \oplus r_t[15]_{(20)} \oplus r_t[15]_{(16)} \oplus r_t[15]_{(15)} \quad (4)$$

holds with bias $0.048828 = 2^{-4.36}$. Combining equations (3) and (4), the following approximation for NFSR holds

$$\begin{aligned} r_t[0]_{(10)} \oplus r_t[0]_{(6)} \oplus r_t[15]_{(20)} \oplus r_t[15]_{(16)} \oplus r_t[15]_{(15)} = \\ r_t[0]_{(13)} \oplus r_t[15]_{(23)} \oplus Konst_{(0)} \oplus r_t[4]_{(0)} \oplus r_{t+1}[16]_{(0)} \end{aligned} \quad (5)$$

with bias $2^{-4.36}$.

7.2 Linear approximation of modular addition [3]

As the least significant bits are linear for modular addition \boxplus so the following equation holds with probability 1.

$$(r[x] \boxplus r[y])_{(0)} = r[x]_{(0)} \oplus r[y]_{(0)} \quad (6)$$

$x_{(i)}$ stands for i^{th} bit of 32-bit word x . All consecutive bits $i > 0$ of \boxplus are nonlinear. Consider the function $(r[x] \boxplus r[y])_{(i)} \oplus (r[x] \boxplus r[y])_{(i-1)}$. The function has a linear approximation as follows

$$(r[x] \boxplus r[y])_{(i)} \oplus (r[x] \boxplus r[y])_{(i-1)} = r[x]_{(i)} \oplus r[y]_{(i)} \oplus r[x]_{(i-1)} \oplus r[y]_{(i-1)} \quad (7)$$

that has the bias 2^{-1} .

In a similar way, the function $(r[x] \boxplus r[y])_{(i)} \oplus (r[x] \boxplus r[y])_{(i-1)} \oplus (r[x] \boxplus r[y])_{(i-2)} \oplus (r[x] \boxplus r[y])_{(i-3)}$ has the following approximation. For $i > 2$,

$$\begin{aligned} (r[x] \boxplus r[y])_{(i)} \oplus (r[x] \boxplus r[y])_{(i-1)} \oplus (r[x] \boxplus r[y])_{(i-2)} \oplus (r[x] \boxplus r[y])_{(i-3)} = \\ r[x]_{(i)} \oplus r[y]_{(i)} \oplus r[x]_{(i-1)} \oplus r[y]_{(i-1)} \oplus r[x]_{(i-2)} \oplus r[y]_{(i-2)} \oplus r[x]_{(i-3)} \oplus r[y]_{(i-3)} \end{aligned} \quad (8)$$

that has a bias of 2^{-2} .

7.3 Linear approximation of modular Slash

Let us look at the change in bias due to the introduction of modular *Slash* in place of modular addition. Let $r[z] = r[x] \oslash r[y]$. As the least significant bits are linear so the following equation holds with probability 1.

$$(r[x] \oslash r[y])_{(0)} = r[x]_{(0)} \oplus r[y]_{(0)} \quad (9)$$

which is same as of equation (6). All consecutive bits $i > 0$ are nonlinear.

The function $(r[x] \oslash r[y])_{(i)} \oplus (r[x] \oslash r[y])_{(i-1)}$ having a linear approximation as follows

$$(r[x] \oslash r[y])_{(i)} \oplus (r[x] \oslash r[y])_{(i-1)} = r[x]_{(i)} \oplus r[y]_{(i)} \oplus r[x]_{(i-1)} \oplus r[y]_{(i-1)} \quad (10)$$

has bias of at most 2^{-i} (theorem 4), considering this as the best linear approximation of the output bit $r[z]_{(i)}$.

In a similar way the function $(r[x] \oslash r[y])_{(i)} \oplus (r[x] \oslash r[y])_{(i-1)} \oplus (r[x] \oslash r[y])_{(i-2)} \oplus (r[x] \oslash r[y])_{(i-3)}$ has the following approximation. For $i > 2$,

$$(r[x] \oslash r[y])_{(i)} \oplus (r[x] \oslash r[y])_{(i-1)} \oplus (r[x] \oslash r[y])_{(i-2)} \oplus (r[x] \oslash r[y])_{(i-3)} = r[x]_{(i)} \oplus r[y]_{(i)} \oplus r[x]_{(i-1)} \oplus r[y]_{(i-1)} \oplus r[x]_{(i-2)} \oplus r[y]_{(i-2)} \oplus r[x]_{(i-3)} \oplus r[y]_{(i-3)} \quad (11)$$

has a bias of at most 2^{-i} (theorem 4), considering this as the best linear approximation of the output bit $r[z]_{(i)}$.

7.4 Linear approximation for NLF

Equation (1) defines the output key-stream generated by the original NLF and equation (2) defines the output key-stream generated by the modified NLF. The relation for the least significant bits of both the original and the modified NLF having the following form holds with probability one (as observed from equation (6) and (9)).

$$\begin{aligned} \nu_{t,(0)}/\nu'_{t,(0)} &= (r_t[0]_{(0)} \oplus r_t[16]_{(0)}) \oplus (r_t[1]_{(0)} \\ &\oplus r_t[13]_{(0)}) \oplus (r_t[6]_{(0)} \oplus Konst_{(0)}) \end{aligned} \quad (12)$$

For $2 \leq i \leq 31$ and using equation (7), the original NLF function has linear approximation of the following form :

$$\begin{aligned} \nu_{t,(i)} \oplus \nu_{t,(i-1)} &= (r_t[0]_{(i)} \oplus r_t[16]_{(i)} \oplus r_t[0]_{(i-1)} \oplus r_t[16]_{(i-1)}) \\ &\oplus (r_t[1]_{(i)} \oplus r_t[13]_{(i)} \oplus (r_t[1]_{(i-1)} \oplus r_t[13]_{(i-1)})) \\ &\oplus (r_t[6]_{(i)} \oplus Konst_{(i)} \oplus r_t[6]_{(i-1)} \oplus Konst_{(i-1)}) \end{aligned} \quad (13)$$

with the bias of $(2^{-1})^2 = 2^{-2}$ under the condition that $Konst = 0$ [3], when modular addition has been used in the Filter function.

When modular *Slash* has been used in the filter function, for $2 \leq i \leq 31$ and using equation (10), the modified NLF function has linear approximation of the following form :

$$\begin{aligned} \nu'_{t,(i)} \oplus \nu'_{t,(i-1)} &= (r_t[0]_{(i)} \oplus r_t[16]_{(i)} \oplus r_t[0]_{(i-1)} \oplus r_t[16]_{(i-1)}) \\ &\oplus (r_t[1]_{(i)} \oplus r_t[13]_{(i)} \oplus (r_t[1]_{(i-1)} \oplus r_t[13]_{(i-1)})) \\ &\oplus (r_t[6]_{(i)} \oplus Konst_{(i)} \oplus r_t[6]_{(i-1)} \oplus Konst_{(i-1)}) \end{aligned} \quad (14)$$

with the bias of $(2^{-i})^2 = 2^{-2i}$ under the condition, $Konst = 0$.

In case of modular addition, applying approximation (8), for $i > 2$ the following

expression holds

$$\begin{aligned}
& \nu_{t,(i)} \oplus \nu_{t,(i-1)} \oplus \nu_{t,(i-2)} \oplus \nu_{t,(i-3)} = \\
& (r_t[0]_{(i)} \oplus r_t[0]_{(i-1)} \oplus r_t[0]_{(i-2)} \oplus r_t[0]_{(i-3)} \oplus r_t[16]_{(i)} \oplus r_t[16]_{(i-1)} \\
& \oplus r_t[16]_{(i-2)} \oplus r_t[16]_{(i-3)}) \oplus (r_t[1]_{(i)} \oplus r_t[1]_{(i-1)} \oplus r_t[1]_{(i-2)} \oplus r_t[1]_{(i-3)} \quad (15) \\
& \oplus (r_t[13]_{(i)} \oplus r_t[13]_{(i-1)} \oplus r_t[13]_{(i-2)} \oplus r_t[13]_{(i-3)}) \oplus (r_t[6]_{(i)} \oplus r_t[6]_{(i-1)} \\
& \oplus r_t[6]_{(i-2)} \oplus r_t[6]_{(i-3)}) \oplus Konst_{(i)} \oplus Konst_{(i-1)} \oplus Konst_{(i-2)} \oplus Konst_{(i-3)}
\end{aligned}$$

with the bias $(2^{-2})^2 = 2^{-4}$ when $Konst = 0$.

In case of modular *Slash*, applying approximation (11), for $i > 2$ the following expression holds

$$\begin{aligned}
& \nu'_{t,(i)} \oplus \nu'_{t,(i-1)} \oplus \nu'_{t,(i-2)} \oplus \nu'_{t,(i-3)} = \\
& (r_t[0]_{(i)} \oplus r_t[0]_{(i-1)} \oplus r_t[0]_{(i-2)} \oplus r_t[0]_{(i-3)} \oplus r_t[16]_{(i)} \oplus r_t[16]_{(i-1)} \\
& \oplus r_t[16]_{(i-2)} \oplus r_t[16]_{(i-3)}) \oplus (r_t[1]_{(i)} \oplus r_t[1]_{(i-1)} \oplus r_t[1]_{(i-2)} \oplus r_t[1]_{(i-3)} \quad (16) \\
& \oplus (r_t[13]_{(i)} \oplus r_t[13]_{(i-1)} \oplus r_t[13]_{(i-2)} \oplus r_t[13]_{(i-3)}) \oplus (r_t[6]_{(i)} \oplus r_t[6]_{(i-1)} \\
& \oplus r_t[6]_{(i-2)} \oplus r_t[6]_{(i-3)}) \oplus Konst_{(i)} \oplus Konst_{(i-1)} \oplus Konst_{(i-2)} \oplus Konst_{(i-3)}
\end{aligned}$$

with bias $(2^{-i})^2 = 2^{-2i}$, when $Konst = 0$.

8 Complexity comparison of CP attack on the original and modified NLS

The main idea behind the CP attack is to find the best combination of approximations for both NFSR and NLF, while the state bits of the shift register vanish and the bias of the resulting approximation is as big as possible [3]. The case for $Konst = 0$ has been studied at first and then the attack has been extended to $Konst \neq 0$. We show that for non-zero $Konst$, even if we assume all zero values for the lower 3 bytes of the $Konst$, the attack proposed in [3] does not work on the modified NLS.

8.1 Case for $Konst = 0$

We first describe here the approximation chosen by the attacker in [3] and then we show that in modified NLS, how the bias of this approximation decreases to a low value such that any practical attack become impossible.

The linear approximations of $\alpha_{t,(0)}$ are given in Table 1. The third approximation from the table has been chosen which is

$$\alpha_{t,(0)} = r_t[0]_{(12)} \oplus r_t[15]_{(22)} \quad (17)$$

and the bias of this approximation is $0.045410 = 2^{-4.46}$. By combining equations (3) and (17) the following approximation has been obtained

$$r_t[0]_{(12)} \oplus r_t[15]_{(22)} \oplus r_t[0]_{(13)} \oplus r_t[15]_{(23)} \oplus r_t[4]_{(0)} \oplus r_{t+1}[16]_{(0)} = 0 \quad (18)$$

which has the same bias.

Approximation (18) has been divided into two parts : the least significant bits and the other bits as

$$\begin{aligned} l_1(r_t) &= r_t[4]_{(0)} \oplus r_{t+1}[16]_{(0)} \\ l_2(r_t) &= r_t[0]_{(12)} \oplus r_t[0]_{(13)} \oplus r_t[15]_{(22)} \oplus r_t[15]_{(23)} \end{aligned} \quad (19)$$

Clearly, $l_1(r_t) \oplus l_2(r_t) = 0$ with the bias $2^{-4.46}$. Since, $l_1(r_t)$ has only the least significant bit variables, approximation (12) can be applied which is true with probability one. The following set of approximations are obtained.

$$\begin{aligned} l_1(r_t) &= r_t[4]_{(0)} \oplus r_{t+1}[16]_{(0)} \\ l_1(r_{t+1}) &= r_{t+1}[4]_{(0)} \oplus r_{t+2}[16]_{(0)} \\ l_1(r_{t+6}) &= r_{t+6}[4]_{(0)} \oplus r_{t+7}[16]_{(0)} \\ l_1(r_{t+13}) &= r_{t+13}[4]_{(0)} \oplus r_{t+14}[16]_{(0)} \\ l_1(r_{t+16}) &= r_{t+16}[4]_{(0)} \oplus r_{t+17}[16]_{(0)} \end{aligned} \quad (20)$$

Adding up all approximations of (20) and by applying approximation (12), the following equation can be written

$$l_1(r_t) \oplus l_1(r_{t+1}) \oplus l_1(r_{t+6}) \oplus l_1(r_{t+13}) \oplus l_1(r_{t+16}) = \nu_{t+4,(0)} \oplus \nu_{t+17,(0)} \quad (21)$$

since $r_{t+p}[0] = r_t[p]$.

Now focusing on $l_2(r_t)$, where the bit positions involved are 12, 13, 22 and 23,

$$\begin{aligned} l_2(r_t) &= r_t[0]_{(12)} \oplus r_t[0]_{(13)} \oplus r_t[15]_{(22)} \oplus r_t[15]_{(23)} \\ l_2(r_{t+1}) &= r_{t+1}[0]_{(12)} \oplus r_{t+1}[0]_{(13)} \oplus r_{t+1}[15]_{(22)} \oplus r_{t+1}[15]_{(23)} \\ l_2(r_{t+6}) &= r_{t+6}[0]_{(12)} \oplus r_{t+6}[0]_{(13)} \oplus r_{t+6}[15]_{(22)} \oplus r_{t+6}[15]_{(23)} \\ l_2(r_{t+13}) &= r_{t+13}[0]_{(12)} \oplus r_{t+13}[0]_{(13)} \oplus r_{t+13}[15]_{(22)} \oplus r_{t+13}[15]_{(23)} \\ l_2(r_{t+16}) &= r_{t+16}[0]_{(12)} \oplus r_{t+16}[0]_{(13)} \oplus r_{t+16}[15]_{(22)} \oplus r_{t+16}[15]_{(23)} \end{aligned} \quad (22)$$

Since, $r_{t+p}[0] = r_t[p]$, the above approximations are presented as follows.

$$\begin{aligned} l_2(r_t) &= r_t[0]_{(12)} \oplus r_t[0]_{(13)} \oplus r_{t+15}[0]_{(22)} \oplus r_{t+15}[0]_{(23)} \\ l_2(r_{t+1}) &= r_t[1]_{(12)} \oplus r_t[1]_{(13)} \oplus r_{t+15}[1]_{(22)} \oplus r_{t+15}[1]_{(23)} \\ l_2(r_{t+6}) &= r_t[6]_{(12)} \oplus r_t[6]_{(13)} \oplus r_{t+15}[6]_{(22)} \oplus r_{t+15}[6]_{(23)} \\ l_2(r_{t+13}) &= r_t[13]_{(12)} \oplus r_t[13]_{(13)} \oplus r_{t+15}[13]_{(22)} \oplus r_{t+15}[13]_{(23)} \\ l_2(r_{t+16}) &= r_t[16]_{(12)} \oplus r_t[16]_{(13)} \oplus r_{t+15}[16]_{(22)} \oplus r_{t+15}[16]_{(23)} \end{aligned} \quad (23)$$

For original NLS, approximations (13) and (23) are combined which leads to the following approximation.

$$\begin{aligned} l_2(r_t) \oplus l_2(r_{t+1}) \oplus l_2(r_{t+6}) \oplus l_2(r_{t+13}) \oplus l_2(r_{t+16}) &= \\ \nu_{t,(12)} \oplus \nu_{t,(13)} \oplus \nu_{t+15,(22)} \oplus \nu_{t+15,(23)} & \end{aligned} \quad (24)$$

By combining the approximations (21) and (24) the final approximation has been obtained that defines the distinguisher in [3], i.e.

$$\begin{aligned} l_1(r_t) \oplus l_1(r_{t+1}) \oplus l_1(r_{t+6}) \oplus l_1(r_{t+13}) \oplus l_1(r_{t+16}) & \\ \oplus l_2(r_t) \oplus l_2(r_{t+1}) \oplus l_2(r_{t+6}) \oplus l_2(r_{t+13}) \oplus l_2(r_{t+16}) & \\ = \nu_{t,(12)} \oplus \nu_{t,(13)} \oplus \nu_{t+15,(22)} \oplus \nu_{t+15,(23)} \oplus \nu_{t+4,(0)} \oplus \nu_{t+17,(0)} & \\ = 0 & \end{aligned} \quad (25)$$

As approximation (18) has been used five times and approximation (13) twice, the bias of the approximation (25) is $(2^{-4.46})^5 \cdot (2^{-2})^2 = 2^{-26.3}$. Therefore, the complexity of the attack is $2^{52.6}$. For the modified NLS, to obtain the same distinguisher defined above, we have to combine approximation (14) and (23) which leads to the following approximation.

$$\begin{aligned} & l_2(r_t) \oplus l_2(r_{t+1}) \oplus l_2(r_{t+6}) \oplus l_2(r_{t+13}) \oplus l_2(r_{t+16}) = \\ & \nu'_{t,(12)} \oplus \nu'_{t,(13)} \oplus \nu'_{t+15,(22)} \oplus \nu'_{t+15,(23)} \end{aligned} \quad (26)$$

By combining the approximations (21) (as same expression holds for ν' also) and (26) the final approximation is obtained that defines the distinguisher

$$\begin{aligned} & l_1(r_t) \oplus l_1(r_{t+1}) \oplus l_1(r_{t+6}) \oplus l_1(r_{t+13}) \oplus l_1(r_{t+16}) \\ & \oplus l_2(r_t) \oplus l_2(r_{t+1}) \oplus l_2(r_{t+6}) \oplus l_2(r_{t+13}) \oplus l_2(r_{t+16}) \\ & = \nu'_{t,(12)} \oplus \nu'_{t,(13)} \oplus \nu'_{t+15,(22)} \oplus \nu'_{t+15,(23)} \oplus \nu'_{t+4,(0)} \oplus \nu'_{t+17,(0)} \\ & = 0 \end{aligned} \quad (27)$$

As approximation (18) has been used five times and approximation (14) twice, the bias of the approximation (27) is $(2^{-4.46})^5 \cdot (2^{-13}) \cdot (2^{-23}) = 2^{-58.3}$ (as bit positions 13 and 23 are used in the approximation). Therefore, the complexity of the attack for the modified NLS is $2^{116.6}$. Since the specification of the NLS cipher allows the adversary to observe up to 2^{80} keystream words per one key/nonce pair [2], the attack is not successful for the modified NLS as bias of the distinguisher is less than 2^{-40} .

8.2 Case for $Konst \neq 0$

The biases of linear approximations of both $\alpha_{t,(0)}$ and the NLF vary with $Konst$ as it occurs as a parameter. Bias of the linear distinguisher has been explored in [3] and it has been showed that with non-zero $Konst$ the bias reduces. According to [3] the $Konst$ has been divided into two parts as $Konst = (Konst_{(H)}, Konst_{(L)})$ where $Konst_{(H)} = (Konst_{(31)}, \dots, Konst_{(24)})$, and $Konst_{(L)} = (Konst_{(23)}, \dots, Konst_{(0)})$. The biases of linear approximations of $\alpha_{t,(0)}$ depend on $Konst_{(H)}$ and those of NLF depend on $Konst_{(L)}$. Here we have explored only the case where $Konst_{(H)} \neq 0$ and $Konst_{(L)} = 0$.

Since the most significant 8-bits of $Konst$ contribute to the form of $\alpha_{t,(0)}$, bias of approximation (17) fluctuates according to the value of $Konst_{(H)}$. The bias of (17) becomes smallest when $Konst_{(H)}$ is around 51 or 179 and the biggest when $Konst_{(H)}$ is around 127 or 255. The average bias of approximation (17) with $Konst_{(H)}$ is $2^{-5.19}$ [3].

As explained in [3], for the original NLS, bias of the NLF with non-zero $Konst_{(L)}$ decreases and the bias of (13) is around 2^{-3} for any $i > 0$. Hence the bias of the distinguisher (25) with non-zero $Konst$ becomes $(2^{-5.19})^5 \cdot (2^{-3}) = 2^{-31.95}$.

In case of the modified NLS, even if we consider $Konst_{(L)} = 0$ and $Konst_{(H)} \neq 0$, combining approximations (17) and (14), the bias of distinguisher (27) becomes $(2^{-5.19})^5 \cdot (2^{-13}) \cdot (2^{-23}) = 2^{-61.95}$, which is low enough to thwart any linear distinguishing attack.

8.3 Multiple Distinguisher

For original NLS the bias of the distinguisher (25) is very small for some values of $Konst_{(H)}$ [3]. In order to address this problem, the attackers took the fourth approximation from Table 1 which is

$$\alpha_{t,(0)} = r_t[0]_{(12)} \oplus r_t[0]_{(11)} \oplus r_t[0]_{(10)} \oplus r_t[15]_{(22)} \oplus r_t[15]_{(21)} \oplus r_t[15]_{(20)} \quad (28)$$

having average bias of $2^{-6.2}$. Using approximation (28), another approximation of NFSR has been built which is

$$\begin{aligned} & r_t[0]_{(10)} \oplus r_t[0]_{(11)} \oplus r_t[0]_{(12)} \oplus r_t[0]_{(13)} \oplus r_t[15]_{(20)} \oplus r_t[15]_{(21)} \oplus r_t[15]_{(22)} \\ & \oplus r_t[15]_{(23)} \oplus Konst_{(0)} \oplus r_t[4]_{(0)} \oplus r_{t+1}[16]_{(0)} = 0 \end{aligned} \quad (29)$$

By combining approximations (15) and (29) a new distinguisher has been built having a bias of $2^{-37.8}$. The distinguisher is as follows

$$\begin{aligned} & \nu_{t,(10)} \oplus \nu_{t,(11)} \oplus \nu_{t,(12)} \oplus \nu_{t,(13)} \oplus \nu_{t+15,(20)} \oplus \nu_{t+15,(21)} \oplus \nu_{t+15,(22)} \oplus \nu_{t+15,(23)} \\ & \oplus \nu_{t+4,(0)} \oplus \nu_{t+17,(0)} = 0 \end{aligned} \quad (30)$$

By observing two distinguishers together and selecting always the better bias among them, the success rate of the distinguishing attack has been improved. For the modified NLS approximation (16) and (29) must be combined to obtain the above distinguisher. The new distinguisher is

$$\begin{aligned} & \nu'_{t,(10)} \oplus \nu'_{t,(11)} \oplus \nu'_{t,(12)} \oplus \nu'_{t,(13)} \oplus \nu'_{t+15,(20)} \oplus \nu'_{t+15,(21)} \oplus \nu'_{t+15,(22)} \oplus \nu'_{t+15,(23)} \\ & \oplus \nu'_{t+4,(0)} \oplus \nu'_{t+17,(0)} = 0 \end{aligned} \quad (31)$$

Distinguisher (31) has a bias $(2^{-6.2})^5 \cdot (2^{-13}) \cdot (2^{-23}) = 2^{-67}$ (the calculation is similar as in section 8.1). Here we observe that for both the distinguisher mentioned in [3], the bias is too low for any attack in case of the modified NLS. The data complexity in both the cases are well above 2^{80} . As only 2^{80} key-stream words per key/nonce pair is allowed to be observed by an adversary (as per the specification of NLS), the linear distinguishing attack or CP attack mentioned in [3] can be resisted by the proposed modification.

9 Hardware and time complexity

We have analyzed the gate count and time delay of our proposed key mixing function slash. Comparison and gate count and time delay is shown in Table 2. Time delay is given in terms of *AND* gate delay. Delay of 1 *XOR* gate is considered to be equivalent to 1.5 *AND* gate delay [8] and delay of 1 *AND* gate and 1 *OR* gate are considered to be equal.

10 Conclusions

In this work we have modified the stream cipher NLS which is a candidate of the eSTREAM project to prevent it against the Crossword Puzzle attack [3].

Table 2. Comparison of Gate Count and Time Delay

Function	Forward				
	Gate Count				Time Delay
	$\#XOR$	$\#AND$	$\#OR$	$\#NOT$	
Addition modulo 2^n	$(2n - 1)$	$(2n - 3)$	$(n - 2)$	-	$2.5n + .5$ AND gate
Slash modulo 2^n	$3(n - 1)$	$n - 1$	-	-	$1.5n$ AND gate
	Reverse				
	Gate Count				Time Delay
	$\#XOR$	$\#AND$	$\#OR$	$\#NOT$	
Subtraction modulo 2^n	$(3n - 1)$	$(2n - 3)$	$(n - 2)$	-	$2.5n + 2$ AND gate
Reverse Slash modulo 2^n	$3(n - 1)$	$(2n - 3)$	-	$2(n - 1)$	$3(n - 1)$ AND gate

We modified the nonlinear filter (NLF) of NLS by replacing the modular addition with a new boolean operator modular *Slash*. The paper shows that the complexity of the CP attack against the modified NLS has been increased to around $O(2^{120})$ keystream words from $O(2^{60})$ keystream words as published in [3] against the original cipher. As the specification of the NLS allows only 2^{80} keystream words to be observed per key/nonce pair [2], this attack becomes impractical against the modified NLS. We also showed that both the hardware cost and time delay of modular *Slash* is less than modular addition. To summarize, the paper shows that by suitably modifying the modular addition with modular *Slash*, the stream cipher NLS could be strengthened against the CP attack at a lower hardware cost.

References

1. eSTREAM project. <http://www.ecrypt.eu.org/stream/>.
2. G. Rose, P. Hawkes, M.P., de Vries, M.W.: Primitive specification for nls. <http://www.ecrypt.eu.org/stream/nls.html> (April 2005)
3. Cho, J.Y., Pieprzyk, J.: Crossword Puzzle Attack on NLS. In: SAC 2006. (2006)
4. Rothaus, O.S.: On “Bent” Functions. Journal of Combinatorial Theory **20**(A) (1976) 300–305
5. Macwilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North Holland (January 1983)
6. Matsui, M.: Linear Cryptanalysis method for DES cipher. In: Advances in Cryptology-Eurocrypt 1993, Springer, volume 765 of LNCS (1993) 386–397
7. Golic, J.D.: Linear models for keystream generators. IEEE Transactions on Computers **45**(1) (1996) 41–49
8. Uyemura, J.P. In: Introduction to VLSI Circuits and Systems. John Wiley & Sons (2002)